

# Библиотека OSCAT Building

## Документация на русском

Версия библиотеки: 1.00

Версия перевода: 1.0 (19.06.2019)



---

## Оглавление

Оглавление.....	2
0. Комментарий переводчиков .....	5
1. Правовые вопросы .....	7
1.1. Отказ от ответственности.....	7
1.2. Лицензионное соглашение .....	7
1.3. Зарегистрированные товарные знаки .....	8
1.4. Использование по назначению.....	8
1.5. Остальное.....	8
2. Введение .....	9
2.1. Задачи.....	9
2.2. Соглашения .....	10
2.3. Средства тестирования .....	11
2.4. Версионность .....	12
2.5. Техподдержка.....	12
2.6. Структура библиотек OSCAT .....	12
3. Специальные функции .....	13
3.1. BUILDING_VERSION .....	13
4. Исполнительные механизмы .....	14
4.1. ACTUATOR_2P.....	14
4.2. ACTUATOR_3P.....	17
4.3. ACTUATOR_A .....	20
4.4. ACTUATOR_COIL .....	22
4.5. ACTUATOR_PUMP.....	23
4.6. ACTUATOR_UD.....	24
4.7. AUTORUN .....	26
5. HVAC.....	28
5.1. AIR_DENSITY.....	28
5.2. AIR_ENTHALPY .....	29
5.3. BOILER.....	31
5.4. BURNER.....	33

---

5.5. DEW_CON .....	36
5.6. DEW_RH .....	37
5.7. DEW_TEMP .....	38
5.8. HEAT_INDEX .....	39
5.9. HEAT_METER.....	40
5.10. HEAT_TEMP.....	41
5.11. LEGIONELLA.....	43
5.12. SDD .....	45
5.13. SDD_NH3 .....	46
5.14. SDT_NH3 .....	47
5.15. T_AVG24.....	48
5.16. TANK_LEVEL .....	49
5.17. TANK_VOL1 .....	50
5.18. TANK_VOL2 .....	51
5.19. TEMP_EXT .....	52
5.20. WATER_CP.....	55
5.21. WATER_DENSITY .....	56
5.22. WATER_ENTHALPY .....	57
5.23. WCT .....	58
6. Кнопки, генераторы импульсов, таймеры .....	59
6.1. CLICK.....	59
6.2. CLICK_MODE .....	61
6.3. DEBOUNCE .....	62
6.4. DIMM_2 .....	63
6.5. DIMM_I .....	65
6.6. F_LAMP .....	67
6.7. PULSE_LENGTH.....	69
6.8. PULSE_T.....	70
6.9. SW_RECONFIG.....	71
6.10. SWITCH_I.....	72
6.11. SWITCH_X.....	73
6.12. TIMER_1 .....	74
6.13. TIMER_2 .....	75
6.14. TIMER_EVENT_DECODE .....	77
6.15. TIMER_EXT .....	78

---

---

6.16. TIMER_P4 .....	81
7. Управление жалюзи .....	86
7.1. Вступление .....	86
7.2. BLIND_ACTUATOR .....	87
7.3. BLIND_CONTROL .....	89
7.4. BLIND_CONTROL_S .....	91
7.5. BLIND_INPUT .....	93
7.6. BLIND_NIGHT .....	96
7.7. BLIND_SCENE .....	98
7.8. BLIND_SECURITY .....	100
7.9. BLIND_SET .....	102
7.10. BLIND_SHADE .....	104
7.11. BLIND_SHADE_S .....	107

## 0. Комментарий переводчиков

Библиотеки **OSCAT** хорошо знакомы значительному количеству специалистов в области программирования ПЛК. Полтора года назад мы опубликовали русскоязычную документацию на основную из OSCAT-библиотек – **OSCAT Basic**. За это время мы получили определенное количество отзывов и сообщений об ошибках – и хотим выразить признательность всем, кто написал нам.

Большую часть 2018 года мы занимались переводом документации от [PLCopen](#), и к осени закончили работу над 3 документами:

- **PLCopen Coding Guidelines** – сборник рекомендаций по разработке ПО на языках стандарта МЭК 61131-3;
- **PLCopen Compliant Libraries** – руководство по созданию библиотек, совместимых с моделью поведения **PLCopen Behavior Model (CAA Behavior Model)**;
- **PLCopen SFC** – описание и особенности использования языка проектирования SFC из стандарта МЭК 61131-3.

Итоги этой работы были подведены в докладе одного из наших участников на конференции [CODESYS User Conference Russia 2018](#).

Немного отдохнув, мы решили обсудить: что делать дальше? Поиски ответа не заняли много времени, и вот мы опять погрузились в OSCAT – на этот раз, в описание библиотеки **OSCAT Building**.

**OSCAT Building** – это библиотека модулей, которые могут оказаться полезными при автоматизации зданий (как вы правильно подумали – речь, в основном, об [HVAC](#)). Библиотека основана на OSCAT Basic и активно использует ее функционал. Структурно OSCAT Building состоит из 4 разделов:

- [Исполнительные механизмы](#);
- [HVAC](#);
- [Кнопки, генераторы импульсов, таймеры](#);
- [Управление жалюзи](#).

Одновременно с релизом описания OSCAT Building мы выложим обновленное описание на OSCAT Basic, где исправим найденные ошибки. Наверное, кому-то будет интересно – планируем ли мы браться за создание описания для третьей и последней библиотеки из семейства OSCAT – **OSCAT Network**? В настоящий момент вполне определенно можно ответить, что нет. Это связано с отсутствием прямой переносимости, сложностью блоков и сомнительной ценностью значительной части функционала в настоящее время (все же, с момента создания библиотеки прошло почти 10 лет, и коммуникационные возможности ПЛК стали значительно более продвинутыми).

Все наши переводы доступны на странице <https://oscat.ru>

Мы приветствуем распространение перевода при соблюдении трех простых условий:

- на некоммерческой основе;
- при отсутствии каких бы то ни было модификаций файла перевода;
- в случае публикации файла (или его фрагментов) на любых ресурсах, использования фрагментов файла в составе ПО и т. д. вы обязуетесь уведомить нас по электронной почте [OscatLibRu@gmail.com](mailto:OscatLibRu@gmail.com)

Если вы нашли в переводе ошибку или неточность, то сообщите, пожалуйста, об этом нам, написав на [OscatLibRu@gmail.com](mailto:OscatLibRu@gmail.com)

**Переводчики:** Евгений Кислов, Екатерина Чибисова

**Редактор:** Алексей Осинский

## 1. Правовые вопросы

### 1.1. Отказ от ответственности

Программные модули (функции и функциональные блоки), включенные в библиотеку **OSCAT**, предоставляются для использования в качестве шаблонов и примеров для разработки прикладного программного обеспечения ПЛК согласно стандарту [МЭК 61131-3](#). Разработчики библиотеки не принимают на себя никаких обязательств по поводу работоспособности программных модулей. Поскольку библиотека распространяется безвозмездно, в допустимых законодательством пределах отсутствуют какие-либо гарантийные обязательства. Если иное не указано в письменной форме, владельцы авторских прав и/или третьи лица распространяют модули «как есть» («as is»), без каких-либо гарантий, явных или подразумеваемых, в том числе по удобству использования в конкретных ситуациях. Вся полнота риска и ответственность за соответствие критериям качества, функциональности и точности лежит на пользователе. В случае, если библиотека или ее отдельные фрагменты содержат ошибки, затраты на исправление и/или изменение программных модулей лежат на пользователе. Если библиотека или ее отдельные фрагменты используются для создания программного обеспечения или применяются при разработке проектов, то пользователь принимает на себя ответственность за его/их производительность, качество и корректное функционирование. Ответственность для сообщества разработчиков **OSCAT** в явном виде исключается.

Пользователи **OSCAT** должны принимать меры по тестированию и контролю качества приложений, чтобы исключить ущерб от потенциальных ошибок библиотеки. Настоящее лицензионное соглашение и отказ от ответственности имеют одинаковую юридическую силу для библиотеки и документации на нее, даже если это не указывается в явном виде.

### 1.2. Лицензионное соглашение

Библиотека **OSCAT** распространяется на безвозмездной основе и может быть использована в личных или служебных целях. Распространение библиотеки приветствуется при условии сохранения ее бесплатности и публикации ссылки на веб-страницу [www.oscat.de](http://www.oscat.de). Если библиотека публикуется для загрузки или распространяется на носителях информации, то должно быть в явном виде указано авторство сообщества **OSCAT** и приведена ссылка на веб-страницу [www.oscat.de](http://www.oscat.de).

### **1.3. Зарегистрированные товарные знаки**

Все товарные знаки, упомянутые в данной документации, приводятся без ссылок на их регистрацию и владельцев. Существование таких прав не может быть исключено. Все упомянутые товарные знаки являются собственностью их владельцев. По этой причине не допускается коммерческое использование документации или ее фрагментов.

### **1.4. Использование по назначению**

Программные модули (функции и функциональные блоки), включенные в библиотеку **OSCAT**, были разработаны исключительно для использования специалистами, обладающими соответствующей квалификацией в области программирования ПЛК. Пользователи несут ответственность за соблюдение всех действующих стандартов и нормативных актов, которые связаны с использованием программных модулей. Библиотека и ее описание не соответствуют никаким стандартам и нормативным актам.

### **1.5. Остальное**

Все юридически обязательные правовые нормы приведены в [главе 1](#) данного документа. Выдвижение юридических претензий на основании содержимого данного документа полностью исключено, кроме случаев, упомянутых в [главе 1](#).



## 2. Введение

### 2.1. Задачи

**OSCAT** является акронимом названия **Open Source Community for Automation Technology** («Сообщество открытого ПО для систем автоматизации»).

Сообщество **OSCAT** занимается созданием библиотеки с открытым исходным кодом в соответствии со стандартом [МЭК 61131-3](#), которая не имеет привязок к конкретному оборудованию и, соответственно, может использоваться на любых ПЛК, поддерживающих данный стандарт. В настоящее время большинство производителей ПЛК бесплатно предоставляют свои проприетарные библиотеки, но этот подход имеет ряд существенных недостатков:

1. Исходный код большинства проприетарных библиотек закрыт для пользователей, что делает внесение изменений и исправление потенциально возможных ошибок сложным и, зачастую, даже невозможным;
2. Использование проприетарных библиотек при разработке программ на графических языках (LD, FBD, CFC) может быть сложным, неэффективным и приводить к ошибкам, поскольку существующие функции не обязательно соответствуют потребностям пользователей. Исходный код библиотек при этом закрыт для редактирования;
3. Использование проприетарных библиотек затрудняет перенос приложений с одних ПЛК на другие (особенно в случае ПЛК различных производителей), тем самым сводя на нет преимущества стандарта [МЭК 61131-3](#). Замена проприетарных библиотек одного производителя библиотеками другого в большинстве случаев невозможна (или как минимум требует затрат на адаптацию программы) из-за специфических различий;
4. Понимание принципов работы сложных программных модулей без наличия их исходного кода может быть крайне затруднено, что снижает эффективность программ и приводит к различного рода ошибкам.

Сообщество **OSCAT** с помощью разработки одноименной библиотеки планирует создать функциональный и понятный стандарт программирования ПЛК, исходные коды которого открыты и тщательно протестированы в различных приложениях. Использование библиотеки в различных проектах поможет исправлять ошибки и улучшать функционал, что является очень практичным. Сообщество **OSCAT** позиционирует библиотеку как шаблон для разработки приложений, а не законченный продукт. Пользователь несет полную ответственность за тестирование своих приложений с целью верификации заданным критериям точности, качества и функциональности. По этой причине мы должны сослаться на лицензионное соглашение и отказ от ответственности, опубликованные в [главе 1](#) данного документа.

## 2.2. Соглашения

### 1. Параметры модулей

Чтобы облегчить процесс разработки и упростить работу со сложными функциями, многие модули библиотеки **OSCAT** имеют параметры. Параметры представляют переменные класса **VAR\_INPUT CONSTANT**, которые задаются при первом вызове модуля и не могут изменить свои значения в процессе его работы. На языке **CFC** для открытия диалога редактирования параметров необходимо дважды нажать **ЛКМ** на вкладку **Параметры**, расположенную внутри элемента. Можно привязать к параметру переменную или задать ему конкретное значение. Задаваемые параметры являются уникальными для каждого экземпляра ФБ или вызова функции.

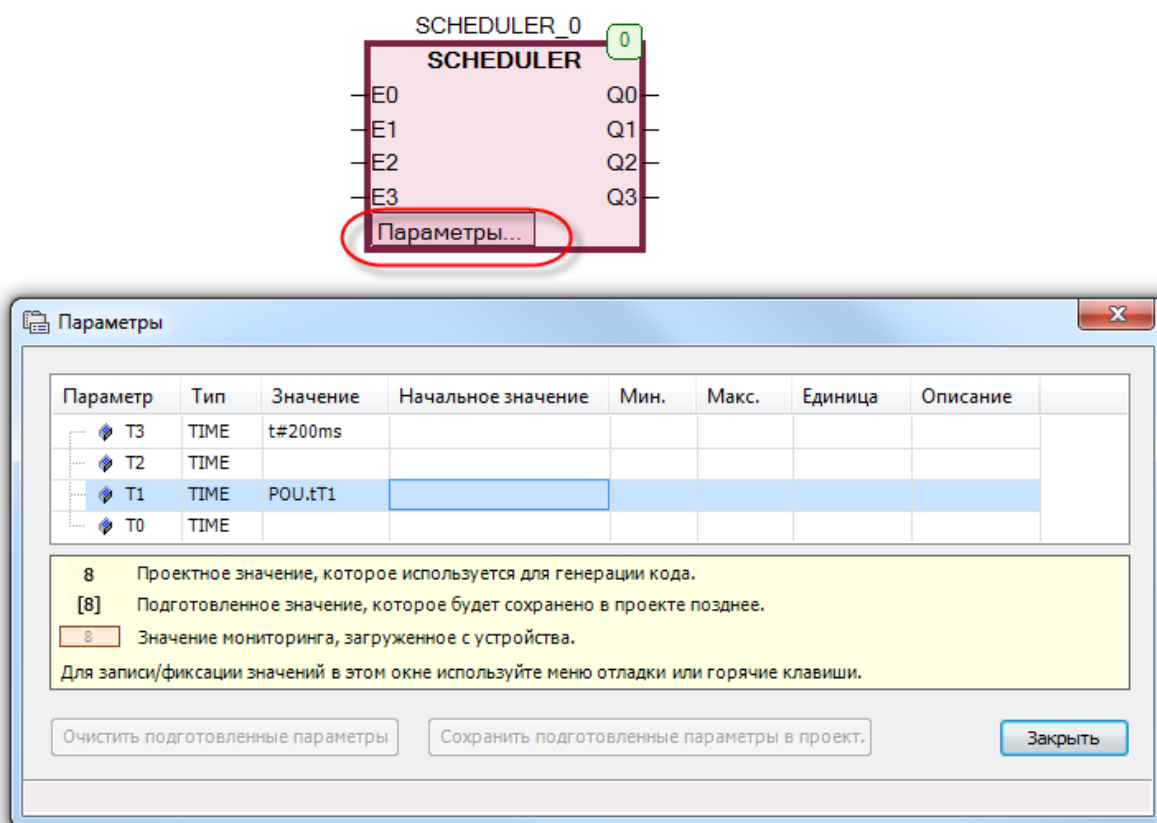


Рис. 2.1. Настройка параметров модуля на языке **CFC**

На языке **ST** значения параметров задаются непосредственно при вызове модуля:

```
3 | SCHEDULER_0 (T1:=tT1, T3:=T#200MS);
```

Рис. 2.2. Вызов модуля с параметрами на языке **ST**

## 2. Отчеты о событиях и ошибках (ESR)

Модули, выполняющие сложные функции, обычно имеют выход **STATIS**, который используется для получения информации о состоянии блока и возникших ошибках. Значение **0** соответствует нормальной работе модуля. Расшифровка кодов приводится в описании соответствующего модуля. Библиотека **OSCAT Basic** содержит специальные ESR-модули, которые могут собирать информацию об ошибках и состояниях с метками времени, которая при необходимости может быть сохранена в архив или передана на верхний уровень АСУ (реализация архивации и передачи должны быть выполнена пользователем). Это позволяет с помощью стандартных возможностей библиотеки вести аварийные и оперативные журналы, не используя дополнительный функционал системы. Более подробная информация приведена в описании на библиотеку.

### 2.3. Средства тестирования

Библиотека **OSCAT** изначально создана в среде **CoDeSys 2.x** и портирована в другие среды разработки. Библиотека протестирована на следующем оборудовании:

ПЛК	Среда разработки
Beckhoff BX 9000	TwinCAT PLC Control Version 2.10.0
Beckhoff CX 9001-1001	TwinCAT PLC Control Version 2.10.0
Wago 750-841	CoDeSys 2.3.9.31
Möller EC4P222	
CoDeSys Simulation on I386	
CODESYS Control Win V3	CODESYS 3.4
-	S7 and STEP 7: Библиотека поддерживается и тестируется на STEP7 начиная с версии 1.5.
-	PCWORX / MULTIPROG: Библиотека поддерживается и тестируется на MULTIPROG начиная с версии 2.6.
Bosch Rexroth IndraLogic XLC L25/L45/L65	Indraworks 12VRS
Bosch Rexroth IndraMotion MLC L25/L45/L65	
Bosch Rexroth IndraMotion MTX L45/L65/L85	

*Примечание переводчиков* – в процессе перевода библиотека тестировалась на следующих ПЛК:

ПЛК	Среда разработки
CODESYS Control Win V3	CODESYS 3.5 SP11 Patch 5
<a href="#">Berghof DC2007</a>	CODESYS 3.5 SP11 Patch 4
<a href="#">ОВЕН СПК1xx [M01]</a>	CODESYS 3.5 SP11 Patch 5

## 2.4. Версионность

Данное руководство постоянно актуализируется. Рекомендуется загрузить последнюю версию руководства с сайта [www.oscat.de](http://www.oscat.de). Помимо руководства, на сайте доступен документ с историей версий всех модулей библиотеки (даты добавления, внесенные изменения и т.д.).

*Примечание переводчиков* – последняя на данный момент (начало 2019 г.) версия библиотеки OSCAT Building и ее документации были выпущены зимой 2011 года (версия **1.00**). На основе этой версии и составлено данное описание.

## 2.5. Техподдержка

Техподдержка осуществляется пользователями форума [www.oscat.de](http://www.oscat.de). При этом техподдержка в каждом конкретном случае не гарантируется, даже при наличии ошибок в библиотеке или отдельных модулях. Техподдержка осуществляется силами пользователей библиотеки и на добровольной основе. Обновление библиотеки и документации обычно происходит раз в месяц, актуальные версии выкладываются на сайте [www.oscat.de](http://www.oscat.de). Никакие претензии по техническому обслуживанию, устранению ошибок и сопровождению не принимаются. Пожалуйста, не отправляйте запросы по технической поддержке на электронную почту. Ответ последует значительно быстрее, если они будут размещены на форуме.

## 2.6. Структура библиотек OSCAT

Изначально библиотека OSCAT представляла собой единую сущность. По мере развития число модулей стало настолько большим, что разработчики приняли решение разделить их на три отдельные библиотеки:

1. **OSCAT Basic** – библиотека базовых, низкоуровневых модулей.
2. **OSCAT Building** – библиотека модулей для автоматизации зданий (например, модули управления жалюзи). Именно ей посвящен данный документ.
3. **OSCAT Network** – библиотека модулей коммуникации и архивации.

Библиотеки OSCAT Building и OSCAT Network основаны на OSCAT Basic и используют ее модули.

### 3. Специальные функции

#### 3.1. BUILDING\_VERSION

Тип модуля: функция	Переменная	Тип	Описание
<b>Входы</b>	IN	BOOL	Вход функции.
<b>Выходы</b>	BUILDING_VERSION	DWORD	Версия библиотеки/дата релиза.

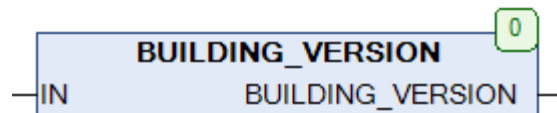


Рис. 3.1. Внешний вид функции **BUILDING\_VERSION** на языке CFC

Функция **BUILDING\_VERSION** возвращает версию библиотеки (в формате DWORD) при значении IN=FALSE и дату релиза данной версии (в формате DWORD) при значении IN=TRUE.

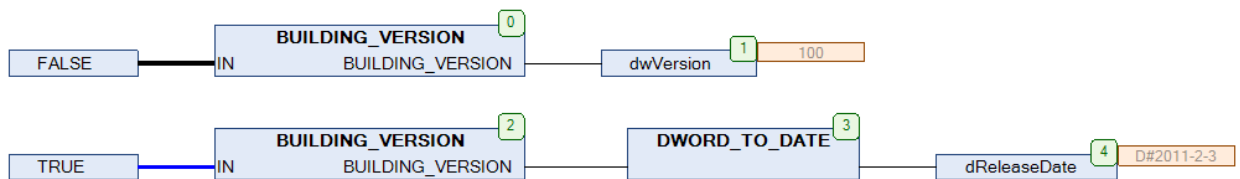


Рис. 3.2. Пример работы с функцией **BUILDING\_VERSION** на языке CFC

## 4. Исполнительные механизмы

### 4.1. ACTUATOR\_2P

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	IN	BYTE	Управляющее значение (0...255).
	TEST	BOOL	Передний фронт – активация ИМ.
	ARE	BOOL	TRUE – разрешить автоактивацию.
<b>Выходы</b>	OUT	BOOL	Сигнал управления ИМ.
	ARO	BOOL	TRUE – выполняется автоактивация.
<b>Входы-выходы</b>	ARX	BOOL	Шина синхронизации автоактивации.
<b>Параметры</b>	CYCLE_TIME	TIME	Период ШИМ.
	SENS	BYTE	Чувствительность входного сигнала.
	SELF_ACT_TIME	TIME	Допустимое время простоя.
	SELF_ACT_PULSE	TIME	Время автоактивации.
	SELF_ACT_CYCLES	INT	Число циклов автоактивации.
Используемые модули	<a href="#">AUTORUN</a> , OSCAT_BASIC.GEN_PULSE, OSCAT_BASIC.BAND_B		

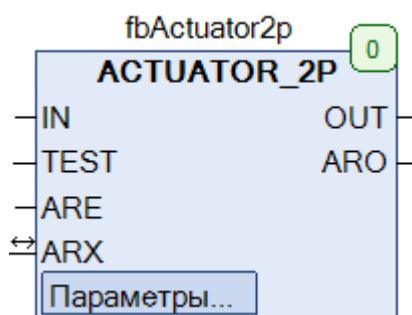


Рис. 4.1. Внешний вид ФБ **ACTUATOR\_2P** на языке CFC

**Примечание переводчика:** данный ФБ использует блок **GEN\_PULSE** из библиотеки **OSCAT Basic**. В последней релизной версии библиотеки блок работает **некорректно**. См. информацию и исправленный код в [русскоязычном описании](#) библиотеки **OSCAT Basic**.

Функциональный блок **ACTUATOR\_2P** представляет собой интерфейсный модуль для исполнительного механизма двухпозиционного регулятора (в частности – для управления двухпозиционными электромагнитными клапанами). Значение входной величины **IN** типа **BYTE** преобразуется в логический сигнал выхода **OUT**. Чтобы предотвратить залипание исполнительного механизма во время длительного простоя используется *режим автоактивации*. Для включения этого режима следует присвоить входу **ARE** значение **TRUE**, при этом выход **ARO** также примет значение **TRUE**. В режиме автоактивации по истечении времени простоя **SELF\_ACT\_TIME** клапан (т.е. выход **OUT**) открывается/закрывается **SELF\_ACT\_CYCLES** раз с периодом  $2 \cdot \text{SELF\_ACT\_PULSE}$ .

Автоактивация выполняется только в том случае, если вход-выход **ARX** имеет значение **FALSE**. Во время автоактивации **ARX** принимает значение **TRUE**. Этот вход-выход используется для организации шины синхронизации между всеми экземплярами подобных ФБ, которые работают в режиме автоактивации – чтобы в каждый момент производилась автоактивация только одного клапана. По переднему фронту входа **TEST** происходит принудительная активация блока.

Параметр **SENS** определяет чувствительность входного сигнала:

$$\left\{ \begin{array}{l} IN < SENS \Rightarrow OUT = FALSE \\ (255 - SENS) < IN \Rightarrow OUT = TRUE \end{array} \right.$$

Если **IN** принадлежит интервалу **[SENS, 255 – SENS]**, то на выходе **OUT** генерируются прямоугольные импульсы с длительностью **PTH** (верхний уровень) и **PTL** (нижний уровень), которая зависит от значения параметра **CYCLE\_TIME**:

$$\left\{ \begin{array}{l} PTH = CYCLE\_TIME \cdot \frac{IN}{255} \\ PTL = CYCLE\_TIME - PTH \end{array} \right.$$

Ниже приведен пример работы ФБ с разными значениями входа **IN**. В рамках примера для всех экземпляров ФБ используются значения **SENS=25**, **CYCLE\_TIME=T#5S**.

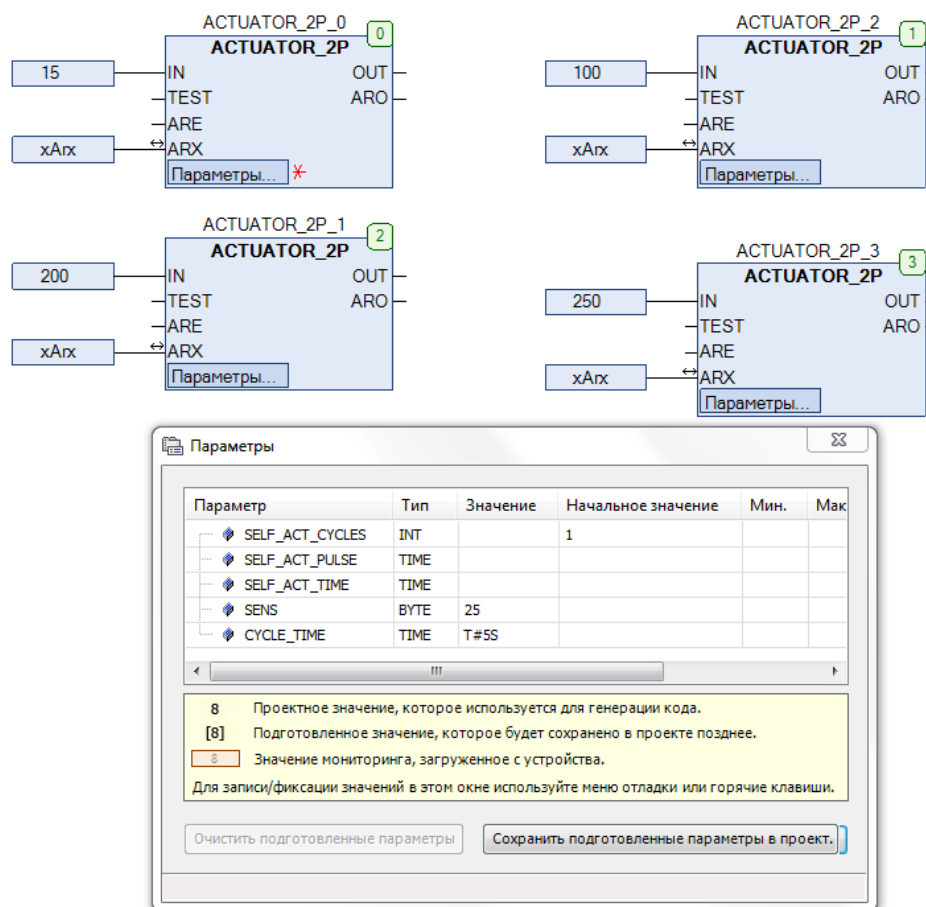


Рис. 4.2. Пример работы с ФБ **ACTUATOR\_2P** на языке CFC

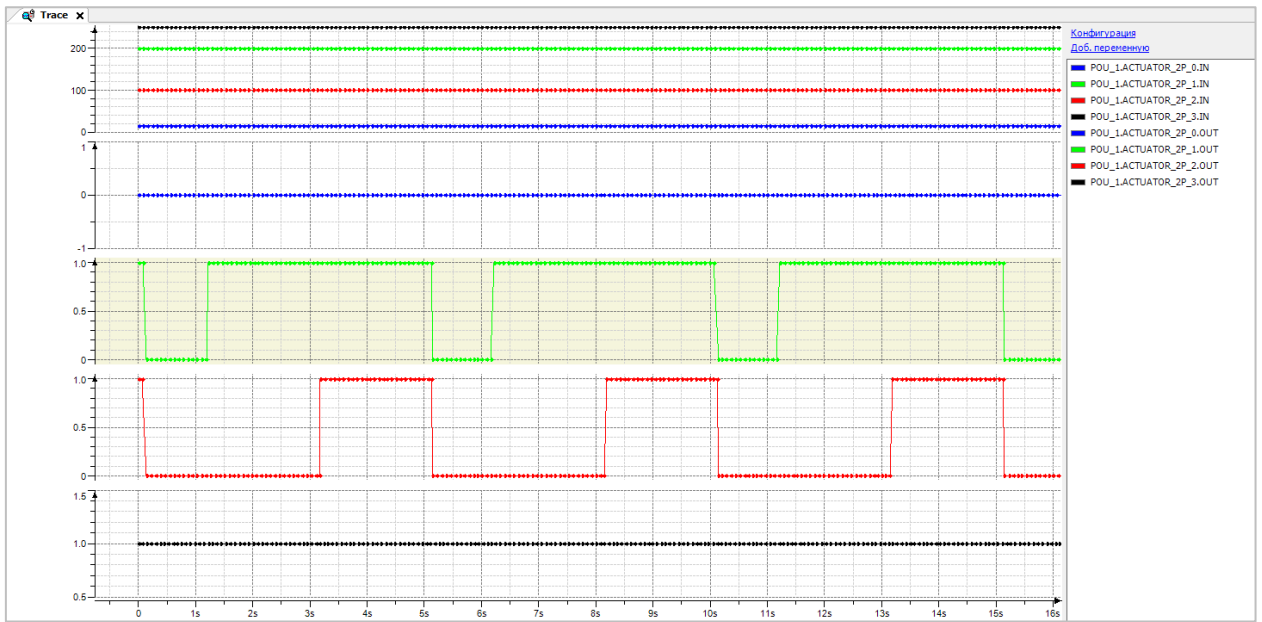
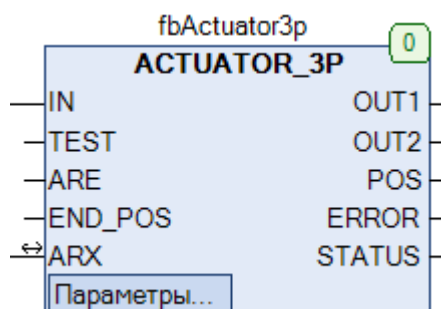


Рис. 4.3. Трассировка работы экземпляров ФБ АКТУАТОР\_2P (см. настройки на рис. 4.2)



## 4.2. ACTUATOR\_3P

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	IN	BYTE	Требуемое положение клапана (0...255).
	TEST	BOOL	Передний фронт – активация ИМ.
	ARE	BOOL	TRUE – разрешить автоактивацию.
	END_POS	BOOL	TRUE – клапан полностью открыт или полностью закрыт.
<b>Выходы</b>	OUT1	BOOL	Сигнал открытия клапана.
	OUT2	BOOL	Сигнал закрытия клапана.
	POS	BYTE	Предполагаемое положение клапана (0...255).
	ERROR	BOOL	Флаг ошибки.
	STATUS	BYTE	ESR-код.
<b>Входы-выходы</b>	ARX	BOOL	Шина синхронизации автоактивации.
<b>Параметры</b>	T_RUN	TIME	Время полного хода клапана.
	T_EXT	TIME	Дополнительное время для диагностики.
	T_CAL	TIME	Период между процедурами калибровки.
	T_DIAG	TIME	Период между процедурами диагностики.
	SWITCH_AVAIL	BOOL	TRUE – ИМ оснащен концевыми выключателями.
Используемые модули	OSCAT_BASIC.T_PLC_MS, OSCAT_BASIC._RMP_NEXT		

Рис. 4.4. Внешний вид ФБ **ACTUATOR\_3P** на языке CFC

Функциональный блок **ACTUATOR\_3P** представляет собой интерфейсный модуль для исполнительного механизма трехпозиционного регулятора (в частности – для управления трехпозиционными электромагнитными клапанами). Значение входной величины **IN** типа **BYTE** преобразуется в логические сигналы выходов **OUT1** (открытие клапана) и **OUT2** (закрытие клапана), которые управляют исполнительным механизмом. Значение **IN=0** соответствует полностью закрытому клапану, **255** – полностью открытому клапану, **127** – клапану, открытому на 50%. Параметр **T\_RUN** определяет время полного хода клапана. Если в системе имеется концевой выключатель, то его сигнал может быть заведен на вход **END\_POS**, при этом параметру **SWITCH\_AVAIL** должно быть присвоено значение **TRUE**. Концевой выключатель должен срабатывать при достижении как верхнего, так и нижнего допустимого предела. Выход **POS** содержит предполагаемое положение клапана (вычисленное на основании заданного времени хода **T\_RUN**).

По переднему фронту входа **TEST** производится принудительная самодиагностика работы клапана. В процессе диагностики клапан закрывается до 0%, затем открывается до 100%, повторно закрывается до 0%, после чего возвращается в исходное положение, определяемое значением входа **IN**. Одновременно происходит измерение времени хода клапана, что используется для повышения точности позиционирования. В случае использования концевого выключателя (**SWITCH\_AVAIL=TRUE**) также проверяется корректность его работы. Если вход **ARE** имеет значение **TRUE**, то диагностика выполняется автоматически с периодом **T\_DIAG**.

Диагностика выполняется только в том случае, если вход-выход **ARX** имеет значение **FALSE**. Во время диагностики **ARX** принимает значение **TRUE**. Этот вход-выход используется для организации шины синхронизации между всеми экземплярами подобных ФБ – чтобы в каждый момент времени производилась диагностика только одного клапана.

Если во время диагностики произошла ошибка, то выход **ERROR** принимает значение **TRUE**.

В процессе регулирования клапан перемещается с определенной погрешностью, что приводит к накоплению ошибки позиционирования. Параметр **T\_CAL** позволяет определить период автоматической калибровки. В процессе калибровки клапан закрывается до 0%, после чего возвращается в исходное положение, определяемое значением входа **IN**. Измеренное время хода используется для повышения точности позиционирования.

Во время калибровки/диагностики без концевого выключателя (**SWITCH\_AVAIL=FALSE**) время полного хода увеличивается на величину параметра **T\_EXT**.

Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**:

Значение выхода STATUS	Описание	Значение ARE	Значение ARX
100	Нормальный режим работы	любое	любое
101	Калибровка	любое	любое
103	Диагностика (открытие клапана)	TRUE	TRUE
104	Диагностика (закрытие клапана)	TRUE	TRUE

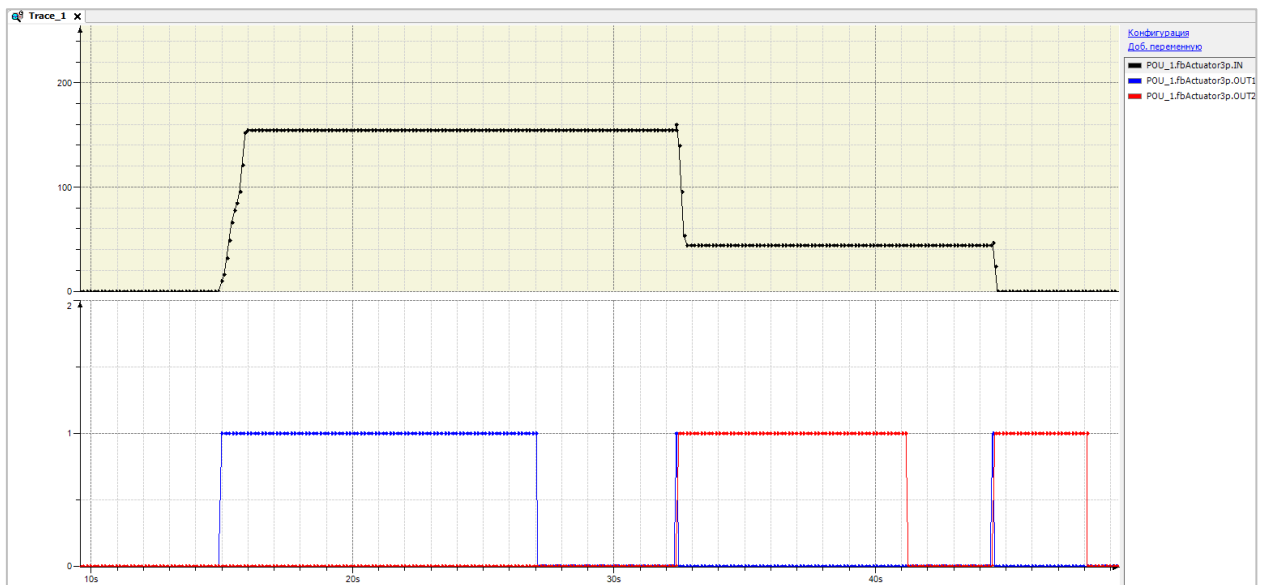
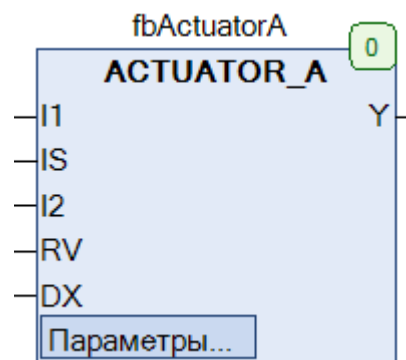


Рис. 4.6. Трассировка работы ФБ АКТУАТОР\_3Р ( $T_{\text{RUN}}=T\#20\text{s}$ )

## 4.3. ACTUATOR\_A

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	I1	BYTE	Сигнал задания 1.
	IS	BOOL	Выбор задания: FALSE – I1, TRUE – I2.
	I2	BYTE	Сигнал задания 2.
	RV	BOOL	TRUE – реверс направления.
	DX	BOOL	Сигнал принудительной активации.
<b>Выходы</b>	Y	WORD	Сигнал управления сервоприводом.
<b>Параметры</b>	RUNTIME	TIME	Время до полного открытия/закрытия регулирующего органа при автоактивации.
	SELF_ACT_TIME	TIME	Допустимое время простоя.
	OUT_MIN	WORD	Минимальное значение выхода.
	OUT_MAX	WORD	Максимальное значение выхода.
Используемые модули	OSCAT_BASIC.CYCLE_4		

Рис. 4.7. Внешний вид ФБ **ACTUATOR\_A** на языке CFC

**Примечание переводчика:** блок имеет несколько известных ошибок. На [форуме OSCAT](#) есть исходный код нескольких исправленных версий: [тема 1](#), [тема 2](#), [тема 3](#).

Функциональный блок **ACTUATOR\_A** представляет собой интерфейсный модуль для управления исполнительными механизмами с аналоговым входом (например, сервоприводами). ФБ имеет два сигнала задания: **I1** и **I2**. Это удобно при наличии различных режимов управления (ручной/автоматический). Вход **IS** определяет используемый сигнал: **FALSE** – используется **I1**, **TRUE** – используется **I2**. Параметры **OUT\_MIN** и **OUT\_MAX** определяют минимальное и максимальное значение управляющего выхода **Y**. Вход **RV** определяет направление вращения сервопривода:

$$\left\{ \begin{array}{l} RV = \text{FALSE} \Rightarrow Y(Ix = 0) = \text{OUT\_MIN}, Y(Ix = 255) = \text{OUT\_MAX} \\ RV = \text{TRUE} \Rightarrow Y(Ix = 0) = \text{OUT\_MAX}, Y(Ix = 255) = \text{OUT\_MIN} \end{array} \right.$$

Значение **Y** может быть напрямую заведено на сервоприводы со встроенным 16-битным ЦАП. Управляющий сигнал рассчитывается по формуле, зависящей от значений **IS** и **RV**:

Значение IS	Значение RV	Значение Y
FALSE	FALSE	Значение <b>Y</b> изменяется пропорционально <b>I1</b> в диапазоне от <b>OUT_MIN</b> до <b>OUT_MAX</b> : $Y = \frac{(OUT\_MAX - OUT\_MIN) \cdot I1}{255} + OUT\_MIN$
TRUE	FALSE	Значение <b>Y</b> изменяется пропорционально <b>I2</b> в диапазоне от <b>OUT_MIN</b> до <b>OUT_MAX</b> : $Y = \frac{(OUT\_MAX - OUT\_MIN) \cdot I2}{255} + OUT\_MIN$
FALSE	TRUE	Значение <b>Y</b> изменяется пропорционально <b>I1</b> в диапазоне от <b>OUT_MAX</b> ( <b>I1=0</b> ) до <b>OUT_MIN</b> ( <b>I1=255</b> ): $Y = OUT\_MAX - \frac{(OUT\_MAX - OUT\_MIN) \cdot I1}{255}$
TRUE	TRUE	Значение <b>Y</b> изменяется пропорционально <b>I2</b> в диапазоне от <b>OUT_MAX</b> ( <b>I2=0</b> ) до <b>OUT_MIN</b> ( <b>I2=255</b> ): $Y = OUT\_MAX - \frac{(OUT\_MAX - OUT\_MIN) \cdot I2}{255}$

Параметр **SELF\_ACT\_TIME** определяет допустимое время простоя регулирующего органа. По истечении этого времени значение выхода **Y** за время **RUNTIME** меняется до **OUT\_MIN**, затем за время **RUNTIME** возрастает до **OUT\_MAX**, после чего возвращается к исходному значению. По переднему фронту входа **DX** эта процедура выполняется принудительно.

## 4.4. ACTUATOR\_COIL

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	IN	BOOL	Требуемое положение клапана.
<b>Выходы</b>	OUT	BOOL	Сигнал управления клапаном.
	STATUS	BYTE	ESR-код.
<b>Параметры</b>	SELF_ACT_CYCLE	TIME	Допустимое время простоя.
	SELF_ACT_TIME	TIME	Время автоактивации.
Используемые модули	OSCAT_BASIC.T_PLC_MS		

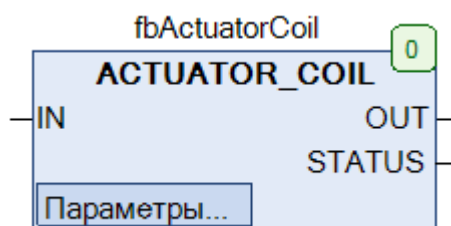


Рис. 4.8. Внешний вид ФБ ACTUATOR\_COIL на языке CFC

Функциональный блок **ACTUATOR\_COIL** представляет собой интерфейсный модуль для управления исполнительными механизмами электромагнитных клапанов. Значение входа **IN** подается на выход **OUT**. Чтобы предотвратить залипание исполнительного механизма во время длительного простоя используется *автоактивация* блока – по истечении времени простоя **SELF\_ACT\_CYCLE** выход **OUT** включается на время **SELF\_ACT\_TIME**.

Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**:

Значение выхода Status	Описание
100	Клапан закрыт.
101	Клапан открыт заданием по входу <b>IN</b> .
102	Клапан открыт в режиме автоактивации.

## 4.5. ACTUATOR\_PUMP

Тип модуля: ФБ	Переменная	Тип	Описание
Входы	IN	BOOL	TRUE – включить насос (сигнал из системы управления).
	MANUAL	BOOL	TRUE – включить насос принудительно
	RST	BOOL	Передний фронт – сброс статистики.
Выходы	PUMP	BOOL	Сигнал включения насоса.
Входы-выходы	RUNTIME	UDINT	Время наработки (в секундах).
	CYCLES	UDINT	Количество включений насоса.
Параметры	MIN_ONTIME	TIME	Минимальное время работы.
	MIN_OFFTIME	TIME	Минимальная пауза (задержка) между включениями.
	RUN_EVERY	TIME	Допустимое время простоя, по истечении которого насос будет запущен на время MIN_ONTIME.
Используемые модули	OSCAT_BASIC.T_PLC_MS, OSCAT_BASIC.ONTIME		

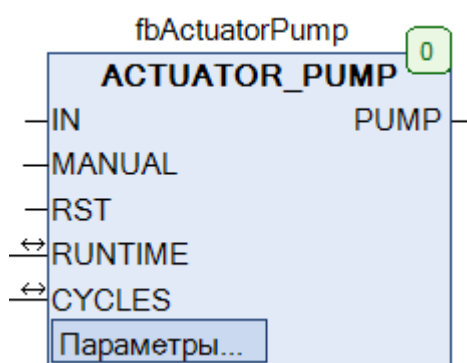


Рис. 4.9. Внешний вид ФБ ACTUATOR\_PUMP на языке CFC

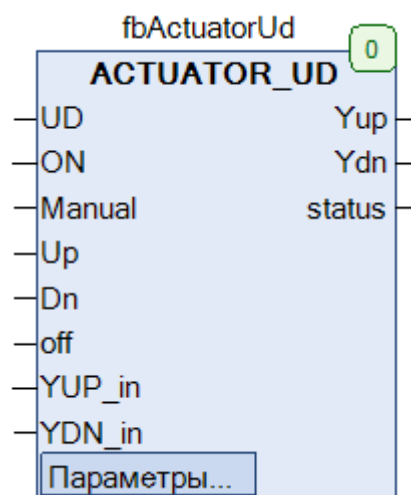
Функциональный блок **ACTUATOR\_PUMP** представляет собой интерфейсный модуль для управления исполнительными механизмами насосов с счетчиком часов наработки. Выход **PUMP** управляется входами **IN** и **MANUAL** (**MANUAL** имеет приоритет над **IN**). Если выход **PUMP** активирован входом **IN**, то он остается в значении **TRUE** до истечения времени **MIN\_ONTIME** (даже если вход **IN** в течение этого времени будет сброшен в **FALSE**). Параметр **MIN\_OFFTIME** определяет задержку между последовательными включениями насоса, в течение которой выход **PUMP** остается в состоянии **FALSE**.

Чтобы предотвратить залипание исполнительного механизма во время длительного простоя используется *автоактивация* блока – по истечении времени простоя **RUN\_EVERY** выход **PUMP** включается на время **MIN\_ONTIME**.

Входы-выходы **RUNTIME** и **CYCLES** содержат время наработки (в секундах) и количество включений насоса соответственно. Эти значения могут быть сброшены в 0 по переднему фронту на входе **RST**.

## 4.6. ACTUATOR\_UD

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	UD	BOOL	Направление в автоматическом режиме (TRUE – UP).
	ON	BOOL	TRUE – выбор автоматического режима работы.
	MANUAL	BOOL	TRUE – выбор ручного режима работы (приоритетнее автоматического).
	UP	BOOL	Требуемое состояние реле «вверх» в ручном режиме работы.
	DN	BOOL	Требуемое состояние реле «вниз» в ручном режиме работы.
	OFF	BOOL	Передний фронт – отключить выход.
	YUP_IN	BOOL	Обратная связь от реле «вверх»: TRUE - реле сработало.
	YDN_IN	BOOL	Обратная связь от реле «вниз»: TRUE - реле сработало.
<b>Выходы</b>	YUP	BOOL	Сигнал управления реле «вверх».
	YDN	BOOL	Сигнал управления реле «вниз».
	STATUS	BYTE	ESR-код.
<b>Параметры</b>	TON	TIME	Минимальное время работы.
	TOFF	TIME	Минимальная пауза (задержка) между включениями.
	OUT_RETURN	BOOL	TRUE – сигналы обратной связи контролируются ФБ, FALSE – не контролируются.
Используемые модули	OSCAT_BASIC.T_PLC_MS		

Рис. 4.10. Внешний вид ФБ **ACTUATOR\_UD** на языке CFC



Функциональный блок **ACTUATOR\_UD** представляет собой интерфейсный модуль для управления реверсивным пускателем. В значительном числе случаев реверсивный пускатель используется для управления направлением вращения электродвигателя.

Блок может работать в двух режимах: ручном и автоматическом (ручное управление имеет приоритет над автоматическим). В ручном режиме управления (**MANUAL=TRUE**) значения выхода **YUP** определяется входом **UP**, а выхода **YDN** – входом **DN**. В каждый момент времени должен быть активен только один из входов. Если оба входа (**UP** и **DN**) имеют значение **TRUE**, то оба выхода (**YUP** и **YDN**) принимают значение **FALSE**.

В автоматическом режиме (**MANUAL=FALSE, ON=TRUE**) значения выходов **YUP** и **YDN** определяются значением входа **UD**:

<b>UD</b>	<b>YUP</b>	<b>YDN</b>
FALSE	FALSE	TRUE
TRUE	TRUE	FALSE

Если вход **OFF** имеет значение **TRUE**, то выходы **YUP** и **YDN** принимают значение **FALSE** независимо от режима управления.

Если параметр **OUT\_RETURN** имеет значение **TRUE**, то блок работает в режиме контроля состояния реле. В этом режиме сигналы от реле, которыми управляют выходы **YUP** и **YDN**, должны быть заведены на входы блока **YUP\_IN** и **YDN\_IN**. В случае залипания одного из реле ФБ не даст команду на включение другому.

Параметры **TON** и **TOFF** определяют минимальное время включения реле и минимальное время между последовательными включениями. Это полезно при управлении мощными двигателями, которым требуется определенное время для разгона и торможения.

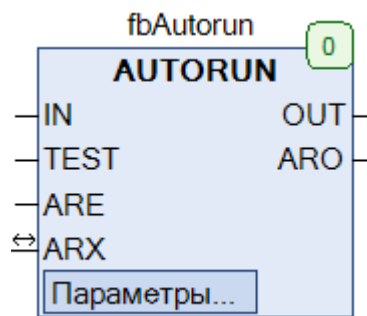
Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**:

<b>MANUAL</b>	<b>UP</b>	<b>DN</b>	<b>ON</b>	<b>UD</b>	<b>OFF</b>	<b>YUP</b>	<b>YDN</b>	<b>STATUS</b>
TRUE	FALSE	FALSE	- <sup>1</sup>	-	FALSE	FALSE	FALSE	102
TRUE	TRUE	FALSE	-	-	FALSE	TRUE	FALSE	103
TRUE	FALSE	TRUE	-	-	FALSE	FALSE	TRUE	104
TRUE	TRUE	TRUE	-	-	FALSE	FALSE	FALSE	102
FALSE	-	-	TRUE	TRUE	FALSE	TRUE	FALSE	111
FALSE	-	-	TRUE	FALSE	FALSE	FALSE	TRUE	112
-	-	-	-	-	TRUE	FALSE		101
TRUE	FALSE	FALSE	FALSE	-	FALSE	FALSE		110
FALSE	-	-	FALSE	-	-	FALSE		110

<sup>1</sup> Символ означает, что состояние входа не имеет значения (прим. пер.)

## 4.7. AUTORUN

Тип модуля: ФБ	Переменная	Тип	Описание
Входы	IN	BYTE	Требуемое состояние выхода OUT.
	TEST	BOOL	Передний фронт – активация ИМ.
	ARE	BOOL	TRUE – разрешить автоактивацию.
Выходы	OUT	BOOL	Сигнал управления ИМ.
	ARO	BOOL	TRUE – выполняется автоактивация.
Входы-выходы	ARX	BOOL	Шина синхронизации автоактивации.
Параметры	TRUN	TIME	Время активации выхода.
	TOFF	TIME	Допустимое время простоя.
Используемые модули	OSCAT_BASIC._RMP_B		

Рис. 4.11. Внешний вид ФБ **AUTORUN** на языке CFC

Функциональный блок **AUTORUN** представляет собой интерфейсный модуль для управления дискретным выходом с режимом автоактивации. В обычном режиме работы значение входа **IN** передается на выход **OUT**.

Для включения режима автоактивации следует присвоить входу **ARE (Autorun Enable)** значение **TRUE**, при этом выход **ARO** также примет значение **TRUE**. В режиме автоактивации по истечении времени простоя **TOFF** выход **OUT** включается на время **TRUN**. Автоактивация выполняется только в том случае, если вход-выход **ARX** имеет значение **FALSE**. Во время автоактивации **ARX** принимает значение **TRUE**. По переднему фронту входа **TEST** происходит принудительная активация блока.

Вход-выход **ARX** может использоваться для двух целей:

- для управление режимом автоактивации (через внешний POU или другое устройство системы управления);
- для синхронизации работы нескольких блоков. Например, ФБ **AUTORUN** используется в ФБ [ACTUATOR\\_2P](#), который предназначен для управления исполнительными механизмами клапанов. ФБ **AUTORUN** позволяет периодически менять состояние клапана, чтобы предотвратить его залипание. Но в целях безопасности в каждый момент времени только один из клапанов должен находиться в режиме автоактивации. Для этого все экземпляры ФБ [ACTUATOR\\_2P](#) соединяются через вход-выход **ARX**.

Пример использования входа-выхода ARX для управления автоактивацией через ФБ **TIMER\_1** (активация обоих экземпляров происходит одновременно):

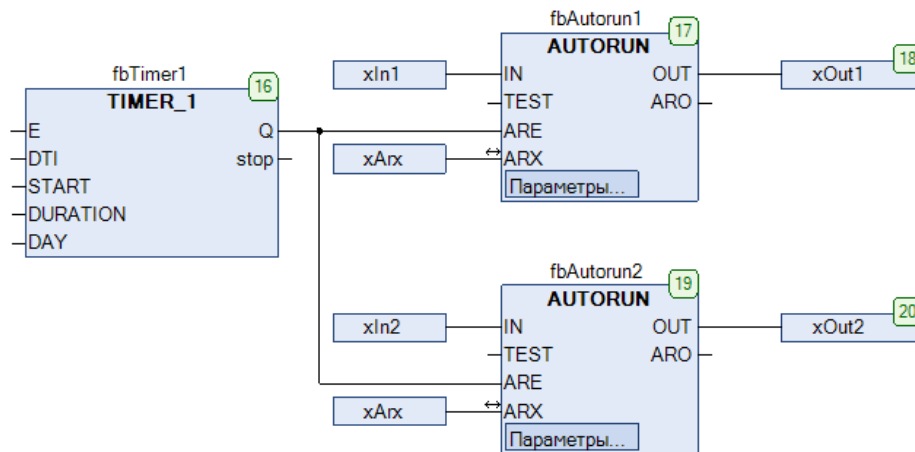


Рис. 4.12. Пример работы с ФБ **AUTORUN** на языке CFC (1)

Пример использования входа-выхода **ARX** для синхронизации автоактивации нескольких экземпляров ФБ (активация одного из экземпляров происходит только тогда, когда остальные являются неактивными):

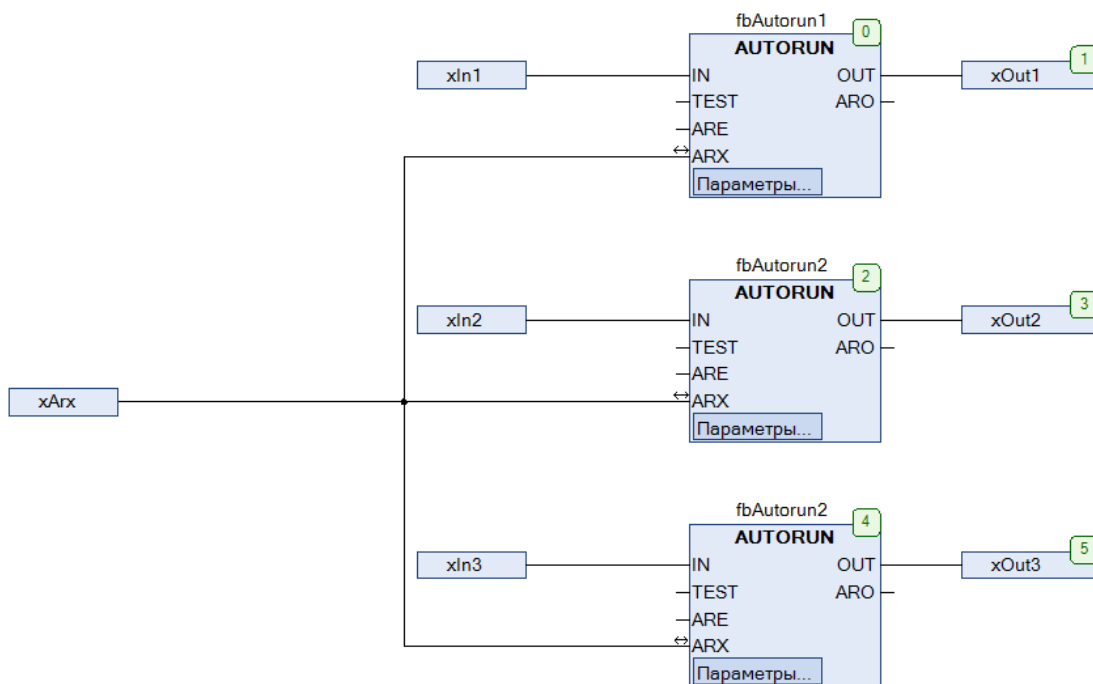


Рис. 4.13. Пример работы с ФБ **AUTORUN** на языке CFC (2)

Если у нескольких блоков одновременно истекает интервал бездействия, то автоактивация выполнится у блока, который вызывается первым. После завершения его автоактивации произойдет автоактивация следующего блока и т.д.

## 5. HVAC

### 5.1. AIR\_DENSITY

Тип модуля: функция	Переменная	Тип	Описание
Входы	T	REAL	Температура, °С.
	P	REAL	Давление, Па.
	RH	REAL	Относительная влажность, %.
Выходы	AIR_DENSITY	REAL	Плотность воздуха, кг/м <sup>3</sup> .
Используемые модули	<a href="#">SDD</a> , OSCAT_BASIC.PHYS		

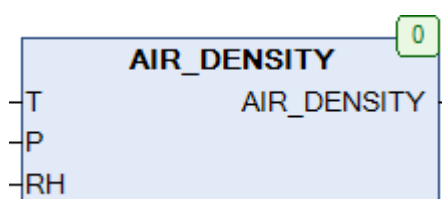


Рис. 5.1. Внешний вид функции **AIR\_DENSITY** на языке CFC

Функция **AIR\_DENSITY** возвращает [плотность воздуха](#) для заданной температуры **T**, давления **P** и относительной влажности **RH**. Функция возвращает корректные значения для  $0 \leq T \leq 100$ .

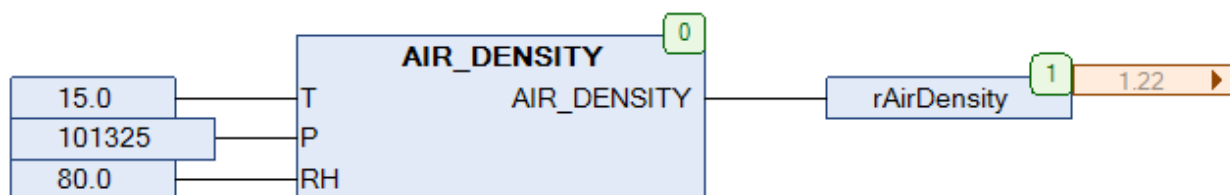


Рис. 5.2. Пример работы с функцией **AIR\_DENSITY** на языке CFC

Для тестирования функции можно воспользоваться [онлайн-калькулятором физических свойств влажного воздуха](#).

## 5.2. AIR\_ENTHALPY

Тип модуля: функция	Переменная	Тип	Описание
Входы	T	REAL	Температура, °С.
	RH	REAL	Относительная влажность, %.
Выходы	AIR_ENTHALPY	REAL	Энтальпия воздуха, кДж/кг.
Используемые модули	<a href="#">DEW_CON</a>		

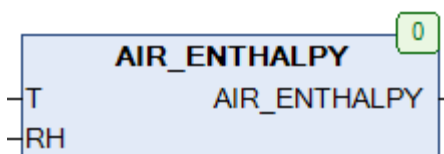


Рис. 5.3. Внешний вид функции **AIR\_ENTHALPY** на языке CFC

**Примечание переводчика:** в текущей версии библиотеки функция работает **некорректно** ([пруф](#)). Для корректной работы необходимо отредактировать код функции следующим образом<sup>2</sup>:

```

1  FUNCTION AIR_ENTHALPY : REAL
2  VAR_INPUT
3      T:      REAL;
4      RH:     REAL;
5  END_VAR
6  VAR_CONSTANT
7      CPL: REAL := 1.00482;
8      CW:  REAL := 1.86;
9      LH:  REAL := 2500.78;
10     C_100: REAL := 100;
11     P:   REAL := 1013; // атм. давление в гектопаскалях
12 END_VAR
13 VAR
14     PS: REAL;
15     Pact: REAL;
16     G:  REAL;
17 END_VAR
18
19 PS := 610.78 * EXP( ( 17.2694 * T ) / (T + 238.3)); (*computation of saturation vapour pressure*)
20 Pact := PS * (RH / 100.0); (*computation of actual vapour pressure*)
21 G := 0.62 * ( Pact / ( ( P * C_100 ) - Pact ) ); (*computation of water content*)
22
23 AIR_ENTHALPY := CPL * T + ( LH * G + CW * G * T );

```

Рис. 5.4. Исправление исходного кода функции **AIR\_ENTHALPY** для обеспечения корректной работы ([скачать в формате PLCopen XML](#))

Функция **AIR\_ENTHALPY** возвращает энтальпию воздуха при нормальном атмосферном давлении для заданной температуры **T** и относительной влажности **RH**. Функция возвращает корректные значения для  $0 \leq T \leq 100$ .

<sup>2</sup> Использован фрагмент кода из библиотеки BUILDING\_HVAC\_03 компании WAGO.

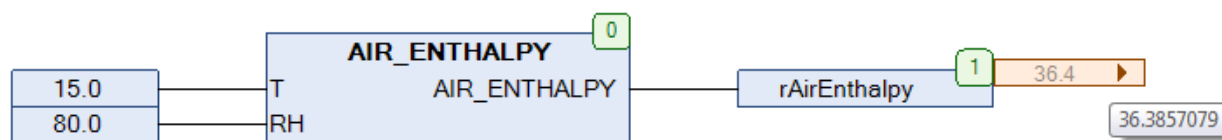
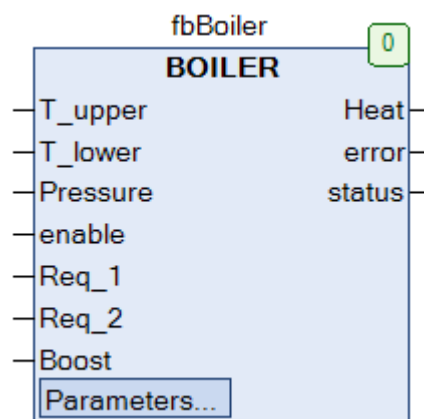


Рис. 5.5. Пример работы с функцией **AIR\_ENTHALPY** на языке CFC

Для тестирования функции можно воспользоваться [онлайн-калькулятором физических свойств влажного воздуха](#).

## 5.3. BOILER

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	T_upper	REAL	Значение температуры в нижней части бойлера.
	T_lower	REAL	Значение температуры в верхней части бойлера.
	Pressure	BOOL	TRUE – недостаточное давление в бойлере.
	enable	BOOL	Сигнал включения бойлера.
	Req_1	BOOL	TRUE – текущая уставка T_request_1.
	Req_2	BOOL	TRUE – текущая уставка T_request_2.
	Boost	BOOL	Принудительное включение нагрева.
<b>Выходы</b>	Heat	BOOL	Сигнал управления нагревателем.
	error	BOOL	Флаг ошибки.
	status	BYTE	ESR-код.
<b>Параметры</b>	T_upper_min	REAL	Минимальная требуемая температура в верхней части бойлера.
	T_upper_max	REAL	Максимальная требуемая температура в верхней части бойлера.
	T_lower_enable	BOOL	Включить обработку нижнего датчика температуры.
	T_lower_max	REAL	Максимальная требуемая температура в нижней части бойлера.
	T_request_1	REAL	Уставка 1.
	T_request_2	REAL	Уставка 2.
	T_request_hyst	REAL	Значение гистерезиса.
	T_protect_high	REAL	Верхний предел допустимой температуры.
T_protect_low	REAL	Нижний предел допустимой температуры.	

Рис. 5.6. Внешний вид ФБ **BOILER** на языке CFC

Функциональный блок **BOILER** предназначен для управления водонагревающими устройствами (в частности, бойлерами). На входы **T\_upper** и **T\_lower** заводятся сигналы датчиков температуры, размещенных в верхней и нижней части бойлера. Наличие датчика температуры в верхней части бойлера является обязательным, в нижней – опциональным. Если датчик температуры в нижней части бойлера отсутствует, то следует присвоить входу **T\_lower\_enable** значение **FALSE**.

Если вход **enable=TRUE** и **T\_lower\_enable=TRUE**, то выход **HEAT**, который управляет нагревательным элементом, имеет значение **TRUE** до тех пор, пока температура в нижней части бойлера не достигнет **T\_lower\_max**. После этого выход **Heat** примет значение **FALSE**. Если значение температуры в верхней части бойлера упадет ниже **T\_upper\_min**, то нагревательный элемент будет включен (**Heat=TRUE**). Если вход **enable=TRUE** и **T\_lower\_enable=FALSE**, то значение входа **T\_lower** не обрабатывается. Управление подогревом происходит на основании значения входа **T\_upper** и параметров **T\_upper\_max/ T\_upper\_min** по аналогичному принципу.

Верхний и нижний пределы допустимой температуры бойлера определяются параметрами **T\_protect\_high** и **T\_protect\_low**. В случае превышения верхнего предела подогрев будет отключен, в случае выхода за нижний предел – включен.

Вход **Pressure** используется для подключения сигнала датчика контроля давления воды. При недостаточном давлении сигнал должен иметь значение **FALSE**, что запретит включение подогрева и предотвратит повреждение бойлера.

По переднему фронту на входе **Boost** происходит нагрев бойлера до температуры **T\_upper\_max** (если **T\_lower\_enable=FALSE**) или **T\_lower\_max** (если **T\_lower\_enable=TRUE**). Это позволяет обеспечить максимальную тепловую нагрузку.

Входы **Req\_1** и **Req\_2** могут использоваться для нагрева до уставок **T\_request\_1** и **T\_request\_2**. Это может применяться в профилактических целях (например, для дезинфекции [легионеллы](#) в трубах). Уставки сравниваются с температурой датчика верхней части бойлера и управляют выходом **Heat** в режиме двухпозиционного регулятора с гистерезисом **T\_request\_hyst**. Обработка первой уставки имеет приоритет по сравнению со второй.

Если в процессе работы ФБ произошла ошибка, то выход **error** принимает значение **TRUE**.

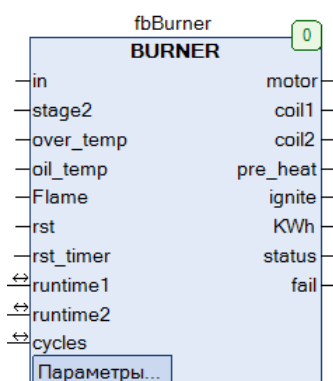
Выход **status** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**:

Значение выхода status	Описание
1	Значение T_upper > T_upper_max
2	Значение T_upper < T_upper_min
3	Значение T_lower > T_lower_max
4	Значение T_lower < T_lower_min
5	Давление воды в бойлере слишком низкое для нагрева
100	Режим ожидания
101	Бойлер в работе (по переднему фронту Boost)
102	Бойлер в работе (Enable=TRUE)
103	Бойлер в работе (Req_1)
104	Бойлер в работе (Req_2)



## 5.4. BURNER

Тип модуля: ФБ	Переменная	Тип	Описание
Входы	in	BOOL	Разрешение на работу блока.
	stage2	BOOL	Разрешение на работу 2-й ступени.
	over_temp	BOOL	TRUE – превышена температура в бойлере.
	oil_temp	BOOL	TRUE – превышена температура топлива.
	Flame	BOOL	TRUE – есть пламя в топке
	rst	BOOL	Передний фронт – перезапуск котла после аварии.
	rst_timer	BOOL	Передний фронт – сброс счетчиков наработки.
Выходы	motor	BOOL	TRUE – включить вентилятор и дымосос.
	coil1	BOOL	Управление топливным клапаном 1-й ступени.
	coil2	BOOL	Управление топливным клапаном 2-й ступени.
	pre_heat	BOOL	TRUE – прогрев топлива
	ignite	BOOL	TRUE – розжиг
	KWh	REAL	Выработанная тепловая энергия в кВт·ч.
	status	BYTE	ESR-код.
	fail	BOOL	Флаг ошибки.
Входы-выходы	runtime1	UDINT	Время наработки 1-й ступени (в секундах).
	runtime2	UDINT	Время наработки 2-й ступени (в секундах).
	cycles	UDINT	Число включений горелки.
Параметры	pre_heat_time	TIME	Время прогрева топливной магистрали.
	pre_vent_time	TIME	Время продувки.
	pre_ignite_time	TIME	Время розжига до включения клапана 1-й ступени.
	post_ignite_time	TIME	Время розжига после включения клапана 1-й ступени.
	stage2_delay	TIME	Время задержки перед открытием топливного клапана 2-й ступени.
	safety_time	TIME	Допустимое время розжига.
	lockout_time	TIME	Время, в течение которого ошибка не может быть квитирована.
	multiple_ignition	BOOL	Режим повторения попыток розжига.
	kw1	REAL	Мощность 1-й ступени горелки.
	kw2	REAL	Мощность 2-й ступени горелки.
Используемые модули	OSCAT_BASIC.T_PLC_MS, OSCAT_BASIC.ONTIME		

Рис. 5.7. Внешний вид ФБ **BURNER** на языке CFC

Функциональный блок **BURNER** представляет собой интерфейсный модуль для управления двухступенчатой жидкотопливной горелкой, регулирующей температуру котла. На вход **IN** подается управляющий сигнал, который запускает цикл работы горелки. Это происходит только в том случае, если температура котловой воды находится в допустимых пределах (**over\_temp=FALSE**).

Цикл работы горелки начинается с прогрева топливной магистрали: выход **pre\_heat** принимает значение **TRUE** на время **pre\_heat\_time**. В течение этого времени на вход **oil\_temp** должно быть подано значение **TRUE**, что сигнализирует о достижении нужной температуры топлива. Если этого не произойдет, то выход **fail** примет значение **TRUE** и работа горелки будет остановлена.

После прогрева топлива выход **motor** принимает значение **TRUE**, что приводит к переходу горелки в режим продувки на время **pre\_vent\_time**. Через заданное время (**pre\_vent\_time – pre\_ignite\_time**) начинается розжиг горелки (**ignite=TRUE**) и по истечении времени **pre\_ignite\_time** происходит открытие топливного клапана первой ступени (**coil1=TRUE**). По истечении времени **post\_ignite\_time** выход **ignite** принимает значение **FALSE**. Если в течение времени **safety\_time** сигнал датчика пламени **Flame** не принимает значение **TRUE**, то выход **fail** примет значение **TRUE** и работа горелки будет остановлена. Но если параметр **multiple\_ignition** имеет значение **TRUE**, то попытки розжига будут предприниматься до тех пор, пока факел не разгорится. Если вход **stage2** имеет значение **TRUE**, то через время **stage2\_delay** от момента появления факела (**FLAME=TRUE**) будет открыт топливный клапан второй ступени (**coil2=TRUE**).

Если температура котловой воды превышает допустимую границу (**over\_temp=TRUE**), то выход **fail** принимает значение **TRUE** и работа горелки останавливается. В случае возникновения ошибки (**fail=TRUE**) цикл работы горелки можно перезапустить по переднему фронту на входе **rst**.

Входы-выходы **runtime1** и **runtime2** содержат время наработки для первой и второй ступени горелки (в секундах). Вход-выход **cycles** содержит число запусков горелки. Параметры **kw1** и **kw2** определяют мощность первой и второй ступени горелки (в кВт). Выход **KWh** содержит значение произведенной энергии (в кВт·ч). По переднему фронту входа **rst\_timer** выполняется сброс счетчиков наработки и запусков горелки.

Выход **status** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**:

Значение выхода status	Описание
1	Ошибка при прогреве топливной магистрали.
2	Сигнал от датчика пламени во время прогрева топлива.
3	Сигнал от датчика пламени во время продувки.
4	Отсутствие сигнала от датчика пламени после розжига.
5	Пропадание сигнала от датчика пламени в процессе работы.
9	Превышена температура котловой воды.
110	Режим ожидания.
111	Запущен цикл работы горелки.
112	Работа первой ступени горелки.
113	Работа второй ступени горелки.

Циклограмма работы блока:

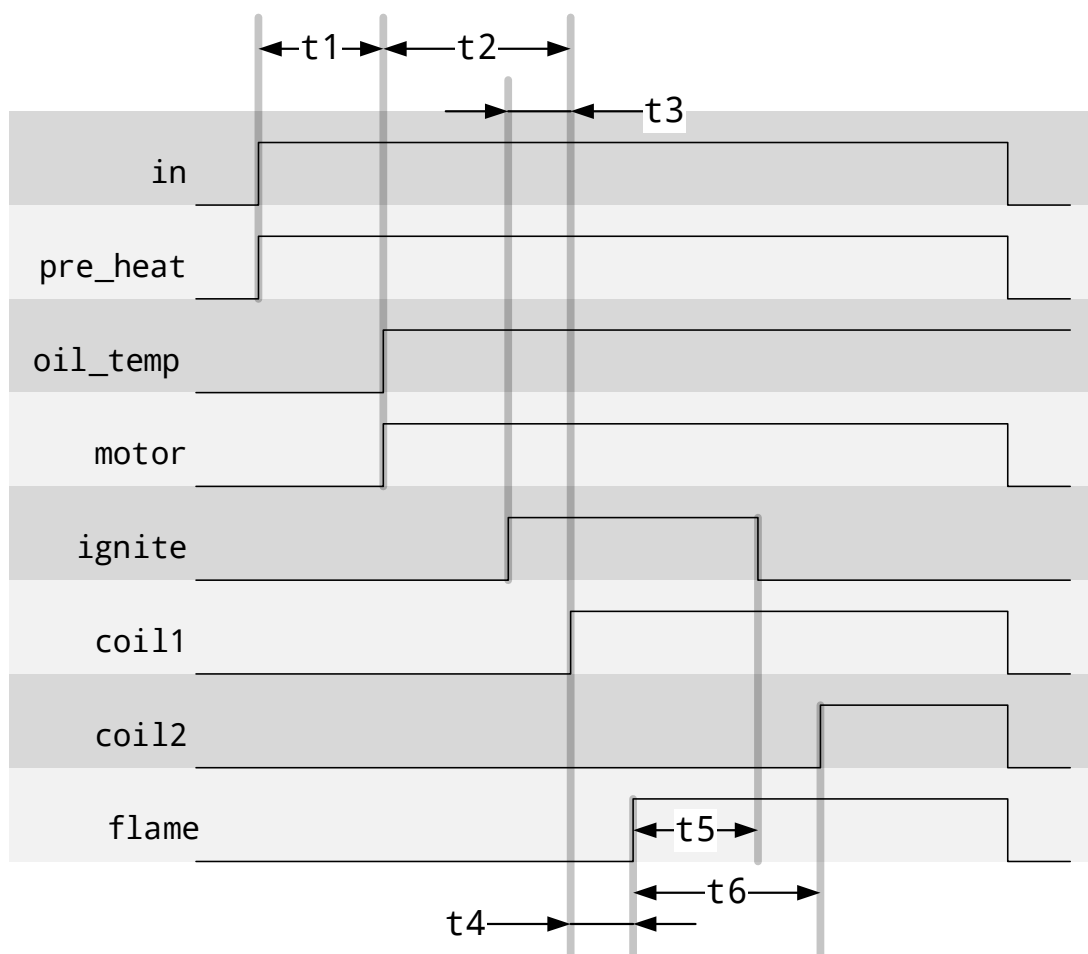


Рис. 5.8. Циклограмма работы ФБ **BURNER**

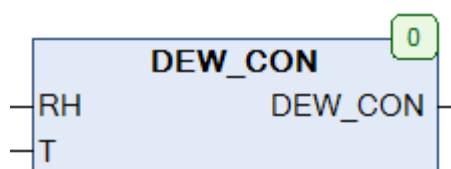
- $t_1 = \text{pre\_heat\_time}$
- $t_2 = \text{pre\_vent\_time}$
- $t_3 = \text{pre\_ignite\_time}$
- $t_4 = \text{safety\_time}$
- $t_5 = \text{post\_ignite\_time}$
- $t_6 = \text{stage2\_delay}$

Таблица зависимости основных входов и выходов:

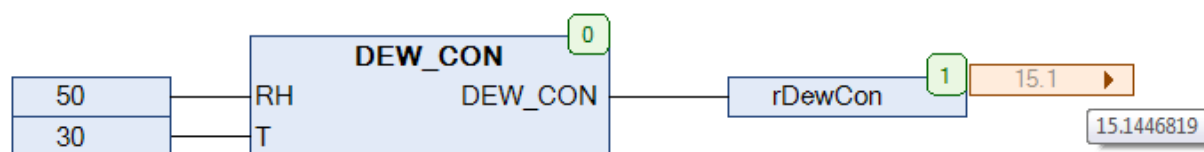
in	over_temp	Oil_temp	flame	rst	motor	coil1	pre_heat	ignite	status	fail
FALSE	FALSE	-	-	FALSE	FALSE	FALSE	FALSE	FALSE	110	FALSE
TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	111	FALSE
TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	111	FALSE
TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	111	FALSE
TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	111	FALSE
TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	112	FALSE
TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	112	FALSE
TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	111	FALSE
-	TRUE	-	-	-	-	-	-	-	9	TRUE
TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	3	TRUE

## 5.5. DEW\_CON

Тип модуля: функция	Переменная	Тип	Описание
Входы	RH	REAL	Относительная влажность, %.
	T	REAL	Температура, °C.
Выходы	DEW_CON	REAL	Абсолютная влажность воздуха, г/м <sup>3</sup> .
Используемые модули	<a href="#">SDD</a> , OSCAT_BASIC.PHYS		

Рис. 5.9. Внешний вид функции **DEW\_CON** на языке CFC

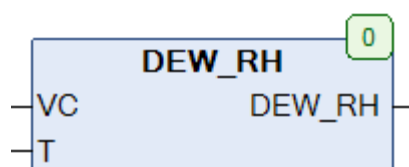
Функция **DEW\_CON** возвращает абсолютную влажность воздуха для заданной температуры **T** и относительной влажности **RH**. Функция возвращает корректные значения для  $-40 \leq T \leq 90$ .

Рис. 5.10. Пример работы с функцией **DEW\_CON** на языке CFC

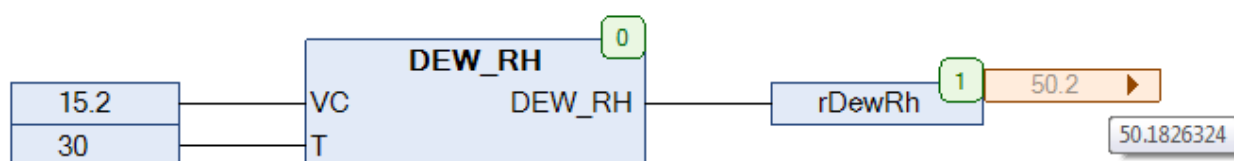
Для тестирования функции можно воспользоваться [онлайн-калькулятором влажности воздуха](#).

## 5.6. DEW\_RH

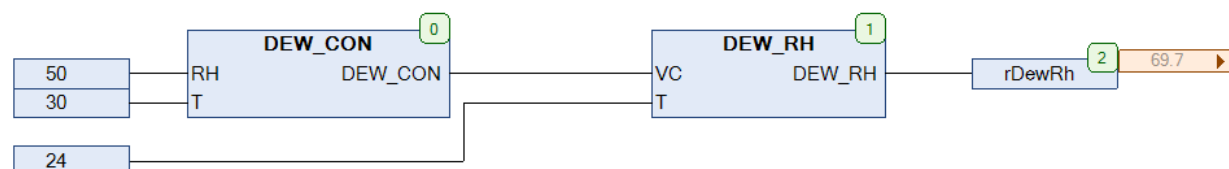
Тип модуля: функция	Переменная	Тип	Описание
Входы	VC	REAL	Абсолютная влажность воздуха, г/м <sup>3</sup> .
	T	REAL	Температура, °С.
Выходы	DEW_RH	REAL	Относительная влажность, %.
Используемые модули	<a href="#">DEW_CON</a>		

Рис. 5.11. Внешний вид функции **DEW\_RH** на языке CFC

Функция **DEW\_RH** возвращает относительную влажность воздуха для заданной температуры **T** (в диапазоне  $-40 \leq T \leq 90$ ) и абсолютной влажности **VC**.

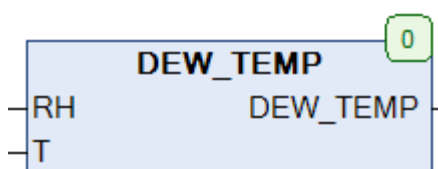
Рис. 5.12. Пример работы с функцией **DEW\_RH** на языке CFC

Функция **DEW\_RH** может использоваться совместно с функцией [DEW\\_CON](#) для определения изменения относительной влажности воздуха при его нагреве или охлаждении. В приведенном ниже примере определяется изменение относительной влажности при охлаждении воздуха от 30 °С (при **RH=50%**) до 24 °С.

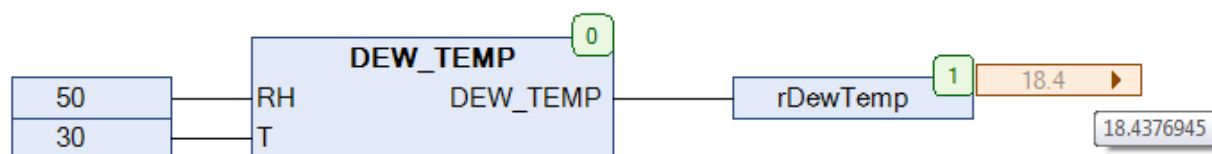
Рис. 5.13. Пример совместного использования функций **DEW\_RH** и **DEW\_CON** на языке CFC

## 5.7. DEW\_TEMP

Тип модуля: функция	Переменная	Тип	Описание
Входы	RH	REAL	Относительная влажность, %.
	T	REAL	Температура, °C.
Выходы	DEW_TEMP	REAL	Температура точки росы, °C.
Используемые модули	OSCAT_BASIC.EXP10, OSCAT_BASIC.PHYS		

Рис. 5.14. Внешний вид функции **DEW\_TEMP** на языке CFC

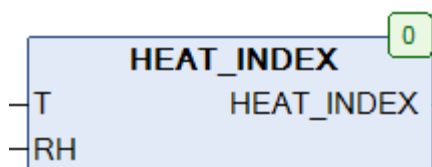
Функция **DEW\_TEMP** возвращает температуру [точки росы](#) воздуха для заданной температуры **T** (в диапазоне  $-40 \leq T \leq 90$ ) и относительной влажности **RH**.

Рис. 5.15. Пример работы с функцией **DEW\_TEMP** на языке CFC

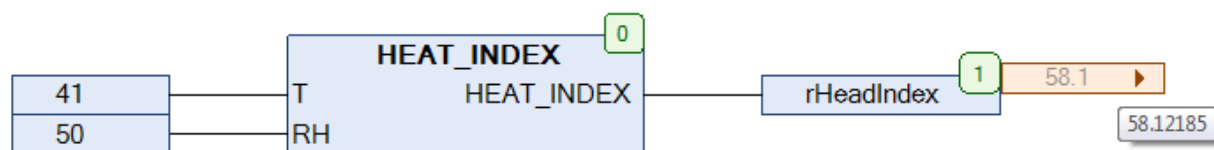
Для тестирования функции можно воспользоваться [онлайн-калькулятором точки росы](#).

## 5.8. HEAT\_INDEX

Тип модуля: функция	Переменная	Тип	Описание
Входы	T	REAL	Температура, °C.
	RH	REAL	Относительная влажность, %.
Выходы	HEAT_INDEX	REAL	Индекс жары, °C.
Используемые модули	OSCAT_BASIC.C_TO_F, OSCAT_BASIC.F_TO_C		

Рис. 5.16. Внешний вид функции **HEAT\_INDEX** на языке CFC

Функция **HEAT\_INDEX** возвращает [индекс жары](#) (ощущаемую температуру) для заданной температуры **T** и относительной влажности **RH**. Функция возвращает корректные значения для  $20 \leq T$  и  $20 \leq RH$ . В противном случае функция возвращает значение температуры **T**.

Рис. 5.17. Пример работы с функцией **HEAT\_INDEX** на языке CFC

## 5.9. HEAT\_METER

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	TF	REAL	Температура подачи воды, °С.
	TR	REAL	Температура обратной воды, °С.
	LPH	REAL	Расход (литр/час или литр/импульс).
	E	BOOL	Сигнал управления расходомером.
	RST	BOOL	Передний фронт – сброс счетчиков.
<b>Выходы</b>	C	REAL	Текущее потребление теплоты, Дж/час.
<b>Входы-выходы</b>	Y	REAL	Выделенная теплота, Дж.
<b>Параметры</b>	CP	REAL	Удельная теплоемкость 2-го вещества, Дж/(кг·К).
	DENSITY	REAL	Плотность 2-го вещества, кг/м <sup>3</sup> .
	CONTENT	REAL	Доля 2-го вещества в растворе.
	PULSE_MODE	BOOL	TRUE – режим импульсного расходомера
	RETURN_METER	BOOL	TRUE – расчет возвращенной теплоты, FALSE – потраченной
	AVG_TIME	TIME	Период расчета текущего расхода тепла (время опроса входов TF/TR).
Используемые модули	<a href="#">WATER_DENSITY</a> , <a href="#">WATER_ENTHALPY</a> , OSCAT_BASIC.FT_INT2, OSCAT_BASIC.T_PLC_MS		

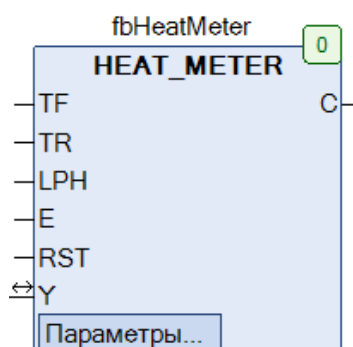


Рис. 5.18. Внешний вид ФБ HEAT\_METER на языке CFC

Функциональный блок **HEAT\_METER** представляет собой модуль измерения теплоты. Вход-выход **Y** содержит значение суммарной теплоты в Дж. Выход **C** характеризует текущее изменение теплоты в Дж/час и обновляется с периодом **AVG\_TIME**. Параметр **RETURN\_METER** определяет, измеряет ли блок потраченную теплоту (**RETURN\_METER=FALSE**) или возвращенную (**RETURN\_METER=TRUE**).

На входы **TF** и **TR** подается температура прямой и обратной воды. Вход **LPH** определяет расход воды. Параметр **PULSE\_MODE** определяет режим работы блока. Если **PULSE\_MODE** имеет значение **FALSE**, то блок работает в режиме интегрирования и пока вход **E** имеет значение **TRUE**, значение входа **LPH** интерпретируется как расход за час. Если **PULSE\_MODE** имеет значение **TRUE**, то блок работает в импульсном режиме и при импульсе по переднему фронту входа **E** увеличивает значение выхода **Y**. По переднему фронту входа **RST** происходит сброс памяти блока, после чего отсчет начинается заново.



Если в воде растворено другое вещество, то параметр **CONTENT** определяет его долю в растворе (0.5 = 50%; для дистиллированной воды **CONTENT=0**), а параметры **CP** и **DENSITY** – удельную теплоемкость и плотность.

### 5.10. HEAT\_TEMP

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	T_EXT	REAL	Текущая уличная температура, °С.
	T_INT	REAL	Текущая комнатная температура, °С.
	OFFSET	REAL	Корректировка комнатной температуры, °С.
	T_REQ	REAL	Уставка температуры контура отопления, °С.
<b>Выходы</b>	TY	REAL	Требуемая температура контура отопления, °С.
	HEAT	BOOL	Флаг «требуется нагрев».
<b>Параметры</b>	TY_MAX	REAL	Макс. допустимая температура контура отопления, °С.
	TY_MIN	REAL	Мин. допустимая температура контура отопления, °С.
	TY_CONFIG	REAL	Расчетная температура контура отопления, °С.
	T_INT_CONFIG	REAL	Расчетная комнатная температура, °С.
	T_EXT_CONFIG	REAL	Расчетная уличная температура, °С.
	T_DIFF	REAL	Разность между температурами подачи и обратной воды, °С.
	C	REAL	Коэффициент нелинейности теплоотдачи.
	H	REAL	Гистерезис нагрева, °С.

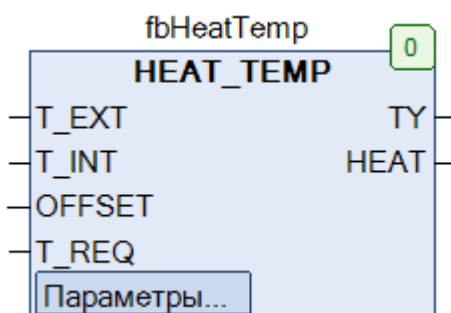


Рис. 5.18. Внешний вид ФБ HEAT\_TEMP на языке CFC

Функциональный блок **HEAT\_TEMP** возвращает требуемую температуру контура отопления **TY** для прогрева помещения до расчетной комнатной температуры **T\_INT\_CONFIG** при текущей комнатной температуре **T\_INT** и уличной температуре **T\_EXT**. Значение входа **OFFSET** позволяет корректировать значение текущей комнатной температуры (например, в случае погрешности показаний датчика). Для расчета используются параметры **TY\_CONFIG** (расчетная температура контура отопления), **T\_EXT\_CONFIG** (расчетная уличная температура) и **T\_DIFF** (разность между температурами подачи и обратной воды). Значение требуемой температуры контура отопления **TY** ограничено верхним и нижним допустимыми пределами **TY\_MAX** и **TY\_MIN**. Расчет **TY**

производится только в том случае, если текущая уличная температура **T\_EXT** ниже комнатной (**T\_INT + OFFSET**) на значение гистерезиса **H**. Если **TY > 0**, то выход **HEAT** имеет значение **TRUE**.

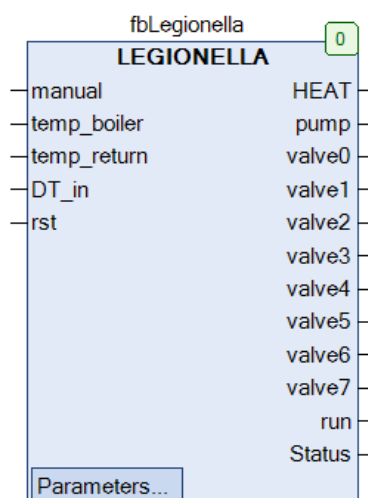
Вход **T\_REQ** может использоваться для задания уставки температуры контура отопления. Если ее значение превышает рассчитанное значение **TY**, то **TY = T\_REQ**.

Параметр **C** определяет коэффициент нелинейности теплоотдачи, зависящий от оборудования, которое производит нагрев помещения:

- Конвекторы:  $C = 1.25 \dots 1.45$
- Панельные радиаторы:  $C = 1.20 \dots 1.30$
- Радиаторы:  $C = 1.30$
- Трубы:  $C = 1.25$
- Теплый пол:  $C = 1.10$

## 5.11. LEGIONELLA

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	manual	REAL	TRUE – принудительный запуск блока.
	temp_boiler	REAL	Текущая температура бойлера.
	temp_return	BOOL	Текущая температура обратной воды.
	DT_in	DT	Текущие дата и время.
	rst	BOOL	Передний фронт – прервать дезинфекцию.
<b>Выходы</b>	HEAT	BOOL	Сигнал управления нагревателем.
	pump	BOOL	Сигнал включения циркуляционного насоса.
	valve0...valve7	BOOL	Управление циркуляционными клапанами 0...7.
	run	BOOL	Флаг «запущен процесс дезинфекции».
	Status	BYTE	ESR-код.
<b>Параметры</b>	T_start	TOD	Плановое время дезинфекции.
	day	INT	День дезинфекции (1 – Понедельник).
	temp_set	REAL	Уставка температуры бойлера.
	temp_offset	REAL	Корректировка температуры бойлера.
	temp_hys	REAL	Гистерезис температуры бойлера.
	t_max_heat	TIME	Максимальное время нагрева бойлера.
	t_max_ret	TIME	Максимальное время нагрева труб.
	tp_0...tp7	TIME	Время дезинфекции для трубы клапана 0...7.

Рис. 5.19. Внешний вид ФБ **LEGIONELLA** на языке CFC

Функциональный блок **LEGIONELLA** предназначен для управления дезинфекцией труб отопления. Процесс дезинфекции производится еженедельно в день недели **day** (1 – Понедельник) и время суток **T\_start**. На вход **DT\_in** блока требуется завести текущее системное время. По переднему фронту входа **manual** процесс дезинфекции запускается принудительно. На входы **temp\_boiler** и **temp\_return** подается текущая температура бойлера и обратной воды.

Процесс дезинфекции происходит следующим образом:

- выход управления нагревателем **HEAT** принимает значение **TRUE**, и происходит нагрев воды в бойлере до заданного значения (**temp\_set + temp\_offset + temp\_hys**). Если в течение времени **t\_max\_heat** текущая температура **temp\_boiler** не поднимается до заданного значения, то процесс дезинфекции прерывается;
- после нагрева происходит включение циркуляционного насоса (**pump=TRUE**);
- далее открывается клапан 0 (**valve0=TRUE**) и выполняется нагрев воды (**HEAT=TRUE**) до тех пор, пока температура обратки (**temp\_return**) не достигнет температуры уставки (**temp\_set**). После этого процесс нагрева продолжается еще в течение времени **tp\_0**. Если значение уставки не достигается за время **t\_max\_ret**, то процесс дезинфекции прерывается. Если в системе отсутствует датчик температуры обратной воды, то на вход **temp\_return** должна быть присвоена константа, значение которой превышает **temp\_set**. В этом случае нагрев воды будет происходить только в течение времени **tp\_0**;
- процедура повторяется для клапанов 1...7.

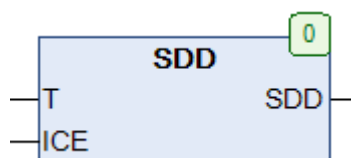
Пока выполняется процесс дезинфекции, выход **run** имеет значение **TRUE**. Дезинфекция может быть прервана принудительно по переднему фронту на входе **rst**.

Выход **Status** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**:

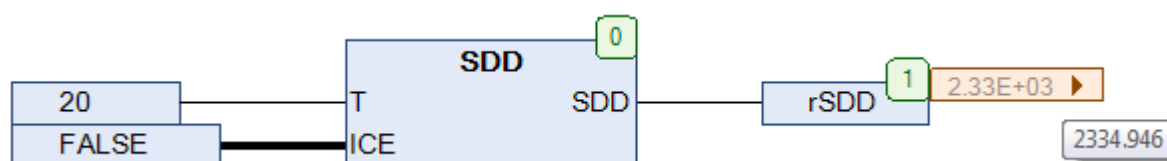
Значение выхода Status	Описание
110	Блок в режиме ожидания.
111	Запущен процесс дезинфекции.
1	Не удалось нагреть бойлер.
2...9	Не удалось нагреть трубу клапана 0...7.

## 5.12. SDD

Тип модуля: функция	Переменная	Тип	Описание
Входы	T	REAL	Температура, °C.
	ICE	REAL	Среда: TRUE – лед, FALSE - вода
Выходы	SDD	REAL	Давление насыщенного пара, Па.

Рис. 5.20. Внешний вид функции **SDD** на языке CFC

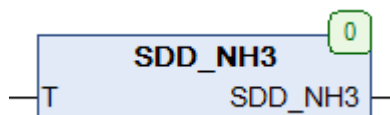
Функция **SDD** возвращает давление [насыщенного пара](#) над водой (при **ICE=FALSE**) или поверхностью льда (при **ICE=TRUE**) для температуры **T** и [нормального атмосферного давления](#), вычисленное по формуле Магнуса. Функция возвращает корректные значения для  $-30 \leq T \leq 70$  (для воды) и  $-60 \leq T \leq 0$  (для льда).

Рис. 5.21. Пример работы с функцией **SDD** на языке CFC

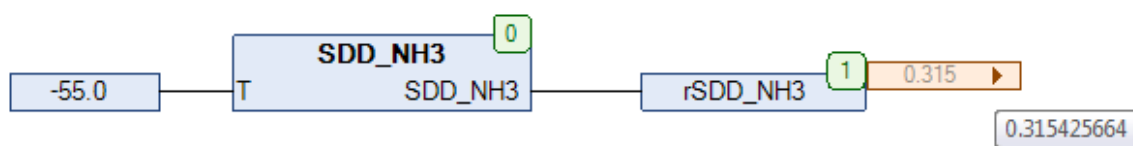
Для тестирования функции можно воспользоваться [таблицей из Википедии](#).

## 5.13. SDD\_NH3

Тип модуля: функция	Переменная	Тип	Описание
<b>Входы</b>	T	REAL	Температура, °С.
<b>Выходы</b>	SDD_NH3	REAL	Давление насыщенного пара, Бар.

Рис. 5.22. Внешний вид функции **SDD\_NH3** на языке CFC

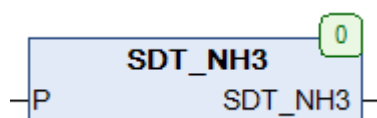
Функция **SDD\_NH3** возвращает давление [насыщенного пара](#) над аммиаком для температуры T и [нормального атмосферного давления](#). Функция возвращает корректные значения для  $-109 \leq T \leq 98$ .

Рис. 5.23. Пример работы с функцией **SDD\_NH3** на языке CFC

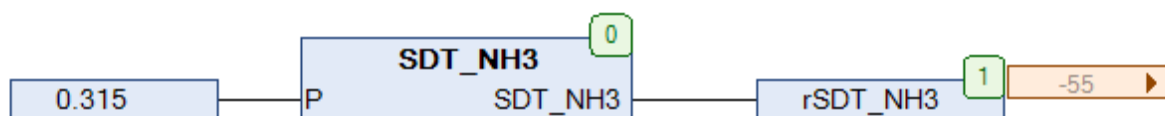
Для тестирования функции можно воспользоваться [таблицей из Википедии](#).

## 5.14. SDT\_NH3

Тип модуля: функция	Переменная	Тип	Описание
<b>Входы</b>	P	REAL	Давление, Бар.
<b>Выходы</b>	SDT_NH3	REAL	Температура насыщенного пара, °C.

Рис. 5.24. Внешний вид функции **SDT\_NH3** на языке CFC

Функция **SDT\_NH3** возвращает температуру [насыщенного пара](#) над аммиаком для давления P. Функция возвращает корректные значения для  $100 \leq P \leq 6000000$ .

Рис. 5.25. Пример работы с функцией **SDT\_NH3** на языке CFC

Для тестирования функции можно воспользоваться [таблицей из Википедии](#).

## 5.15. T\_AVG24

Тип модуля: ФБ	Переменная	Тип	Описание
Входы	TS	INT	Измеренная температура (243 = 24.3 °C).
	DTI	DT	Текущие дата и время.
	RST	BOOL	Передний фронт – сброс накопленных значений.
Выходы	TA	REAL	Последняя сохраненная температура.
	TP	BOOL	Флаг «значение выхода T24 обновлено».
Входы-выходы	T24	REAL	Среднесуточная температура за 24 часа.
	T24_MAX	REAL	Максимальная температура за 24 часа.
	T24_MIN	REAL	Минимальная температура за 24 часа.
Параметры	T_FILTER	TIME	Постоянная времени фильтра.
	SCALE	REAL	Коэффициент «наклона».
	OFS	REAL	Коэффициент «сдвига».
Используемые модули	OSCAT_BASIC.FILTER_I, OSCAT_BASIC.INC1		

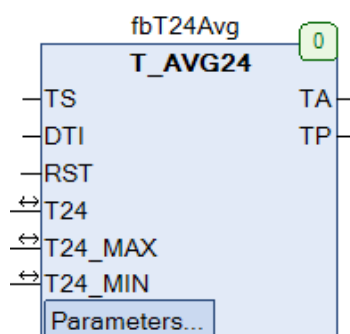


Рис. 5.26. Внешний вид ФБ T\_AVG24 на языке CFC

Функциональный блок **T\_AVG24** представляет собой модуль измерения среднесуточной температуры. На вход **TS** подается сигнал от датчика температуры в виде целого числа ( $TS = 234 \Rightarrow 24.3 \text{ °C}$ ). При обработке входа **TS** используется [фильтр нижних частот](#) с постоянной времени **T\_FILTER**, масштабирование с коэффициентом «наклона» **SCALE** и «сдвига» **OFS**. На вход **DTI** подается текущее значение системного времени.

На выходе **TA** отображается последнее сохраненное значение температуры. Блок сохраняет значение входа **TS** каждые полчаса и хранит 48 значений в режиме циклического буфера. Входы-выходы **T24**, **T24\_MAX** и **T24\_MIN** содержат среднесуточную, максимальную и минимальную температуру за последние 24 часа, рассчитанную на основании значений этого буфера. При обновлении входа-выхода **T24** выход **TP** принимает значение **TRUE** на один цикл ПЛК.

По переднему фронту входа **RST** происходит обнуление входов-выходов блока.

Если при первом вызове блока (при старте ПЛК или после сброса через вход **RST**) вход-выход **T24** имеет значение **-1000**, то первое значение для буфера рассчитывается на основании текущего значения входа **TS**. Если же **T24** имеет значение, отличное от **-1000**, то буфер заполняется этим значением. Это может использоваться для немедленного запуска блока после перезагрузки ПЛК.



## 5.16. TANK\_LEVEL

Тип модуля: ФБ	Переменная	Тип	Описание
Входы	LEVEL	BOOL	TRUE – уровень жидкости в резервуаре слишком низкий.
	LEAK	BOOL	TRUE – протечка.
	ACLR	BOOL	Передний фронт – квитирование аварии.
Выходы	VALVE	BOOL	TRUE – включить насос.
	ALARM	BOOL	TRUE – есть несквитированная авария.
	STATUS	BYTE	ESR-код.
Параметры	MAX_VALVE_TIME	TIME	Допустимое время открытия клапана.
	LEVEL_DELAY_TIME	TIME	Время задержки обработки входа LEVEL.
Используемые модули	<a href="#">ACTUATOR_COIL</a> , <a href="#">OSCAT_BASIC.TONOF</a>		

Рис. 5.27. Внешний вид ФБ **TANK\_LEVEL** на языке CFC

Функциональный блок **TANK\_LEVEL** используется для поддержания уровня жидкости в резервуаре при помощи насоса. На вход **LEVEL** подключается сигнал датчика контроля уровня (**TRUE** – уровень жидкости слишком низкий, **FALSE** – уровень жидкости в норме). Если вход **LEVEL** имеет значение **TRUE**, то выход **VALVE**, используемый для управления насосом, также принимает значение **TRUE**. При достижении заданного уровня жидкости сигнал датчика контроля уровня должен принимать значение **FALSE**.

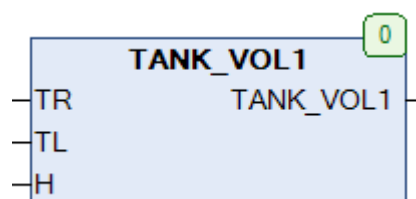
Параметр **LEVEL\_DELAY\_TIME** определяет задержку между получением сигнала о недостаточном уровне жидкости и включением насоса. Если время работы насоса превышает **MAX\_VALVE\_TIME** или вход **LEAK** (сигнал контроля протечки) принимает значение **TRUE**, то выход **VALVE** принимает значение **FALSE**, а выход **ALARM** – значение **TRUE**. По переднему фронту входа **ACLR** происходит деактивация выхода **ALARM**, после чего блок продолжает работу в нормальном режиме.

Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**:

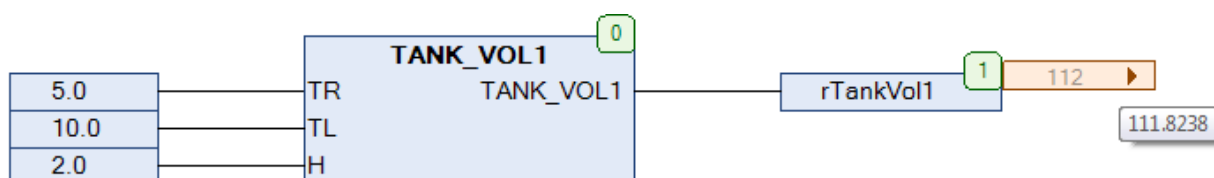
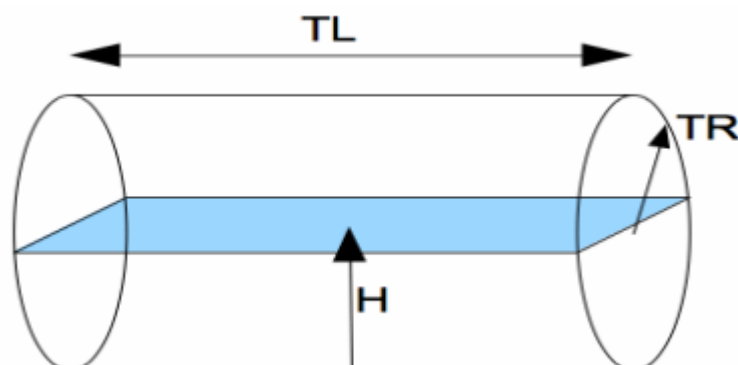
Значение выхода STATUS	Описание
1	Срабатывание датчика контроля протечки.
2	Время открытия клапана превысило <b>MAX_VALVE_TIME</b> .
100	Закрытие клапана после наполнения резервуара.
101	Выполнен сброс аварии.
102	Открытие клапана для наполнения резервуара.

## 5.17. TANK\_VOL1

Тип модуля: функция	Переменная	Тип	Описание
Входы	TR	REAL	Радиус резервуара, м.
	TL	REAL	Длина резервуара, м.
	H	REAL	Уровень жидкости в резервуаре, м.
Выходы	TANK_VOL1	REAL	Объем жидкости в резервуаре, м <sup>3</sup> .
Используемые модули	OSCAT_BASIC.CIRCLE_SEG		

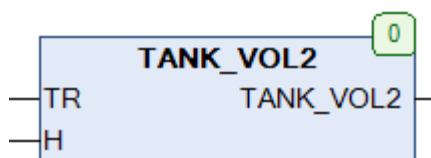
Рис. 5.28. Внешний вид функции **TANK\_VOL1** на языке CFC

Функция **TANK\_VOL1** возвращает объем жидкости в трубчатом резервуаре с длиной **TL** и радиусом стенки **TR**, заполненном на **H** метров.

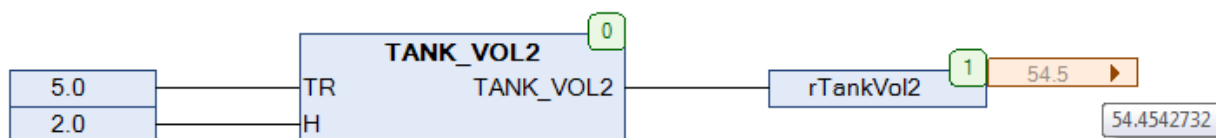
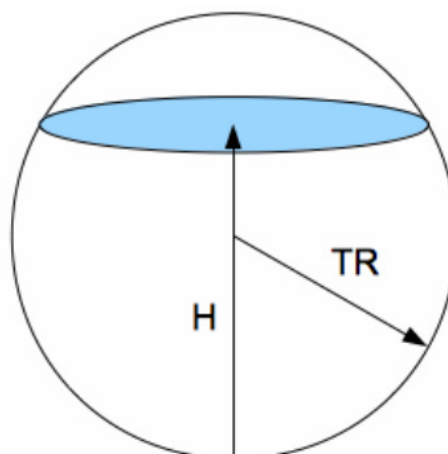
Рис. 5.29. Пример работы с функцией **TANK\_VOL1** на языке CFC

## 5.18. TANK\_VOL2

Тип модуля: функция	Переменная	Тип	Описание
Входы	TR	REAL	Радиус резервуара, м.
	H	REAL	Уровень жидкости в резервуаре, м.
Выходы	TANK_VOL2	REAL	Объем жидкости в резервуаре, м <sup>3</sup> .
Используемые модули	OSCAT_BASIC.MATH		

Рис. 5.30. Внешний вид функции **TANK\_VOL2** на языке CFC

Функция **TANK\_VOL2** возвращает объем жидкости в сферическом резервуаре с радиусом **TR**, заполненном на **H** метров.

Рис. 5.31. Пример работы с функцией **TANK\_VOL2** на языке CFC

## 5.19. TEMP\_EXT

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	t_ext1	REAL	Температура датчика 1.
	t_ext2	REAL	Температура датчика 2.
	t_ext3	REAL	Температура датчика 3.
	t_ext_config	BYTE	Режим работы блока.
	dt_in	DT	Текущие дата и время.
<b>Выходы</b>	t_ext	REAL	Выбранное значение температуры.
	heat	BOOL	Флаг «требуется обогрев».
	cool	BOOL	Флаг «требуется охлаждение».
<b>Параметры</b>	t_ext_min	REAL	Нижняя граница допустимых значений.
	t_ext_max	REAL	Верхняя граница допустимых значений.
	t_ext_default	REAL	Значение по умолчанию.
	heat_period_start	DATE	Дата начала периода обогрева.
	heat_period_stop	DATE	Дата окончания периода обогрева.
	cool_period_start	DATE	Дата начала периода охлаждения.
	cool_period_stop	DATE	Дата окончания периода охлаждения.
	heat_start_temp_day	REAL	Минимально допустимая дневная температура в период обогрева.
	heat_start_temp_night	REAL	Минимально допустимая ночная температура в период обогрева.
	heat_stop_temp	REAL	Максимально допустимая температура в период обогрева.
	cool_start_temp_day	REAL	Максимально допустимая дневная температура в период охлаждения.
	cool_start_temp_night	REAL	Максимально допустимая ночная температура в период охлаждения.
	cool_stop_temp	REAL	Минимально допустимая температура в период охлаждения.
	start_day	TOD	Время суток, принимаемое за начало дня.
	start_night	TOD	Время суток, принимаемое за начало ночи.
CYCLE_TIME	TIME	Период обновления выходов блока.	
Используемые модули	OSCAT_BASIC.T_PLC_MS, OSCAT_BASIC.SET_DATE, OSCAT_BASIC.MONTH_OF_DATE, OSCAT_BASIC.DAY_OF_MONTH, OSCAT_BASIC.MULTI_IN		

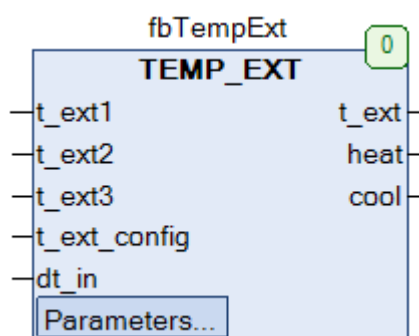


Рис. 5.32. Внешний вид ФБ TEMP\_EXT на языке CFC

Функциональный блок **TEMP\_EXT** используется для контроля уличной температуры для обогрева или охлаждения помещения. На входы **t\_ext1**, **t\_ext2**, **t\_ext3** заводятся сигналы датчиков температуры. На вход **dt\_in** подается значение системного времени.

Вход **t\_ext\_config** определяет режим работы блока, от которого зависит значение выхода **t\_ext**:

Значение входа <b>t_ext_config</b>	Значение выхода <b>t_ext</b>
0	Среднее арифметическое для <b>t_ext1...t_ext3</b>
1	<b>t_ext1</b>
2	<b>t_ext2</b>
3	<b>t_ext3</b>
4	<b>t_ext_default</b>
5	Наименьшее из <b>t_ext1...t_ext3</b>
6	Наибольшее из <b>t_ext1...t_ext3</b>
7	Среднее из <b>t_ext1...t_ext3</b>

Параметры **t\_ext\_min** и **t\_ext\_max** определяют нижний и верхний допустимый предел входных сигналов. Если значение не входит в заданный диапазон или совпадает с его границами, то в режимах 1-4 вместо него используется величина параметра **t\_ext\_default**, а в режимах 0 и 5-7 значение не обрабатывается.

Параметры **heat\_period\_start/heat\_period\_stop** и **cool\_period\_start/cool\_period\_stop** определяют даты начала/окончания периодов обогрева и охлаждения. Параметры **start\_day** и **start\_night** определяют время суток, принимаемое за начало дня и ночи. Параметры **heat\_start\_temp\_day** и **heat\_start\_temp\_night** определяют минимально допустимую дневную и ночную температуру в период обогрева, параметр **heat\_stop\_temp** – максимально допустимую температуру. Параметры **cool\_start\_temp\_day** и **cool\_start\_temp\_night** определяют максимально допустимую дневную и ночную температуру в период охлаждения, параметр **cool\_stop\_temp** – минимально допустимую температуру. В зависимости от значений этих параметров и выхода **t\_ext** определяются значения выходов **heat** (флаг обогрева) и **cool** (флаг охлаждения). См. диаграмму на рис. 5.32а.

Параметр **CYCLE\_TIME** определяет период обновления выходов блока.

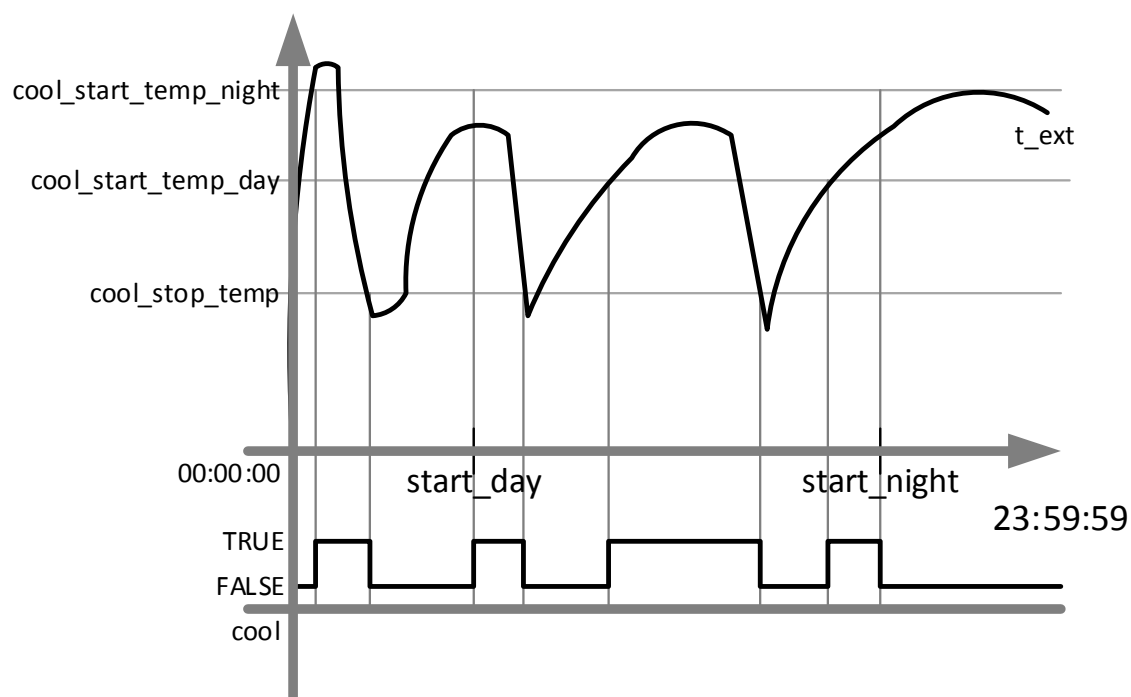
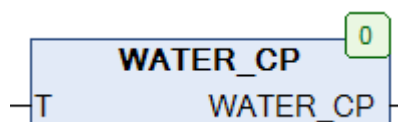


Рис. 5.32а. Временная диаграмма работа функции **TEMP\_EXT**

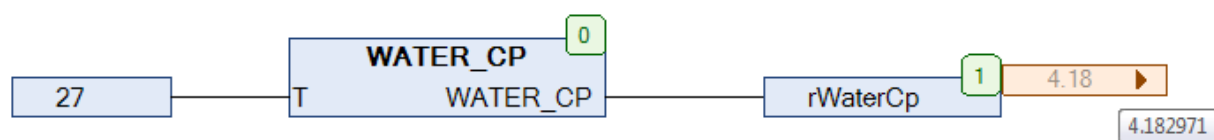
На приведенной выше диаграмме отражена зависимость состояния выхода `cool` от настроек охлаждения и текущей температуры `t_ext`, рассчитанной на основании значений входов `t_ext1`, `t_ext2`, `t_ext3` и `t_ext_config` при условии, что текущая дата `dt_in` находится внутри диапазона `cool_period_start...cool_period_stop`.

## 5.20. WATER\_CP

Тип модуля: функция	Переменная	Тип	Описание
<b>Входы</b>	T	REAL	Температура воды, °С.
<b>Выходы</b>	WATER_CP	REAL	Удельная теплоемкость воды, Дж/(кг·К).
<b>Используемые модули</b>	OSCAT_BASIC.LINEAR_INT		

Рис. 5.33. Внешний вид функции **WATER\_CP** на языке CFC

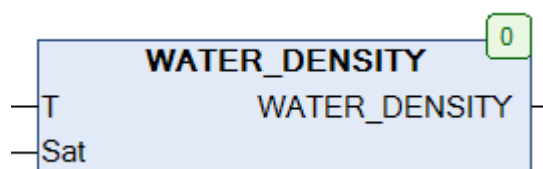
Функция **WATER\_CP** возвращает [удельную теплоемкость](#) воды для заданной температуры **T** при [нормальном атмосферном давлении](#). Функция возвращает корректные значения для диапазона  $0 \leq T \leq 100$ .

Рис. 5.34. Пример работы с функцией **WATER\_CP** на языке CFC

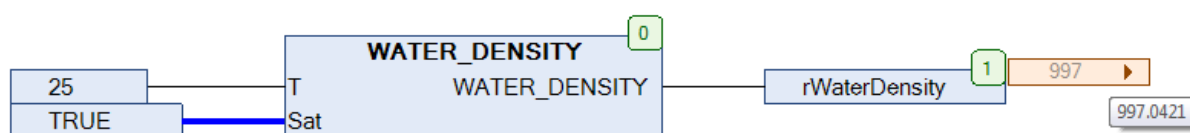
Для тестирования функции можно воспользоваться [таблицей из Википедии](#).

## 5.21. WATER\_DENSITY

Тип модуля: функция	Переменная	Тип	Описание
Входы	T	REAL	Температура воды, °C.
	Sat	BOOL	Тип воды (FALSE – деаэрированная, TRUE – содержащая кислород).
Выходы	WATER_DENSITY	REAL	Плотность пресной воды, г/л.

Рис. 5.35. Внешний вид функции **WATER\_DENSITY** на языке CFC

Функция **WATER\_DENSITY** возвращает плотность пресной воды для заданной температуры **T** при [нормальном атмосферном давлении](#). Функция позволяет рассчитать плотность воды в жидком состоянии (но не твердом и газообразном). Если вход Sat имеет значение FALSE, то функция возвращает плотность [деаэрированной](#) воды; если вход Sat имеет значение TRUE, то функция возвращает плотность воды, содержащей растворенный кислород. Функция возвращает корректные значения для  $0 \leq T \leq 100$ , при этом относительная погрешность вычислений не превышает 0.01%.

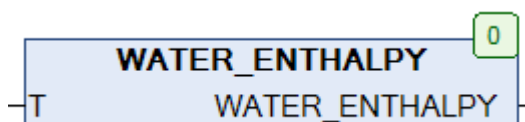
Рис. 5.36. Пример работы с функцией **WATER\_DENSITY** на языке CFC

Для тестирования функции можно воспользоваться [таблицей из Википедии](#).



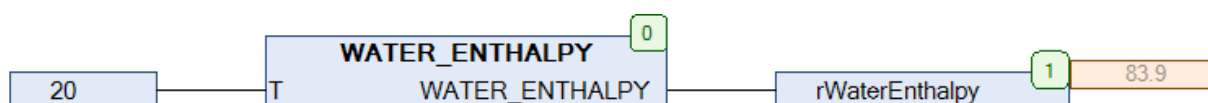
## 5.22. WATER\_ENTHALPY

Тип модуля: функция	Переменная	Тип	Описание
<b>Входы</b>	T	REAL	Температура воды, °С.
<b>Выходы</b>	WATER_ENTHALPY	REAL	Энтальпия воды, кДж/кг.
<b>Используемые модули</b>	OSCAT_BASIC.LINEAR_INT		

Рис. 5.37. Внешний вид функции **WATER\_ENTHALPY** на языке CFC

Функция **WATER\_ENTHALPY** возвращает [энтальпию](#) воздуха при нормальном атмосферном давлении для заданной температуры **T**. Функция возвращает корректные значения для  $0 \leq T \leq 100$ . Расчет выполняется на основе табличной интерполяции с шагом в 10 °С, что дает приемлемую точность для большинства реальных применений. Функция может выполняться для вычисления количества теплоты, необходимого для нагрева воды с температурой T1 до температуры T2:

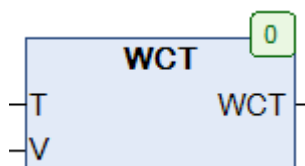
$$H = \text{WATER\_ENTHALPY}(T2) - \text{WATER\_ENTHALPY}(T1)$$

Рис. 5.38. Пример работы с функцией **WATER\_ENTHALPY** на языке CFC

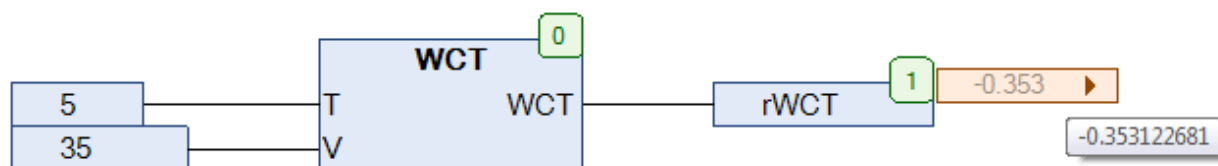
Для тестирования функции можно воспользоваться [таблицей из Википедии](#).

## 5.23. WCT

Тип модуля: функция	Переменная	Тип	Описание
Входы	T	REAL	Температура, °С.
	V	REAL	Скорость ветра, км/ч.
Выходы	WCT	REAL	Ветро-холодовой индекс, °С.

Рис. 5.39. Внешний вид функции **WCT** на языке CFC

Функция **WCT** возвращает [ветро-холодовой индекс](#) для измеренной температуры **T** и скорости ветра **V**. Функция возвращает корректные значения для  $T < 10$  при  $V > 5$ . Во всех остальных случаях функция возвращает значение **T**.

Рис. 5.40. Пример работы с функцией **WCT** на языке CFC

## 6. Кнопки, генераторы импульсов, таймеры

### 6.1. CLICK

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	in	BOOL	Текущее состояние кнопки.
<b>Выходы</b>	Q	BOOL	Выходное значение.
	single	BOOL	Единичный импульс при первом нажатии в последовательности.
	double	BOOL	Единичный импульс при втором нажатии в последовательности.
	triple	BOOL	Единичный импульс при третьем нажатии в последовательности.
	status	BYTE	ESR-код.
<b>Параметры</b>	T_debounce	TIME	Минимальное время нажатия.
	T_short	TIME	Максимальное время нажатия.
	T_pause	TIME	Максимальное время между последовательными нажатиями.
	T_reconfig	TIME	Время определения типа кнопки ( <a href="#">НО\НЗ</a> ).
Используемые модули	<a href="#">SW_RECONFIG</a> , <a href="#">OSCAT_BASIC.T_PLC_MS</a>		

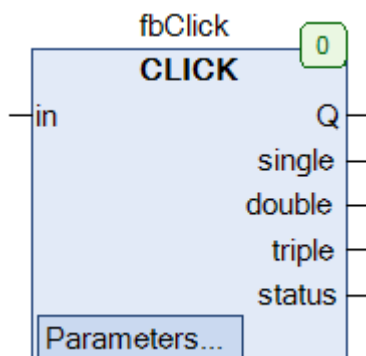


Рис. 6.1. Внешний вид ФБ **CLICK** на языке CFC

Функциональный блок **CLICK** представляет собой интерфейсный модуль для кнопки. На вход **IN** подключается дискретный сигнал от кнопки, который передается на выход **Q**. По умолчанию считается, что кнопка подключена с помощью нормально разомкнутого контакта. Если вход **IN** в течение времени **T\_reconfig** имеет значение **TRUE**, то контакт считается нормально замкнутым, и значение выхода **Q** будет являться инверсным по отношению к входу **IN**. Параметр **T\_debounce** определяет минимально фиксируемое время нажатия для игнорирования дребезга контактов.

Выходы **single/double/triple** принимают значение **TRUE** на один цикл ПЛК, если текущее нажатие является первым/вторым/третьим в последовательности. Последовательность определяется параметрами **T\_short** (максимальное время нажатия) и **T\_pause** (максимальное время между нажатиями). Если время нажатия превышает **T\_short** или пауза между нажатиями

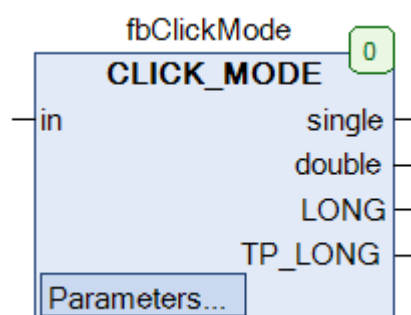
превышает **T\_pause**, то последовательность считается прерванной, и выходы **single/double/triple** принимают значение **FALSE**. В пределах цикла ПЛК только один из этих выходов может иметь значение **TRUE**.

Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**:

Значение выхода STATUS	Описание
110	Кнопка не нажата
111	Выход single имеет значение <b>TRUE</b>
112	Выход double имеет значение <b>TRUE</b>
113	Выход triple имеет значение <b>TRUE</b>

## 6.2. CLICK\_MODE

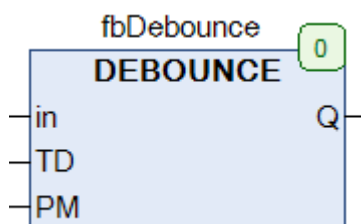
Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	in	BOOL	Текущее состояние кнопки.
<b>Выходы</b>	single	BOOL	Единичный импульс при первом нажатии в последовательности.
	double	BOOL	Единичный импульс при втором нажатии в последовательности.
	LONG	BOOL	Флаг «удержание кнопки».
	TP_LONG	BOOL	Единичный импульс при длительном нажатии.
<b>Параметры</b>	T_LONG	TIME	Уставка таймера.

Рис. 6.2. Внешний вид ФБ **CLICK\_MODE** на языке CFC

Функциональный блок **CLICK\_MODE** представляет собой интерфейсный модуль для кнопки. На вход **IN** подключается дискретный сигнал от кнопки. Параметр **T\_LONG** определяет время уставки таймера блока. Таймер запускается по переднему фронту на входе **IN** (т.е. при нажатии кнопки). Если за время **T\_LONG** не произойдет других нажатий, то после срабатывания таймера выход **SINGLE** примет значение **TRUE** на один цикл ПЛК. Если за время **T\_LONG** произойдет еще одно нажатие, то после срабатывания таймера выход **DOUBLE** примет значение **TRUE** на один цикл ПЛК. Если за время **T\_LONG** произойдет еще два или более нажатия, то выходы блока не изменятся. Если вход **IN** удерживается в состоянии **TRUE** на протяжении времени **T\_LONG**, то на выходе **TP\_LONG** генерируется единичный импульс, а выход **LONG** принимает значение **TRUE** и сохраняет его до тех пор, пока вход **IN** не примет значение **FALSE**.

## 6.3. DEBOUNCE

Тип модуля: ФБ	Переменная	Тип	Описание
Входы	in	BOOL	Текущее состояние кнопки или выключателя.
	TD	TIME	Уставка таймера.
	PM	BOOL	Режим работы блока (TRUE – единичный импульс, FALSE – удержание на время TD).
Выходы	Q	BOOL	Флаг «произошло нажатие».

Рис. 6.3. Внешний вид ФБ **DEBOUNCE** на языке CFC

Функциональный блок **DEBOUNCE** представляет собой интерфейсный модуль для кнопки. На вход **IN** подключается дискретный сигнал от кнопки. Если вход **PM** имеет значение **FALSE**, то по переднему фронту на входе **IN** выход **Q** активируется на время **TD**. Если вход **PM** имеет значение **TRUE**, то по переднему фронту на входе **IN** на выходе **Q** генерируется единичный импульс. Следующий импульс может быть сгенерирован только после удержания значения **FALSE** на входе **in** в течение времени **TD**.

## 6.4. DIMM\_2

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	SET	BOOL	Передний фронт – установить яркость VAL.
	VAL	BYTE	Требуемое значение яркости.
	I1	BOOL	Сигнал повышения яркости.
	I2	BOOL	Сигнал понижения яркости.
	RST	BOOL	Передний фронт – выключить свет.
<b>Выходы</b>	Q	BOOL	TRUE – свет включен (сигнал для промежуточного реле).
	D1	BOOL	Двойное нажатие на кнопку I1 (зависит от настроек: единичный импульс, или инверсия).
	D2	BOOL	Двойное нажатие на кнопку I2(зависит от настроек: единичный импульс, или инверсия).
<b>Входы-выходы</b>	OUT	BYTE	Текущая яркость (0...MAX_ON).
<b>Параметры</b>	T_DEBOUNCE	TIME	Минимальное время нажатия кнопок.
	T_ON_MAX	TIME	Максимальное время активности выхода Q (T#0s – время не ограничено).
	T_DIMM_START	TIME	Время длительного нажатия кнопки I1/I2.
	T_DIMM	TIME	Время диммирования.
	MIN_ON	BYTE	Минимальное значение яркости.
	MAX_ON	BYTE	Максимальное значение яркости.
	RST_OUT	BOOL	TRUE – обнулять OUT по фронту RST.
	SOFT_DIMM	BOOL	Режим плавного повышения яркости.
	DBL1_TOG	BOOL	TRUE – выход Dx инвертируется по двойному нажатию кнопки Ix.
	DBL2_TOG	BOOL	
	DBL1_SET	BOOL	TRUE – по двойному нажатию кнопки Ix VAL принимает значение DBLx_POS.
	DBL2_SET	BOOL	
	DBL1_POS	BYTE	Уставка для VAL от кнопки I1.
DBL2_POS	BYTE	Уставка для VAL от кнопки I2.	
Используемые модули	<a href="#">CLICK_MODE</a> , OSCAT_BASIC._RMP_B		

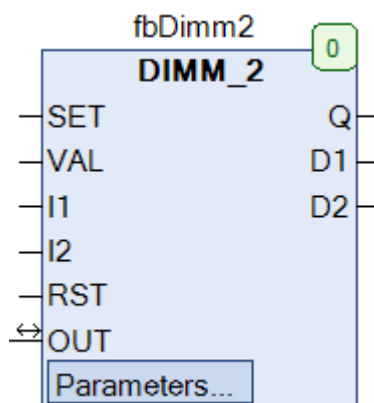


Рис. 6.4. Внешний вид ФБ DIMM\_2 на языке CFC

Функциональный блок **DIMM\_2** представляет собой интерфейсный модуль для управления диммером, оснащенным кнопками «больше»/«меньше». По переднему фронту на входе **SET** выход **Q** принимает значение **TRUE**, а сигнал управления диммером **OUT** – значение **VAL**. По переднему фронту на входе **RST** выходы блока принимают значение **FALSE**. Если параметр **RST\_OUT** имеет значение **TRUE**, то по переднему фронту на входе **RST** также происходит обнуление сигнала **OUT**.

Сигнал **Q** может использоваться в тех случаях, когда необходимо гарантировать отсутствие потенциала (например, в целях безопасности) на «выключенной» лампе. В этом случае для возможности отключения света устанавливается дополнительное промежуточное механическое реле, управляемое дискретным сигналом **Q**. Параметр **T\_ON\_MAX** определяет максимальное время активности выхода **Q**, после которого он сбрасывается в **FALSE**. Если **T\_ON\_MAX=T#0s**, то время активности выхода не ограничивается.

Входы **I1** и **I2** используются для управления яркостью освещения. Параметр **T\_DEBOUNCE** определяет минимальное фиксируемое время нажатия кнопок и используется для предотвращениядребезга контактов.

При одиночном нажатии на кнопку **I1** выход **Q** принимает значение **TRUE**, а значение входа-выхода **OUT** ограничивается диапазоном [**MIN\_ON**..**MAX\_ON**]. При одиночном нажатии на кнопку **I2** выход **Q** принимает значение **FALSE**.

При длительном (превышающем время **T\_DIMM\_START**) нажатии на кнопку **I1** уровень яркости **OUT** плавно повышается от текущего значения **VAL** (если **SOFT\_DIM=FALSE**) или от значения **MIN\_ON** (если **SOFT\_DIM=TRUE**) до значения **MAX\_ON**. При длительном (превышающем время **T\_DIMM\_START**) нажатии на кнопку **I2** уровень яркости **OUT** плавно снижается до **0**. Параметр **T\_DIMM** определяет время изменения яркости. После начала изменения яркости кнопка **I1/I2** может быть отпущена.

Выходы **D1** и **D2** управляются входами **I1** и **I2**. При двойном нажатии на кнопку **I<sub>x</sub>**, соответствующий выход **D<sub>x</sub>** принимает значение **TRUE** на один цикл ПЛК. Если параметр **DBL<sub>x</sub>\_TOG** имеет значение **TRUE**, то при двойном нажатии на кнопку **I<sub>x</sub>** выход **D<sub>x</sub>** инвертируется. Если переменная **DBL<sub>x</sub>\_SET** имеет значение **TRUE**, то при двойном нажатии на кнопку **I<sub>x</sub>** выход **D<sub>x</sub>** не меняет свое значение, а вход-выход **OUT** принимает значение **DBL<sub>x</sub>\_POS**, при этом выход **Q** принимает значение **TRUE**.



## 6.5. DIMM\_I

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	SET	BOOL	Передний фронт – установить яркость VAL.
	VAL	BYTE	Требуемое значение яркости.
	IN	BOOL	Сигнал управления яркостью.
	RST	BOOL	Передний фронт – выключить свет.
<b>Выходы</b>	Q	BOOL	TRUE – свет включен (сигнал для промежуточного реле).
	DBL	BOOL	Сигнал от кнопки IN.
<b>Входы-выходы</b>	OUT	BYTE	Сигнал управления диммером (текущая степень яркости).
<b>Параметры</b>	T_DEBOUNCE	TIME	Минимальное время нажатия кнопки.
	T_RECONFIG	TIME	Время определения типа кнопки (НО\НЗ).
	T_ON_MAX	TIME	Максимальное время активности выхода Q (T#0s – время не ограничено).
	T_DIMM_START	TIME	Время длительного нажатия кнопки I1/I2.
	T_DIMM	TIME	Время диммирования.
	MIN_ON	BYTE	Минимальное значение яркости.
	MAX_ON	BYTE	Максимальное значение яркости.
	SOFT_DIMM	BOOL	Режим плавного повышения яркости.
	DBL_toggle	BOOL	TRUE – выход DBL инвертируется по двойному нажатию кнопки IN.
RST_OUT	BOOL	TRUE – обнулять OUT по фронту RST.	
Используемые модули	<a href="#">CLICK_MODE</a> , <a href="#">SW_RECONFIG</a> , OSCAT_BASIC._RMP_V		

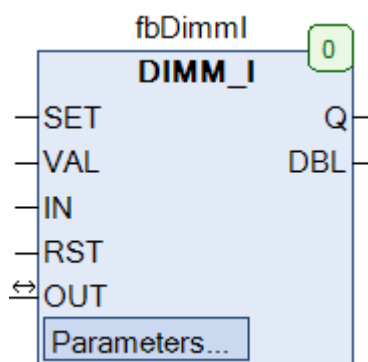


Рис. 6.5. Внешний вид ФБ DIMM\_I на языке CFC

Функциональный блок **DIMM\_I** представляет собой интерфейсный модуль для управления [диммером](#) с одной кнопкой. По переднему фронту на входе **SET** выход **Q** принимает значение **TRUE**, а сигнал управления диммером **OUT** – значение **VAL**. По переднему фронту на входе **RST** выходы блока принимают значение **FALSE**. Если параметр **RST\_OUT** имеет значение **TRUE**, то по переднему фронту на входе **RST** также происходит обнуление сигнала **OUT**.

Сигнал **Q** может использоваться в тех случаях, когда необходимо гарантировать отсутствие потенциала (например, в целях безопасности) на «выключенной» лампе. В этом случае для возможности отключения света устанавливается дополнительное промежуточное механическое реле, управляемое дискретным сигналом **Q**. Параметр **T\_ON\_MAX** определяет максимальное время активности выхода **Q**, после которого он сбрасывается в **FALSE**. Если **T\_ON\_MAX=T#0s**, то время активности выхода не ограничивается.

На вход **IN** подключается сигнал кнопки управления освещением. По умолчанию считается, что кнопка подключена с помощью нормально разомкнутого контакта. Если вход **IN** в течение времени **T\_reconfig** имеет значение **TRUE**, то контакт считается нормально замкнутым, и значение выхода **Q** будет являться инверсным по отношению к входу **IN**. Параметр **T\_debounce** определяет минимально фиксируемое время нажатия для игнорирования дребезга контактов.

При одиночном нажатии на кнопку **IN** значения выхода **Q** инвертируется, при этом если выход **Q** принимает значение **TRUE**, то значение входа-выхода **OUT** ограничивается диапазоном [**MIN\_ON...MAX\_ON**].

Если **Q=FALSE**, то при длительном (превышающем время **T\_DIMM\_START**) нажатии на кнопку **IN** уровень яркости **OUT** начинает плавно повышаться от текущего значения **VAL** (если **SOFT\_DIM=FALSE**) или от значения **MIN\_ON** (если **SOFT\_DIM=TRUE**) до значения **MAX\_ON**. Если **Q=TRUE**, то при длительном (превышающем время **T\_DIMM\_START**) нажатии на кнопку **IN** уровень яркости **OUT** плавно снижается до **0**. Параметр **T\_DIMM** определяет время изменения яркости. После начала изменения яркости кнопка **IN** может быть отпущена.

При двойном нажатии на кнопку **IN** выход **DBL** принимает значение **TRUE** на один цикл ПЛК. Если параметр **DBL\_toggle** имеет значение **TRUE**, то при двойном нажатии на кнопку **IN** выход **DBL** инвертируется.

## 6.6. F\_LAMP

Тип модуля: ФБ	Переменная	Тип	Описание
Входы	SWITCH	BOOL	TRUE – включить лампу.
	DIMM	BYTE	Требуемая яркость свечения (0...255).
	RST	BOOL	Передний фронт – выключить свет.
Выходы	LAMP	BYTE	Требуемая яркость свечения (0...255).
	STATUS	BYTE	ESR-код.
Входы-выходы	ONTIME	UDINT	Время наработки лампы, в секундах.
	CYCLES	UDINT	Число включений лампы.
Параметры	T_NO_DIMM	UINT	Время задержки диммирования, в часах.
	T_MAINTANCE	UINT	Предполагаемый срок эксплуатации, в часах.
Используемые модули	OSCAT_BASIC.ONTIME		

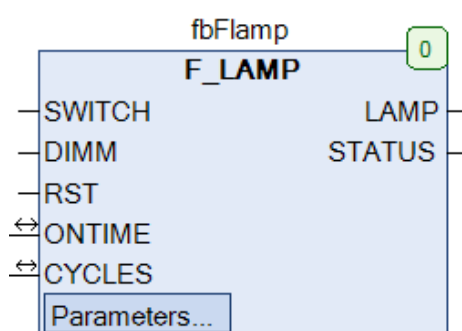


Рис. 6.6. Внешний вид ФБ F\_LAMP на языке CFC

Функциональный блок **F\_LAMP** представляет собой интерфейсный модуль для люминесцентной лампы. Пока вход **SWITCH** находится в состоянии **TRUE**, значение входа **DIMM** передается на выход **LAMP**. Параметр **T\_NO\_DIMM** определяет, через какое время после активации входа **SWITCH** выход **LAMP** начинает реагировать на изменения входа **DIMM**. Это необходимо для запрета диммирования на начальном периоде эксплуатации (значение по умолчанию – 100 часов), так как в этот период происходит изменение химического состава люминофора, и в случае диммирования срок эксплуатации лампы резко снизится. По переднему фронту входа **RST** выход **LAMP** принимает значение **0**.

Входы-выходы **ONTIME** и **CYCLES** содержит информацию о времени наработки лампы (в секундах) и количестве включений. Параметр **T\_MAINTANCE** определяет предполагаемый срок эксплуатации. По истечению этого срока выход **STATUS** принимает значение 120 (см. ниже).

Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**:

Значение выхода STATUS	Описание
110	Лампа выключена
111	Лампа включена, диммирование запрещено
112	Лампа включена, диммирование разрешено
120	Срок эксплуатации истек, требуется замена лампы

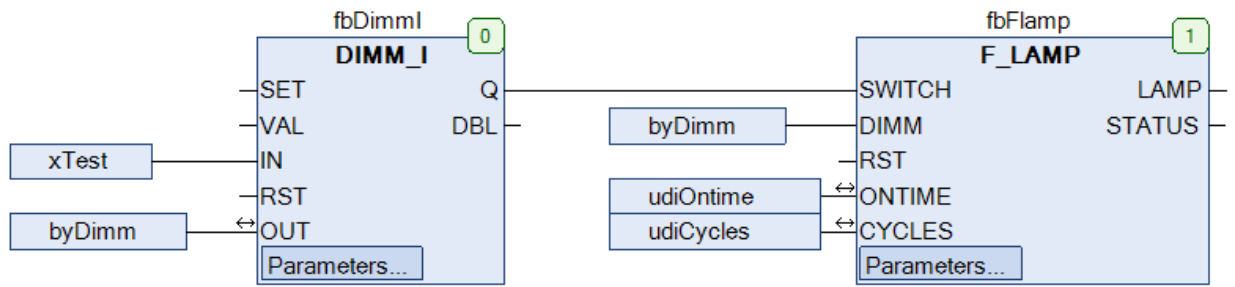
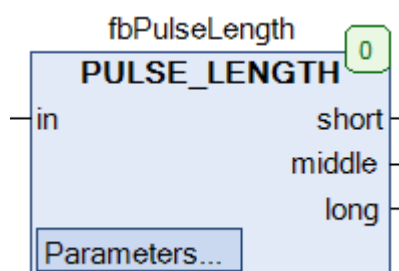


Рис. 6.7. Пример использования ФБ **F\_LAMP** совместно с ФБ **DIMM\_I** на языке CFC

## 6.7. PULSE\_LENGTH

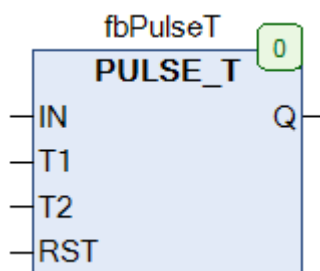
Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	in	BOOL	Состояние кнопки.
<b>Выходы</b>	short	BOOL	Единичный импульс при коротком нажатии.
	middle	BOOL	Единичный импульс при обычном нажатии.
	long	BOOL	Единичный импульс при длительном нажатии.
<b>Параметры</b>	T_short	TIME	Макс. время фиксации короткого нажатия.
	T_long	TIME	Мин. время фиксации длительного нажатия.
Используемые модули	OSCAT_BASIC.T_PLC_MS		

Рис. 6.8. Внешний вид ФБ **PULSE\_LENGTH** на языке CFC

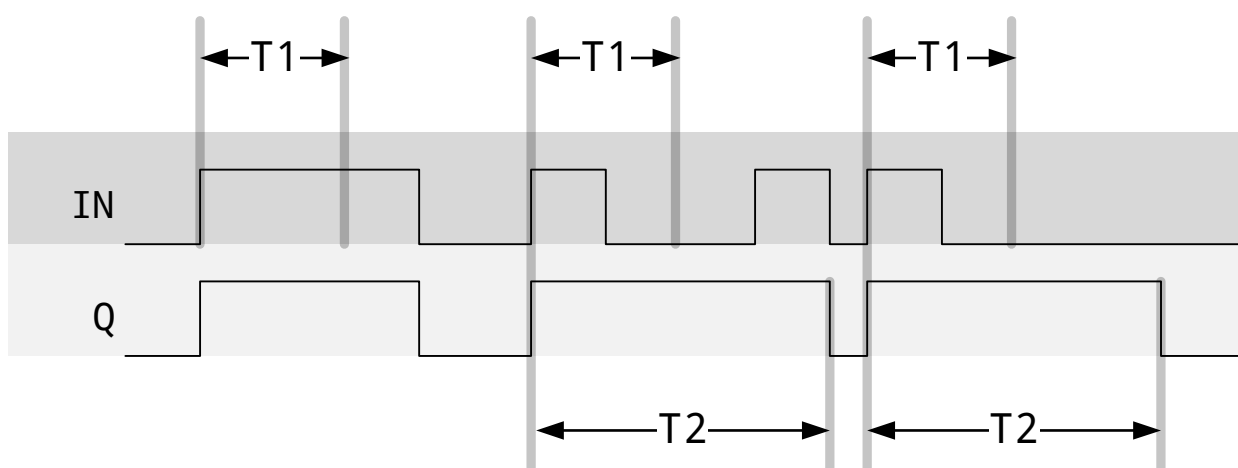
Функциональный блок **PULSE\_LENGTH** используется для обработки нажатий кнопки. Сигнал от кнопки подключается к входу **IN**. Если время нажатия кнопки не превысило времени **T\_short**, то после отпущения кнопки выход **short** принимает значение **TRUE** на один цикл ПЛК. Если время нажатия кнопки находится в интервале [**T\_short**...**T\_long**), то после отпущения кнопки выход **middle** принимает значение **TRUE** на один цикл ПЛК. Если время нажатия кнопки превышает **T\_long**, то выход **long** принимает значение **TRUE** и сохраняет его до тех пор, пока вход **IN** не примет значение **FALSE**.

## 6.8. PULSE\_T

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	IN	BOOL	Состояние кнопки.
	T1	TIME	Максимальное время нажатие.
	T2	TIME	Длительность импульса.
	RST	BOOL	Передний фронт – сброс выхода.
<b>Выходы</b>	Q	BOOL	Выходной сигнал.
Используемые модули	OSCAT_BASIC.T_PLC_MS		

Рис. 6.9. Внешний вид ФБ **PULSE\_T** на языке CFC

Функциональный блок **PULSE\_T** используется для генерирования импульсов по нажатию кнопки. Сигнал от кнопки подключается к входу **IN**. По переднему фронту входа **IN** выход **Q** принимает значение **TRUE**. Если кнопка отпускается до истечения времени **T1**, то выход **Q** сохраняет значение **TRUE** до истечения времени **T2**, после чего принимает значение **FALSE**. Если время нажатия кнопки превышает **T2**, то выход **Q** принимает значение **FALSE**. Отсчет интервалов **T1** и **T2** начинается по переднему фронту входа **IN**. По переднему фронту **RST** выход **Q** принимает значение **FALSE**.

Рис. 6.10. Трассировка работы ФБ **PULSE\_T**

## 6.9. SW\_RECONFIG

Тип модуля: ФБ	Переменная	Тип	Описание
Входы	IN	BOOL	Состояние кнопки.
	TD	TIME	Минимальное время нажатия.
	TR	TIME	Время определения типа кнопки (НО\НЗ).
Выходы	Q	BOOL	Выходное значение.

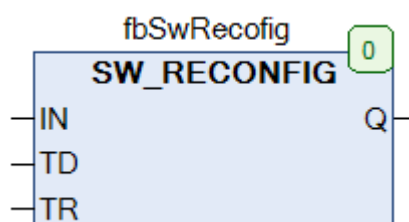


Рис. 6.11. Внешний вид ФБ SW\_RECONFIG на языке CFC

Функциональный блок **SW\_RECONFIG** представляет собой интерфейсный модуль для кнопки. На вход **IN** подключается дискретный сигнал от кнопки, который передается на выход **Q**. По умолчанию считается, что кнопка подключена с помощью нормально разомкнутого контакта. Если вход **IN** в течение времени **TR** имеет значение **TRUE**, то контакт считается нормально замкнутым, и значение выхода **Q** будет являться инверсным по отношению к входу **IN**. Если вход **IN** в течение времени **TR** имеет значение **FALSE**, то контакт считается нормально разомкнутым. Параметр **TD** определяет минимально фиксируемое время нажатия для игнорирования дребезга контактов.

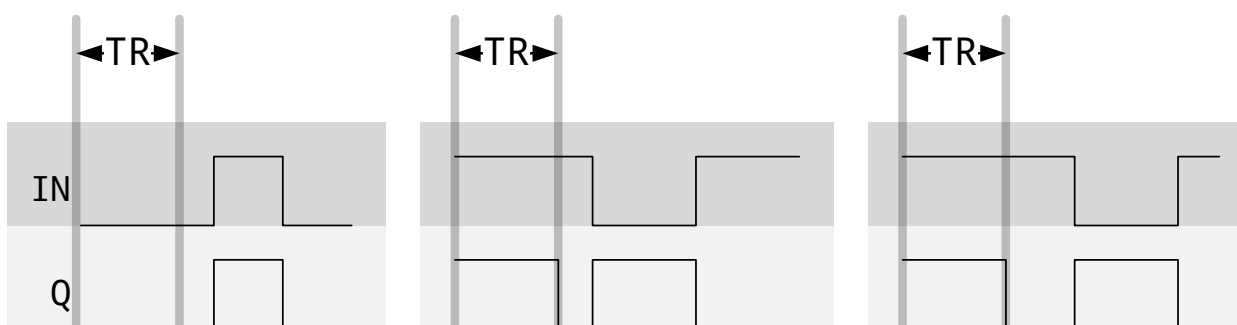


Рис. 6.12. Трассировка работы ФБ SW\_RECONFIG

## 6.10. SWITCH\_I

Тип модуля: ФБ	Переменная	Тип	Описание
Входы	set	BOOL	Передний фронт – принудительное включение.
	in	BOOL	Состояние кнопки.
	rst	BOOL	Передний фронт – принудительное отключение.
Выходы	Q	BOOL	Текущее состояние.
Параметры	T_debounce	TIME	Минимальное время нажатия.
	T_reconfig	TIME	Время определения типа кнопки (НО\НЗ).
	T_on_max	TIME	Максимальное время активности выхода (T#0s – время не ограничено).
Используемые модули	OSCAT_BASIC.T_PLC_MS		

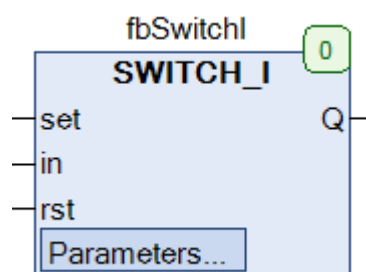


Рис. 6.13. Внешний вид ФБ SWITCH\_I на языке CFC

Функциональный блок **SWITCH\_I** представляет собой интерфейсный модуль для кнопки. На вход **in** подключается дискретный сигнал от кнопки, который передается на выход **Q**. По умолчанию считается, что кнопка подключена с помощью нормально разомкнутого контакта. Если вход **in** в течение времени **T\_reconfig** имеет значение **TRUE**, то контакт считается нормально замкнутым, и значение выхода **Q** будет являться инверсным по отношению к входу **in**. Параметр **T\_debounce** определяет минимально фиксируемое время нажатия для игнорирования дребезга контактов. Параметр **T\_on\_max** определяет максимальное время активности выхода **Q**. По истечению этого времени выход **Q** принимает значение **FALSE**. Если **T\_ON\_MAX=T#0s**, то время активности выхода не ограничивается.

По переднему фронту входа **set** выход **Q** принудительно устанавливается в состояние **TRUE**. По переднему фронту входа **rst** выход **Q** принимает значение **FALSE** независимо от значений других входов блока.



## 6.11. SWITCH\_X

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	IN1...IN6	BOOL	Состояние кнопок.
<b>Выходы</b>	Q1...Q2	BOOL	Импульс при отпускании кнопки IN1...IN2.
	Q3...Q6	BOOL	Флаг «нажатие кнопки IN3...IN6».
	Q31...Q62	BOOL	Флаг «нажатие двух кнопок».
<b>Параметры</b>	t_debounce	TIME	Минимальное время нажатия.

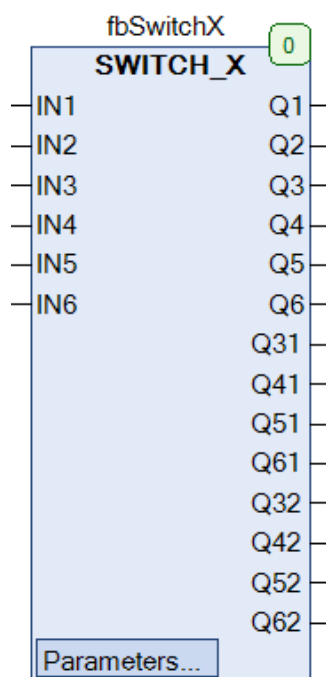


Рис. 6.14. Внешний вид ФБ SWITCH\_X на языке CFC

Функциональный блок **SWITCH\_X** представляет собой интерфейсный модуль для шестикнопочной клавиатуры. Сигналы кнопок подключаются к входам **IN1...IN6**. В каждый момент времени активен только один из выходов блока.

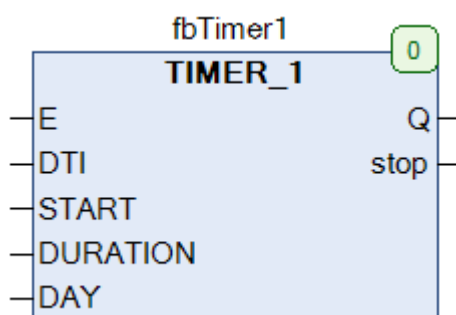
Если активен только один из входов **IN3...IN6**, то соответствующий выход **Q3...Q6** имеет значение **TRUE**. По заднему фронту входа **IN1...IN2** на выходе **Q1...Q2** генерируется единичный импульс (при условии, что остальные входы имеют значение **FALSE**). При одновременном нажатии одной из кнопок групп **IN1...IN2** и **IN3...IN6** соответствующий выход принимает значение **TRUE**. Например, при одновременном нажатии **IN1** и **IN3** активируется выход **Q31**. Это может использоваться для создания комбинаций клавиш.

При одновременном нажатии нескольких кнопок обрабатываются те, которые имеют наименьший номер. Например, при одновременном нажатии **IN3** и **IN4** будет активирован выход **Q3**. При одновременном нажатии **IN1**, **IN2**, **IN4** и **IN5** будет активирован выход **Q41**.

Параметр **t\_debounce** определяет минимально фиксируемое время нажатия для игнорирования дребезга контактов.

## 6.12. TIMER\_1

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	E	BOOL	TRUE – разрешить работу блока.
	DTI	DT	Текущие дата и время.
	START	TOD	Время начала генерации импульса.
	DURATION	TIME	Длительность импульса.
	DAY	BYTE	Маска дней недели.
<b>Выходы</b>	Q	BOOL	Выход блока.
	stop	TIME	Время завершения импульса.
Используемые модули	OSCAT_BASIC.TIMECHECK, OSCAT_BASIC.DAY_OF_WEEK		

Рис. 6.15. Внешний вид ФБ **TIMER\_1** на языке CFC

Функциональный блок **TIMER\_1** используется для реализации недельного таймера, который генерирует на выходе **Q** импульс длительностью **DURATION** в выбранное время суток **START** определенных дней недели. Вход **DAY** содержит битовую маску дней недели, в которые будут генерироваться импульсы. Самый младший бит соответствует воскресенью. Примеры задания значений входа:

- DAY = 2#0000\_0000 – импульсы никогда не генерируются;
- DAY = 2#0100\_0000 – импульс генерируются только в понедельник;
- DAY = 2#0110\_0000 – импульсы генерируется в понедельник и вторник;
- DAY = 2#0000\_1000 – импульс генерируется только в четверг;
- DAY = 2#0111\_1111 – импульсы генерируются каждый день.

На вход **DTI** подается значение системного времени.

Вход **E** управляет работой блока. Если он имеет значение **FALSE**, то блок остановлен и выход **Q** имеет значение **FALSE**. Если вход **E** имеет значение **TRUE**, то блок запускается в работу. Если в момент запуска блока текущее время суток превышает значение **START**, но не превышает **START + DURATION**, то импульс начнет генерироваться сразу после включения (при условии, что текущий день недели выбран в битовой маске **DAY**).

Выход **STOP** определяет время суток, в которое будет завершен импульс.

Если **duration=0**, то выход **Q** принимает значение **TRUE** на один цикл ПЛК.

Ниже приведен пример работы с блоком на языке CFC. Функция **SysRtcGetTime** из библиотеки **SysRtc23** используется для получения текущего системного времени ПЛК.

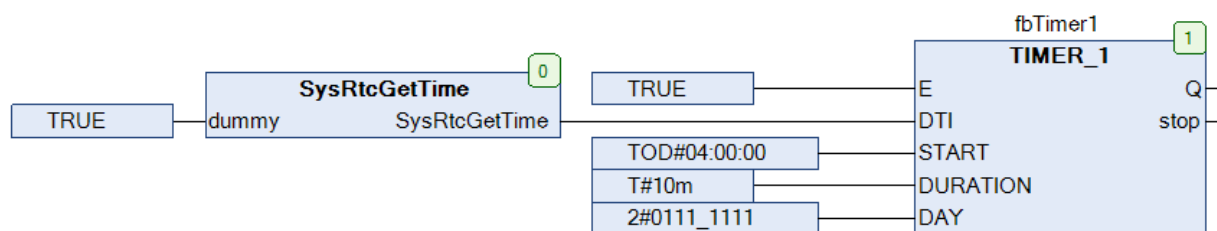


Рис. 6.16. Пример работы с ФБ **TIMER\_1** на языке CFC

### 6.13. TIMER\_2

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	DT_in	DT	Текущие дата и время.
	start	TOD	Время начала генерации импульса.
	duration	TIME	Длительность импульса.
	mode	BYTE	Режим работы блока.
	HOLIDAY	BOOL	TRUE – сегодня праздник.
<b>Выходы</b>	Q	BOOL	Выход блока.
Используемые модули	OSCAT_BASIC.T_PLC_MS, OSCAT_BASIC.DAY_OF_WEEK, OSCAT_BASIC.DAY_OF_MONTH, OSCAT_BASIC.MONTH_OF_DATE, OSCAT_BASIC.DAY_OF_YEAR		

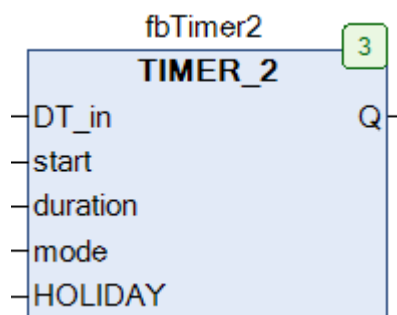


Рис. 6.17. Внешний вид ФБ **TIMER\_2** на языке CFC

Функциональный блок **TIMER\_2** используется для реализации календарного таймера, который генерирует на выходе **Q** импульс длительностью **duration** в выбранное время суток **start** определенных дней. На вход **DT\_in** подается значение системного времени, на вход **HOLIDAY** – логический сигнал, принимающий значение **TRUE** в дни праздников (можно использовать выход блока **HOLIDAY** из библиотеки **OSCAT Basic**).

Если **duration=0**, то выход **Q** принимает значение **TRUE** на один цикл ПЛК.

Вход **mode** определяет дни работы таймера:

mode	Описание
0	никогда
1	только по понедельникам
2	только по вторникам
3	только по средам
4	только по четвергам
5	только по пятницам
6	только по субботам
7	только по воскресеньям
11	каждый день
12	раз в два дня
13	раз в три дня
14	раз в четыре дня
15	раз в пять дней
16	раз в шесть дней
20	в будни (с понедельника по пятницу)
21	в выходные (субботу и воскресенье)
22	в будни, за исключением праздников
23	в выходные и праздники
24	только по праздникам
25	в первый день месяца
26	в последний день месяца
27	в последний день года (31 декабря)
28	в первый день года (1 января)

Ниже приведен пример работы с блоком на языке CFC. Функция **SysRtcGetTime** из библиотеки **SysRtc23** используется для получения текущего системного времени ПЛК. ФБ **HOLIDAY** из библиотеки **OSCAT Basic** используется для определения праздничных дней.

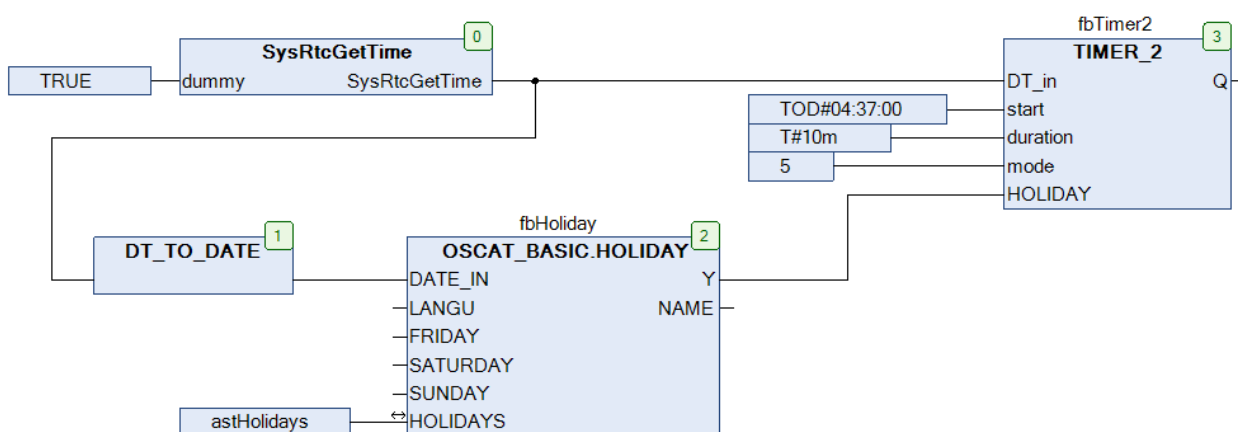
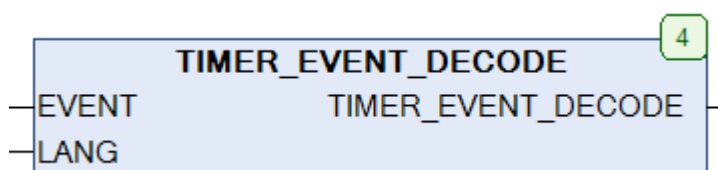


Рис. 6.18. Пример работы с ФБ **TIMER\_2** на языке CFC

## 6.14. TIMER\_EVENT\_DECODE

Тип модуля: функция	Переменная	Тип	Описание
Входы	EVENT	STRING	Параметры события в виде строки.
	LANG	INT	Язык приложения.
Выходы	TIMER_EVENT_DECODE	OSCAT_BASIC.TIMER_EVENT	Параметры события в виде структуры.
Используемые модули	OSCAT_BASIC.TIMER_EVENT, OSCAT_BASIC.FSTRING_TO_BYTE, OSCAT_BASIC.FSTRING_TO_WEEK, OSCAT_BASIC.FSTRING_TO_WEEKDAY, OSCAT_BASIC.IS_CC		

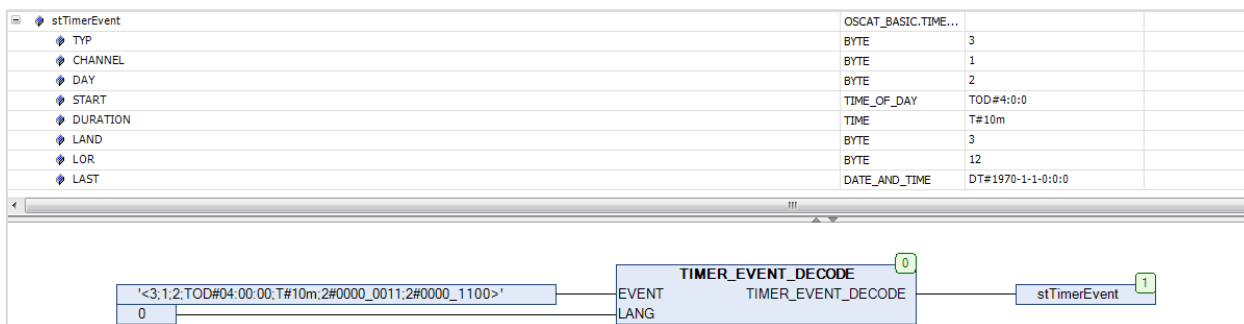
Рис. 6.19. Внешний вид функции **TIMER\_EVENT\_DECODE** на языке CFC

Функция **TIMER\_EVENT\_DECODE** конвертирует строку с описанием параметров события **EVENT** в структуру **TIMER\_EVENT**, которая объявлена в библиотеке **OSCAT Basic** и используется в ФБ [TIMER\\_P4](#). Вход **LANG** определяет язык приложения, который используется при указании дней недели. Допустимые названия дней содержатся в библиотеке **OSCAT Basic** в структуре **CONSTANTS\_LANGUAGE**. Описание элементов структуры **TIMER\_EVENT** приведено в [п. 6.16](#).

Строка **EVENT** должна иметь следующий формат:

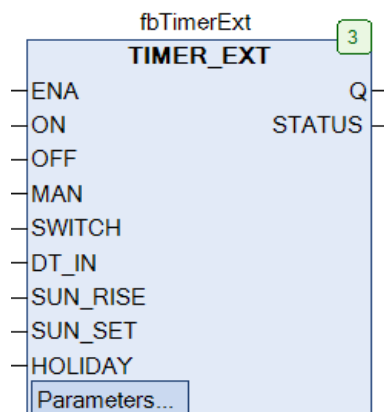
<TYPE; CHANNEL; DAY; START; DURATION; LAND; LOR>

Заполнитель	Допустимые форматы
TYPE	'123', '2#0101', '8#33', '16#FF'
CHANNEL	'123', '2#0101', '8#33', '16#FF'
DAY	'123', '2#0101', '8#33', '16#FF', 'Mo' 'MO', 'DI', 'DO'
START	'TOD#12:00'
DURATION	'T#1h3m22s'
LAND	'123', '2#0101', '8#33', '16#FF'
LOR	'123', '2#0101', '8#33', '16#FF'

Рис. 6.20. Пример работы с функцией **TIMER\_EVENT\_DECODE** на языке CFC

## 6.15. TIMER\_EXT

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	ENA	BOOL	TRUE – разрешить работу блока.
	ON	BOOL	Передний фронт – включить выход.
	OFF	BOOL	Передний фронт – отключить выход.
	MAN	BOOL	Состояние выхода в режиме ручного управления (ON = TRUE, OFF = TRUE).
	SWITCH	BOOL	Передний фронт - переключение выхода в автоматическом режиме (ON = FALSE, OFF = FALSE).
	DT_IN	DT	Текущие дата и время.
	SUN_RISE	TOD	Время восхода солнца.
	SUN_SET	TOD	Время заката солнца.
	HOLIDAY	BOOL	TRUE – сегодня праздник.
<b>Выходы</b>	Q	BOOL	Выход блока.
	STATUS	BYTE	ESR-код.
<b>Параметры</b>	T_RISE_START	TIME	Включить выход за это время до восхода.
	T_RISE_STOP	TIME	Отключить выход спустя это время после восхода.
	T_SET_START	TIME	Включить выход за это время до заката.
	T_SET_STOP	TIME	Отключить выход спустя это время после заката.
	T_DAY_START	TOD	Включить выход в это время суток.
	T_DAY_STOP	TOD	Отключить выход в это время суток.
	T_DEBOUNCE	TIME	Минимальное время нажатия SWITCH.
	ENABLE_SATURDAY	BOOL	TRUE – разрешить работу блока в субботу.
	ENABLE_SUNDAY	BOOL	TRUE – разрешить работу блока в воскресенье.
	ENABLE_HOLIDAY	BOOL	TRUE – разрешить работу блока в праздничные дни.
Используемые модули	DEBOUNCE, OSCAT_BASIC.MANUAL_2, OSCAT_BASIC.T_PLC_MS, OSCAT_BASIC.DAY_OF_WEEK		

Рис. 6.21. Внешний вид ФБ **TIMER\_EXT** на языке CFC

Функциональный блок **TIMER\_EXT** используется для реализации суточного таймера. Пока вход **ENA** имеет значение **TRUE**, блок находится в работе.

Выход блока **Q** активируется автоматически в заданное время суток, а также может управляться вручную с помощью входов блока. По переднему фронту входа **ON** выход **Q** принимает значение **TRUE**. По переднему фронту входа **OFF** выход **Q** принимает значение **FALSE**. Если входы **ON** и **OFF** одновременно имеют значение **TRUE**, то значение выхода **Q** определяется входом **MAN**. Если входы **ON** и **OFF** одновременно имеют значение **FALSE**, то по нажатию кнопки, подключенной к входу **SWITCH**, значение выхода **Q** инвертируется. Параметр **T\_DEBOUNCE** определяет минимальное фиксируемое время нажатия.

На вход **DT\_IN** подается значение системного времени, на вход **HOLIDAY** – логический сигнал, принимающий значение **TRUE** в дни праздников (можно использовать выход блока **HOLIDAY** из библиотеки **OSCAT Basic**). На входы **SUN\_RISE** и **SUN\_SET** подается время восхода и заката солнца для заданной широты и долготы (можно использовать выходы блока **SUN\_TIME** из библиотеки **OSCAT\_BASIC**).

Параметры **T\_RISE\_START** и **T\_RISE\_STOP** определяют период активации выхода блока **Q** относительно времени восхода (выход активируется за время **T\_RISE\_START** до восхода и деактивируется спустя время **T\_RISE\_STOP** после восхода). Параметры **T\_SET\_START** и **T\_SET\_STOP** определяют период активации выхода блока **Q** относительно времени заката (выход активируется за время **T\_SET\_START** до заката и деактивируется спустя время **T\_SET\_STOP** после заката). Параметры **T\_DAY\_START** и **T\_DAY\_STOP** определяют период активации выхода блока **Q** между восходом и закатом. Параметры **ENABLE\_SATURDAY**, **ENABLE\_SUNDAY** и **ENABLE\_HOLIDAY** определяют, выполняется ли активация блока в субботу, воскресенье и праздники.

Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**. Ниже описана зависимость основных входов и выходов блока:

ENA	ON	OFF	MAN	SWITCH	Таймер	Q	STATUS
FALSE	-	-	-	-	-	FALSE	104
TRUE	TRUE	FALSE	-	-	-	TRUE	101
TRUE	FALSE	TRUE	-	-	-	FALSE	102
TRUE	TRUE	TRUE	MAN		-	MAN	103
TRUE	FALSE	FALSE	-	TRUE	-	NOT(Q)	110
TRUE	FALSE	FALSE	-	-	TOD = T_DAY_START	TRUE	111
TRUE	FALSE	FALSE	-	-	TOD = T_DAY_STOP	FALSE	112
TRUE	FALSE	FALSE	-	-	TOD = SUN_RISE – T_RISE_START	TRUE	113
TRUE	FALSE	FALSE	-	-	TOD = SUN_RISE + T_RISE_STOP	FALSE	114
TRUE	FALSE	FALSE	-	-	TOD = SUN_SET – T_SET_START	TRUE	115
TRUE	FALSE	FALSE	-	-	TOD = SUN_SET + T_SET_STOP	FALSE	116

Ниже приведен пример работы с блоком на языке CFC. Функция **SysRtcGetTime** из библиотеки **SysRtc23** используется для получения текущего системного времени ПЛК. ФБ **HOLIDAY** из библиотеки **OSCAT Basic** используется для определения праздничных дней. ФБ **SUN\_TIME** из библиотеки **OSCAT Basic** используется для определения времени восхода и заката для заданной широты и долготы.

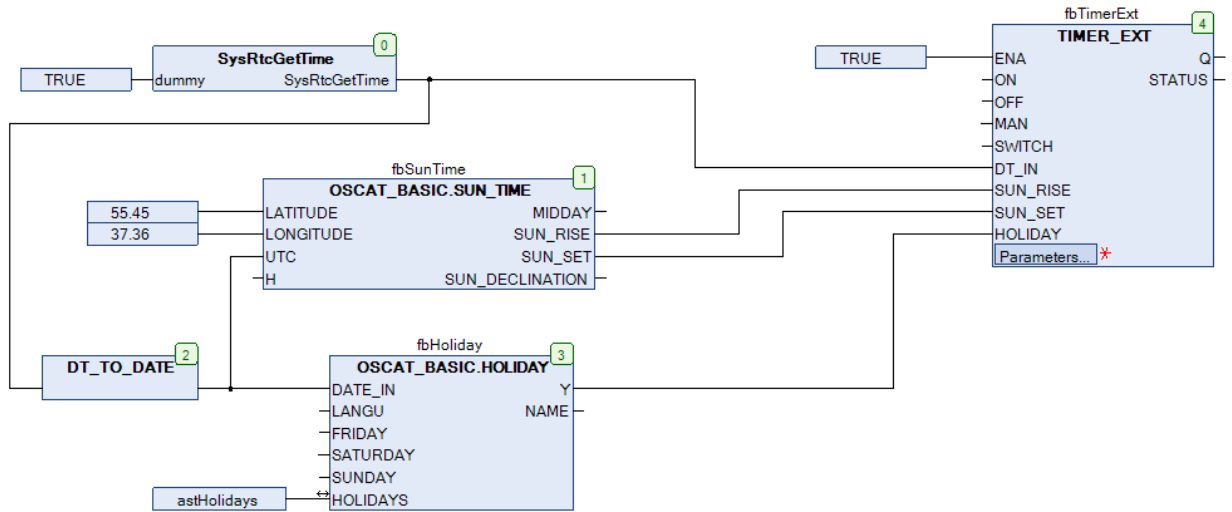
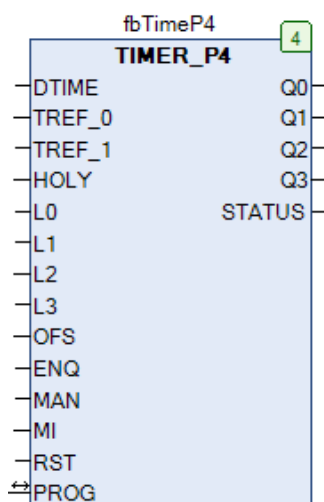


Рис. 6.22. Пример работы с ФБ **TIMER\_EXT** на языке CFC



## 6.16. TIMER\_P4

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	DTIME	DT	Текущие дата и время.
	TREF_0	TOD	Базовое время 0 для режимов 50-55.
	TREF_1	TOD	Базовое время 1 для режимов 50-55.
	HOLY	BOOL	TRUE – сегодня праздник.
	L0...L3	BOOL	Доп. логические сигналы.
	OFS	INT	Смещение номера канала.
	ENQ	BOOL	TRUE – разрешить работу блока.
	MAN	BOOL	TRUE – ручной режим управления выходами.
	MI	BYTE	Битовая маска выходов в ручном режиме (бит 0 соответствует Q0).
	RST	BOOL	Передний фронт – очистить события.
<b>Выходы</b>	Q0...Q3	BOOL	Выходы блока.
	STATUS	BYTE	ESR-код.
<b>Входы-выходы</b>	PROG	<sup>3</sup>	Массив событий.
Используемые модули	OSCAT_BASIC.TIMER_EVENT, OSCAT_BASIC.DAY_OF_WEEK, OSCAT_BASIC.DAY_OF_MONTH, OSCAT_BASIC.MONTH_END, OSCAT_BASIC.DAY_OF_YEAR, OSCAT_BASIC.YEAR_END, OSCAT_BASIC.YEAR_OF_DATE, OSCAT_BASIC.LEAP_DAY,		

Рис. 6.23. Внешний вид ФБ **TIMER\_P4** на языке CFC

Функциональный блок **TIMER\_P4** представляет собой гибко настраиваемый таймер, который активируется при наступлении созданных пользователем событий, а также имеет специальные логические входы для задания нужного алгоритма работы. События сохраняются с помощью входа-выхода **PROG**, который представляет собой массив структур **TIMER\_EVENT**. Структура **TIMER\_EVENT** объявлена в библиотеке **OSCAT Basic**. Размер массива определяется константой **array\_max**, которая может быть изменена пользователем путем редактирования исходного кода ФБ (значение по умолчанию – **63**).

<sup>3</sup> ARRAY[0..array\_max] OF OSCAT\_BASIC.TIMER\_EVENT

Структура `TIMER_EVENT` состоит из следующих элементов:

Элемент	Тип	Описание
CHANNEL	BYTE	Номер канала (см. пояснения ниже).
TYPE	BYTE	Тип события (см. таблицу ниже)
DAY	BYTE	День наступления события (в зависимости от события – маска, число и т.д.)
START	TOD	Время начала генерации импульса.
DURATION	TIME	Длительность импульса.
LAND	BYTE	Битовая маска для операции AND.
LOR	BYTE	Битовая маска для операции OR.
LAST	DWORD	Дополнительная переменная, используемая ФБ для хранения данных.

Одним из параметров события является канал, который представляет собой номер выхода блока, активируемый при наступлении события. По умолчанию блок имеет 4 канала (**Q0...Q3**), но это количество может быть изменено (потребуется отредактировать количество выходов блока, константу **channel\_max** и добавить нужные строки после комментария (*\* set the outputs \**)).

Параметр **TYPE** определяет тип события. Типы событий описаны в конце пункта. Параметр **DAY** определяет день наступления события. В зависимости от типа события он может представлять собой битовую маску дней недели, номер дня месяца или года и т.д. Параметр **START** определяет время наступления события, а параметр **DURATION** – его продолжительность. Параметры **LAND** и **LOR** содержат битовые маски для операций AND и OR, в которых также используются входы блока **L0...L3**. Подробная информация по этому поводу приедена ниже. Параметр **LAST** используется как ячейка памяти блока и не должен изменяться из программы пользователя.

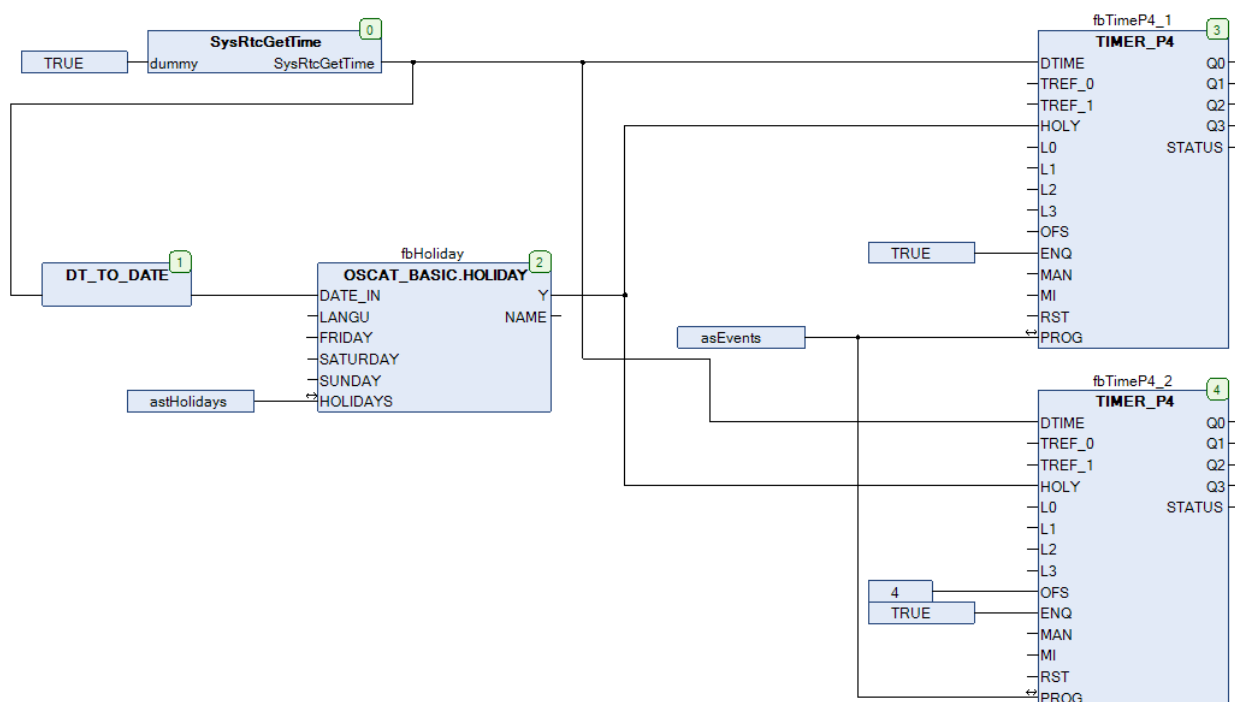
Пока вход **ENQ** имеет значение **TRUE**, блок находится в работе. На вход **DTIME** подается значение системного времени, на вход **HOLIDAY** – логический сигнал, принимающий значение **TRUE** в дни праздников (можно использовать выход блока **HOLIDAY** из библиотеки **OSCAT Basic**). Входы **TREF\_0** и **TREF\_1** используются для задания дополнительных меток времени, используемых в событиях типа 50-55. Если вход **MAN** имеет значение **TRUE**, то управления выходами блока осуществляется напрямую с помощью с помощью битовой маски **MI** (младший бит маски соответствует выходу **Q0**).

Вход **OFS** позволяет организовать смещение номера канала. Это требуется для синхронизации работы нескольких экземпляров. На рисунке ниже оба экземпляра работают с одним массивом событий **astEvents**, при этом блок **fbTimerP4\_1** соответствуют каналы **0-3**, а блоку **fbTimerP4\_2** – каналы **4-7**.

По переднему фронту входа **RST** происходит очистка массива событий.

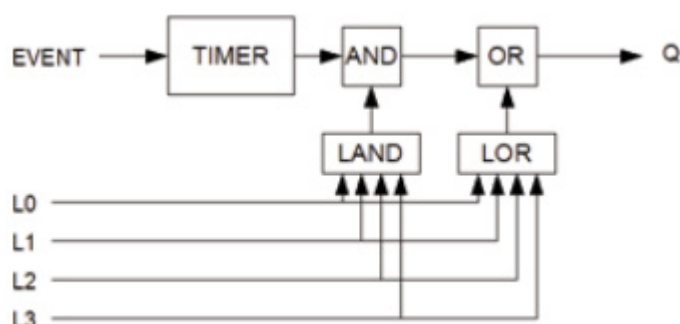
Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**.

STATUS	Описание
100	Блок отключен (ENQ=FALSE)
101	Блок в режиме ручного управления (ENQ=TRUE, MAN=TRUE)
102	Блок в режиме автоматического управления (ENQ=TRUE, MAN=FALSE)

Рис. 6.24. Пример работы с ФБ **TIMER\_P4** на языке CFC

Входы **L0...L3** используются для определения дополнительных условий активации выхода таймера. По умолчанию активация происходит при наступлении заданного события. Параметры события **LAND** (Logical AND) и **LOR** (Logical OR) определяют маску входов для соответствующих логических операций (младший бит маски соответствует **L0**). Например, если **LAND = 2#0000\_0011** и **LOR = 2#0000\_1100**, то выход таймера при наступлении события будет активирован только в том случае, если входы **L0** и **L1** имеют значение **TRUE**; при этом если входы **L3** и **L4** имеют значение **TRUE**, то таймер будет активирован даже в случае отсутствия события.

Структурная схема логического контура таймера приведена ниже:

Рис. 6.25. Структурная схема логического контура ФБ **TIMER\_P4**

Возможные типы событий описаны ниже:

TYPE	DAY	Время наступления	Длительность	Событие происходит...
1	-	START	DURATION	...каждый день
2	Битовая маска дней недели	START	DURATION	...в заданные дни недели
3	N	START	DURATION	...раз в N дней
10	Номер дня недели	START	DURATION	...раз в неделю
20	Номер дня месяца	START	DURATION	...раз в месяц
21	-	START	DURATION	...в последний день месяца
30	Номер дня года	START	DURATION	...раз в год
31	-	START	DURATION	...в последний день года
40	-	START	DURATION	...в високосные дни
41	-	START	DURATION	...в праздники
42	-	START	DURATION	...в праздники и выходные
43	-	START	DURATION	...каждый будний день
50	0, 1	OFS	DURATION	
51	0, 1	-OFS	DURATION	
52	0, 1, 2	OFS		
53	0, 1, 2	OFS		
54	0, 1, 2	OFS		
55	0, 1, 2	OFS		

Более подробное описание типов событий:

1. Событие происходит каждый день. Значение параметра **DAY** не учитывается.

2. Событие происходит в заданные дни недели. Параметр **DAY** содержит битовую маску дней недели, в которой самый младший бит соответствует воскресенью. Примеры задания значений параметра:

- DAY = 2#0000\_0000 – импульсы никогда не генерируются;
- DAY = 2#0100\_0000 – импульсы генерируются только в понедельник;
- DAY = 2#0110\_0000 – импульсы генерируются в понедельник и вторник;
- DAY = 2#0000\_1000 – импульс генерируется только в четверг;
- DAY = 2#0111\_1111 – импульсы генерируются каждый день.

3. Событие происходит раз в N дней. Значение N записывается в параметр **DAY**.

10. Событие происходит раз в неделю. Номер нужного дня недели записывается в параметр **DAY**.

20. Событие происходит раз в месяц. Номер нужного месяца записывается в параметр **DAY**.

21. Событие происходит в последний день каждого месяца. Значение параметра **DAY** не учитывается.

30. Событие происходит раз в год. Номер нужного дня года записывается в параметр **DAY**.

**31.** Событие происходит в последний день каждого года (31 декабря). Значение параметра **DAY** не учитывается.

**40.** Событие происходит в високосные дни (29 февраля). Значение параметра **DAY** не учитывается.

**41.** Событие происходит в праздники. Вход **HOLY** в эти дни должен иметь значение **TRUE**. Значение параметра **DAY** не учитывается.

**42.** Событие происходит в праздники и выходные. Вход **HOLY** в праздничные дни должен иметь значение **TRUE**. Значение параметра **DAY** не учитывается.

**43.** Событие происходит каждый будний день (с понедельника по пятницу). Значение параметра **DAY** не учитывается.

В описании формул событий 50-55 используется параметр **TREF\_X**, который зависит от значения параметра **DAY**:

- если  $DAY=0$ , то  $TREF\_X = TREF\_0$  (вход ФБ);
- если  $DAY=1$ , то  $TREF\_X = TREF\_1$  (вход ФБ);
- если  $DAY>1$ , то  $TREF\_X = 00:00:00$ .

**50.** Событие происходит каждый день в момент времени X:

$$X = TREF\_X + START$$

**51.** Событие происходит каждый день в момент времени X:

$$X = TREF\_X - START$$

**52.** Выход блока активируется в момент времени X:

$$X = TREF\_X + START$$

Деактивация выхода происходит после обработки другого события или с помощью событий 52 или 55.

**53.** Выход блока деактивируется в момент времени X:

$$X = TREF\_X + START$$

**54.** Выход блока активируется в момент времени X:

$$X = TREF\_X - START$$

Деактивация выхода происходит после обработки другого события или с помощью событий 52 или 55.

**55.** Выход блока деактивируется в момент времени X:

$$X = TREF\_X - START$$

## 7. Управление жалюзи

### 7.1. Вступление

Функциональные блоки, описанные в данной главе, представляют собой набор модулей для управления светозащитными устройствами. Они используются для управления жалюзи, внутренними ставнями и шторами. Блоки позволяют создать гибкую, легко расширяемую систему управления, которая может быть адаптирована под конкретные задачи.

Все ФБ имеют входы **UP/DN**, которые используются для ручного открытия/закрытия жалюзи (штор, ставней). Соответствующие им управляющие выходы **QU/QD** подключаются к входам **UP/DN** следующего блока. Выходы **QU/QD** последнего блока программы подключатся к дискретным выходам, которые связаны с исполнительным механизмом.

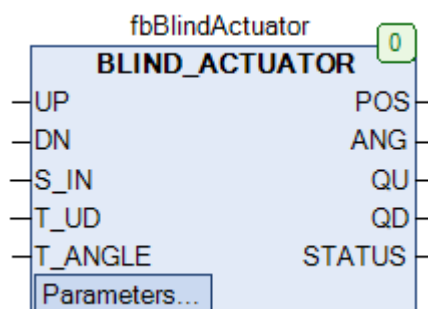
Если входы **UP** и **DN** одновременно имеют значение **TRUE**, то блок переходит в автоматический режим управления. В этом случае состояние выходов зависит от значений входов **PI** (требуемое положение жалюзи) и **AI** (требуемый угол наклона ламелей).

Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**.

Пример системы управления жалюзи с использованием большинства модулей, описанных в данном разделе, приведен на рис. 7.8.

## 7.2. BLIND\_ACTUATOR

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	UP	BOOL	Команда ручного управления (открытие).
	DN	BOOL	Команда ручного управления (закрытие).
	S_IN	BYTE	ESR-код предыдущего модуля.
	T_UD	BYTE	Время полного открытия жалюзи.
	T_ANGLE	BYTE	Время полного наклона ламелей.
<b>Выходы</b>	POS	BYTE	Положение жалюзи (0 – полностью закрыто, 255 – полностью открыто).
	ANG	BYTE	Угол наклона ламелей (0 – вертикально, 255 – горизонтально).
	QU	BOOL	Сигнал для ИМ (открытие).
	QD	BOOL	Сигнал для ИМ (закрытие).
	STATUS	BYTE	ESR-код.
<b>Параметры</b>	T_LOCKOUT	TIME	Задержка при переключении направления.
Используемые модули	OSCAT_BASIC.RMP_B, OSCAT_BASIC.INTERLOCK		

Рис. 7.1. Внешний вид ФБ **BLIND\_ACTUATOR** на языке CFC

Функциональный блок **BLIND\_ACTUATOR** представляет собой интерфейсный модуль для исполнительного механизма, управляющего жалюзи или ставнями. Пока вход **UP** имеет значение **TRUE**, происходит открытие жалюзи: сигнал управления ИМ **QU** принимает значение **TRUE**, значение выхода **ANG** (угол наклона ламелей) начинается увеличиваться со скоростью  $(255/T\_ANGLE)$  ед./с; после того, как **ANG** достигнет **255**, значение выхода **POS** (положение жалюзи) начинает увеличиваться со скоростью  $(255/T\_UD)$  ед./с. Аналогичным образом вход **DN** используется для закрытия жалюзи. Входы **UP/DN** не должны принимать значение **TRUE** одновременно – в этом случае блок не включается в работу.

Выходы **POS** и **ANG** не отражают реальное состояние жалюзи, и вычисляются на основе значений входов **T\_UD** и **T\_ANGLE**, которые должны быть определены экспериментально для конкретной модели жалюзи.

Параметр **T\_LOCKOUT** определяет время блокировки блока при изменении направления движения жалюзи. Например, если **T\_LOCKOUT = T#5s**, то после остановки открытия жалюзи сигнал на закрытие жалюзи начнет обрабатываться после 5-секундной задержки.

Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**. Вход **S\_IN** содержит ESR-код, полученный от другого модуля. Он транслируется на выход **STATUS** в тех случаях, когда блок не имеет собственных сообщений.

STATUS	Описание
0	Нет сообщений
1	Ошибка управления (одновременно активны входы UP и DN)
101	Жалюзи открываются
102	Жалюзи закрываются
S_IN	Передача сообщения от другого блока

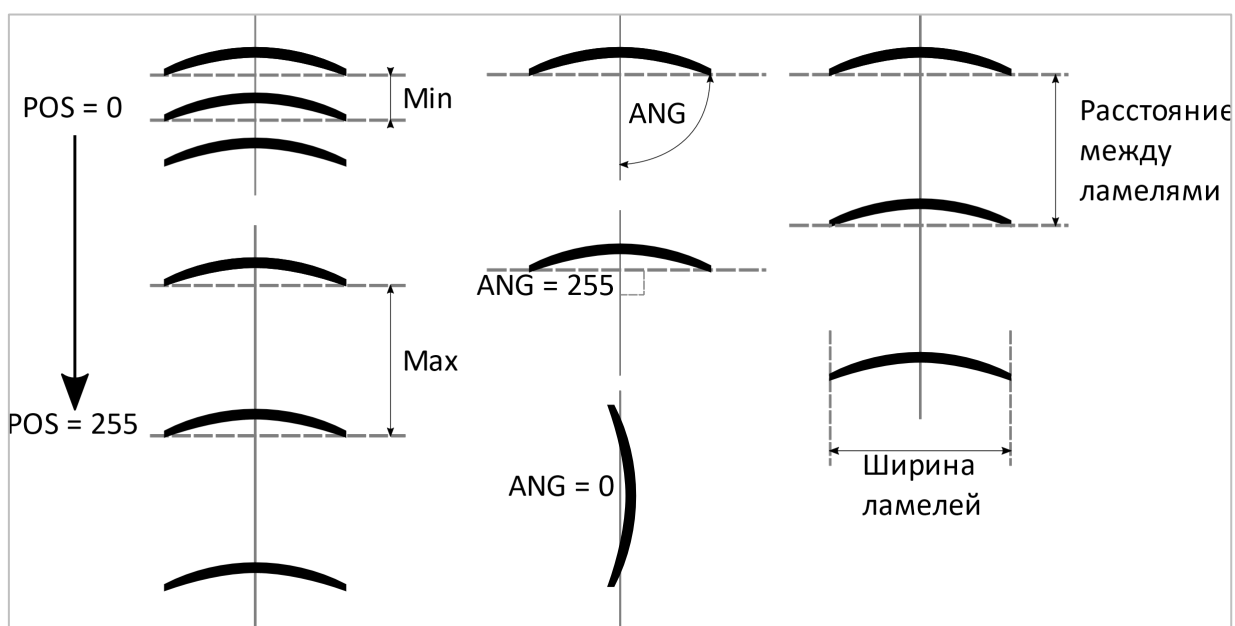
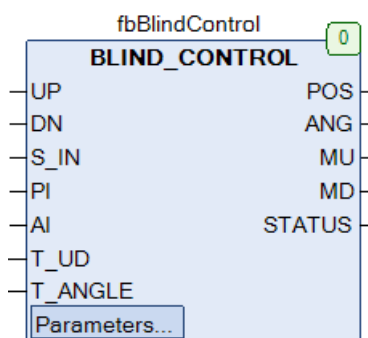


Рис. 7.2. Схема управления жалюзи



## 7.3. BLIND\_CONTROL

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	UP	BOOL	Команда ручного управления (открытие).
	DN	BOOL	Команда ручного управления (закрытие).
	S_IN	BYTE	ESR-код предыдущего модуля.
	PI	BYTE	Требуемая степень открытия жалюзи (0 – полностью закрыто, 255 – полностью открыто).
	AI	BYTE	Требуемый угол наклона ламелей (0 – вертикально, 255 – горизонтально).
	T_UD	BYTE	Время полного открытия жалюзи.
	T_ANGLE	BYTE	Время полного наклона ламелей.
<b>Выходы</b>	POS	BYTE	Положение жалюзи (0 – полностью закрыто, 255 – полностью открыто).
	ANG	BYTE	Угол наклона ламелей (0 – вертикально, 255 – горизонтально).
	MU	BOOL	Сигнал для ИМ (открытие).
	MD	BOOL	Сигнал для ИМ (закрытие).
	STATUS	BYTE	ESR-код.
<b>Параметры</b>	SENS	BYTE	Зона нечувствительности.
	T_LOCKOUT	TIME	Задержка при переключении направления.
Используемые модули	<a href="#">BLIND_ACTUATOR</a>		

Рис. 7.3. Внешний вид ФБ **BLIND\_CONTROL** на языке CFC

Функциональный блок **BLIND\_CONTROL** представляет собой интерфейсный модуль для исполнительного механизма, управляющего жалюзи или ставнями, с поддержкой автоматического режима работы. Пока вход **UP** имеет значение **TRUE**, происходит открытие жалюзи: сигнал управления ИМ **MU** принимает значение **TRUE**, значение выхода **ANG** (угол наклона ламелей) начинает увеличиваться со скоростью  $(255/T\_ANGLE)$  ед./с; после того, как **ANG** достигнет **255**, значение выхода **POS** (положение жалюзи) начинается увеличиваться со скоростью  $(255/T\_UD)$  ед./с. Аналогичным образом вход **DN** используется для закрытия жалюзи.

Выходы **POS** и **ANG** не отражают реальное состояние жалюзи, и вычисляются на основе значений входов **T\_UD** и **T\_ANGLE**, которые должны быть определены экспериментально для конкретной модели жалюзи.

Параметр **T\_LOCKOUT** определяет время блокировки блока при изменении направления движения жалюзи. Например, если **T\_LOCKOUT = T#5s**, то после остановки открытия жалюзи сигнал на закрытие жалюзи начнет обрабатываться после 5-секундной задержки.

Если входы **UP** и **DN** имеют значение **TRUE** одновременно, то блок переходит в режим автоматического управления. В этом режиме значения выходов **POS** и **ANG** приводятся в соответствие со значениями уставок **PI** и **AI** с учетом зоны нечувствительности **SENS**. Зона нечувствительности должна выбираться в зависимости от возможной точности позиционирования жалюзи с помощью конкретного ИМ. Выход **MU** имеет значение **TRUE** во время открытия жалюзи, выход **MD** имеет значение **TRUE** во время закрытия.

Если до перехода в автоматический режим **POS = ANG = 0** или **POS = ANG = 255**, то сначала ламели переводятся в горизонтальное (**ANG = 255**) или вертикальное (**ANG = 0**) положение, затем значение **POS** увеличивается/уменьшается до значения **PI**, после чего значения **ANG** уменьшается/увеличивается до значения **AI**.

Ниже описана зависимость основных входов и выходов блока:

UP	DN	PI	AI	MU	MD	Описание
FALSE	FALSE	-	-	FALSE	FALSE	Режим ожидания
TRUE	FALSE	-	-	TRUE	FALSE	Ручное управление (открытие)
FALSE	TRUE	-	-	FALSE	TRUE	Ручное управление (закрытие)
TRUE	TRUE	P	A	TRUE или FALSE	TRUE или FALSE	Автоматическое открытие или закрытие до значений P и A

Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**. Вход **S\_IN** содержит ESR-код, полученный от другого модуля. Он транслируется на выход **STATUS** в тех случаях, когда блок не имеет собственных сообщений.

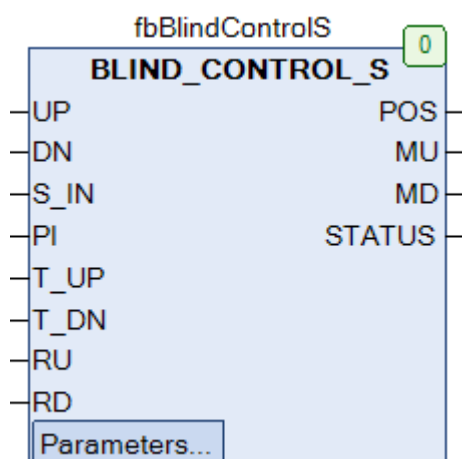
STATUS	Описание
0	Нет сообщений
101	Жалюзи открываются в режиме ручного управления
102	Жалюзи закрываются в режиме ручного управления
121	Жалюзи открываются в режиме автоматического управления
122	Жалюзи закрываются в режиме автоматического управления
123	Открытие ламелей в режиме автоматического управления
124	Закрытие ламелей в режиме автоматического управления
S_IN	Передача сообщения от другого блока

**Примечание:** из-за ошибки в коде ФБ для корректной работы выхода **STATUS** требуется удалить последнюю строку кода блока ([пруф](#)):

```
status := act.status; // нужно удалить эту строку
```

## 7.4. BLIND\_CONTROL\_S

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	UP	BOOL	Команда ручного управления (открытие).
	DN	BOOL	Команда ручного управления (закрытие).
	S_IN	BYTE	ESR-код предыдущего модуля.
	PI	BYTE	Требуемая степень открытия штор (0 – полностью закрыто, 255 – полностью открыто).
	T_UD	BYTE	Время полного открытия штор.
	T_DN	BYTE	Время полного закрытия штор.
	RU	BOOL	TRUE – учитывать макс. уровень открытия штор.
	RD	BOOL	TRUE – учитывать мин. уровень открытия штор.
<b>Выходы</b>	POS	BYTE	Положение штор (0 – полностью закрыто, 255 – полностью открыто).
	MU	BOOL	Сигнал для ИМ (открытие).
	MD	BOOL	Сигнал для ИМ (закрытие).
	STATUS	BYTE	ESR-код.
<b>Параметры</b>	T_LOCKOUT	TIME	Задержка при переключении направления.
	T_EXT	TIME	Дополнительное время хода.
	EXT_TRIG	BYTE	Степень открытия для выделения доп. времени.
	R_POS_TOP	BYTE	Максимальный уровень открытия штор.
	R_POS_BOT	BYTE	Минимальный уровень открытия штор.
Используемые модули	OSCAT_BASIC.T_PLC_MS, OSCAT_BASIC._RMP_NEXT		

Рис. 7.4. Внешний вид ФБ **BLIND\_CONTROL\_S** на языке CFC

Функциональный блок **BLIND\_CONTROL\_S** представляет собой интерфейсный модуль для исполнительного механизма, управляющего шторами. Пока вход **UP** имеет значение **TRUE**, происходит открытие штор: сигнал управления ИМ **MU** принимает значение **TRUE**, значение

выхода **POS** (положение штор) начинает увеличиваться со скоростью  $(255/T\_UP)$  ед./с. Аналогичным образом вход **DN** используется для закрытия штор (**MD=TRUE**) со скоростью  $(255/T\_DN)$  ед./с.

Выход **POS** отражают реальное положение штор, и вычисляются на основе значений входов **T\_UD** и **T\_DN**, которые должны быть определены экспериментально для конкретной модели штор.

Параметр **T\_LOCKOUT** определяет время блокировки блока при изменении направления движения штор. Например, если **T\_LOCKOUT = T#5s**, то после остановки открытия сигнал на закрытие штор начнет обрабатываться после 5-секундной задержки.

Если входы **UP** и **DN** имеют значение **TRUE** одновременно, то блок переходит в режим автоматического управления. В этом режиме значение выхода **POS** приводится в соответствие со значением уставки **PI**. Выход **MU** имеет значение **TRUE** во время открытия штор, выход **MD** имеет значение **TRUE** во время закрытия. Если входы **RU/RD** имеют значение **TRUE**, то в автоматическом режиме уровень открытия штор ограничивается значениями **R\_POS\_TOP/ R\_POS\_BOT** (максимальная/минимальная степень открытия).

Если при первом запуске блока он уже находится в автоматическом режиме (например, после перезагрузки ПЛК), то выполняется процедура калибровки. В режиме калибровки шторы полностью открываются, а затем устанавливаются в позицию **PI**.

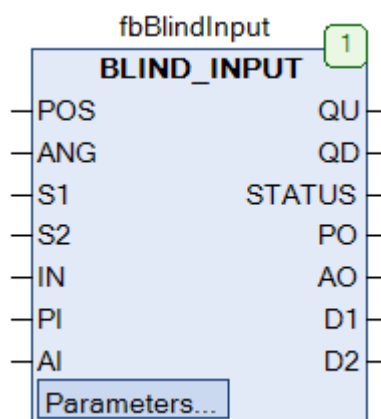
Так как разные ИМ могут иметь разное время хода, то для подстройки могут использоваться параметры **T\_EXT** (дополнительное время хода) и **EXT\_TRIG**. Параметр **EX\_TRIG** определяет нижнее/верхнее положение штор ( $0+EXT\_TRIG / 255-EXT\_TRIG$ ), начиная с которого выделяется дополнительное время на закрытие/открытие штор.

Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**. Вход **S\_IN** содержит ESR-код, полученный от другого модуля. Он транслируется на выход **STATUS** в тех случаях, когда блок не имеет собственных сообщений. Ниже описана зависимость основных входов и выходов блока:

UP	DN	MU	MD	STATUS	Описание
FALSE	FALSE	FALSE	FALSE	125/S_IN	Режим ожидания
TRUE	FALSE	TRUE	FALSE	121	Ручное управление (открытие)
FALSE	TRUE	FALSE	TRUE	122	Ручное управление (закрытие)
TRUE	TRUE	TRUE или FALSE	TRUE или FALSE	123	Режим автоматического управление
-	-	FALSE	FALSE	124	Достигнут предел (R_POS_TOP или R_POS_BOT)
-	-	FALSE	FALSE	127	Блокировка управления (T_LOCKOUT)
TRUE	TRUE	TRUE	FALSE	128	Режим калибровки
TRUE или FALSE	TRUE или FALSE	TRUE или FALSE	TRUE или FALSE	129	Выделено дополнительное время (T_EXT)

## 7.5. BLIND\_INPUT

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	POS	BYTE	Положение жалюзи (0 – полностью закрыто, 255 – полностью открыто).
	ANG	BYTE	Текущий угол открытия ламелей.
	S1	BOOL	Команда ручного управления (открытие).
	S2	BOOL	Команда ручного управления (закрытие).
	IN	BOOL	TRUE – автоматический режим управления. FALSE – ручной.
	PI	BYTE	Требуемая степень открытия жалюзи (0 – полностью закрыто, 255 – полностью открыто).
	AI	BYTE	Требуемый угол наклона ламелей (0 – вертикально, 255 – горизонтально).
<b>Выходы</b>	QU	BOOL	Сигнал для ИМ (открытие).
	QD	BOOL	Сигнал для ИМ (закрытие).
	STATUS	BYTE	ESR-код.
	PO	BYTE	Уставка положения жалюзи (0 – полностью закрыто, 255 – полностью открыто).
	AO	BYTE	Уставка угла наклона ламелей (0 – вертикально, 255 – горизонтально).
	D1	BOOL	Двойное нажатие на S1/S2 (в зависимости от настроек либо единичный импульс, либо инверсия).
	D2	BOOL	
<b>Параметры</b>	SINGLE_SWITCH	BOOL	Управление выходами QU/QD кнопкой S1.
	CLICK_EN	BOOL	Обработка коротких нажатий.
	CLICK_TIME	TIME	Макс. время короткого нажатия.
	MAX_RUNTIME	TIME	Макс. время хода жалюзи.
	MANUAL_TIMEOUT	TIME	Таймаут ожидания ручного управления.
	DEBOUNCE_TIME	TIME	Минимальное время нажатия кнопки.
	DBL_CLK1	BOOL	Запись PO/AO по двойному нажатию S1.
	DBL_POS1	BYTE	Уставка PO для S1.
	DBL_ANG1	BYTE	Уставка АО для S1.
	DBL_CLK2	BOOL	Запись PO/AO по двойному нажатию S2.
	DBL_POS2	BYTE	Уставка PO для S2.
	DBL_ANG2	BYTE	Уставка АО для S2.
	D1_TOGGLE	BOOL	Режим обработки двойного нажатия S1.
	D2_TOGGLE	BOOL	Режим обработки двойного нажатия S2.
MASTER_MODE	BOOL	Заморозка выходов PO/AO в режиме ожидания.	
Используемые модули	<a href="#">CLICK_MODE</a> , OSCAT_BASIC.T_PLC_MS		

Рис. 7.4. Внешний вид ФБ **BLIND\_INPUT** на языке CFC

Функциональный блок **BLIND\_INPUT** представляет собой интерфейсный модуль для обработки нажатий кнопок при управлении жалюзи. Блок может работать в ручном или автоматическом режиме. К входам **S1** и **S2** подключаются сигналы кнопок управления. Параметр **T\_DEBOUNCE** определяет минимально фиксируемое время нажатия для игнорирования дребезга контактов.

Если вход **IN** имеет значение **FALSE**, то блок работает в ручном режиме. В этом случае значение входов **S1/S2** транслируются на выходы **QU/QD** (открытие/закрытие жалюзи). Если параметр **SINGLE\_SWITCH** имеет значение **TRUE**, то значение кнопки **S2** не обрабатывается, и управление обоими выходами осуществляется с помощью импульсов по переднему фронту на входе **S1**: по нечетным импульсам выход **QU** принимает значение **TRUE**, а **QD** – **FALSE**, по четным – выход **QU** принимает значение **FALSE**, а **QD** – **TRUE**.

Если параметр **CLICK\_EN** имеет значение **TRUE**, то первое короткое (не длиннее времени **CLICK\_TIME**) нажатие на кнопку **S1/S2** активирует выход **QU/QD**, а второе – деактивирует. Если же время нажатия превышает **CLICK\_TIME**, то выход будет деактивирован сразу после отпускания кнопки. Параметр **MAX\_RUNTIME** определяет максимальное время хода жалюзи. По истечению этого времени выход **QU/QD** автоматически отключается и блок переходит в режим ожидания ручного или автоматического управления.

Выходы **D1/D2** используются, чтобы детектировать двойные нажатия на кнопки **S1/S2** (двойное нажатие должно бы осуществлено в течение времени **CLICK\_TIME**). Если параметр **Dx\_TOGGLE** имеет значение **FALSE**, то по двойному нажатию кнопки **Sx** на выходе **Dx** генерируется единичный импульс. Если параметр **Dx\_TOGGLE** имеет значение **TRUE**, то по двойному нажатию кнопки **Sx** выход **Dx** инвертирует свое значение. Если параметр **DBL\_CLKx** имеет значение **TRUE**, то при двойном нажатии кнопки **Sx** выходам **PO/AO** присваиваются значения **DBL\_POSx/ DBL\_ANGx**.

Параметр **MANUAL\_TIMEOUT** определяет время неактивности кнопок **S1** и **S2**, спустя которое осуществляется переход из режима ожидания ручного управления (**STATUS = 131**) в режим ожидания автоматического управления (**STATUS = 130**). Если обе кнопки зажаты в течение времени **CLICK\_TIME**, то блок принудительно переходит в режим ожидания ручного управления (**STATUS = 139**). В этом случае значения параметра **MANUAL\_TIMEOUT** не учитывается, и переход в режим ожидания автоматического управления не будет осуществлен.

Автоматический режим может быть активирован присвоением значения **TRUE** входу **IN**. В этом режиме оба выхода **QU/QD** имеют значение **TRUE** (чтобы следующий блок в схеме работал в автоматическом режиме), а значения входов **PI/AI** транслируются на выходы **PO/AO**. На входы **POS** и **ANG** должны быть подключены сигналы текущего положения жалюзи и угла наклона ламелей, рассчитанные с помощью ФБ [BLIND\\_CONTROL](#).

Если параметр **MASTER\_MODE** имеет значение **TRUE**, то в режиме ожидания автоматического управления (**STATUS = 130**) выходы **PO/AO** сохраняют свои последние значения. В противном случае (**MASTER\_MODE = FALSE**) на эти выходы передаются значения входов **POS/ANG**. Параметр должен быть выставлен в **TRUE** при совместной работе блока с ФБ [BLIND\\_SHADE](#), [BLIND\\_NIGHT](#) и т.д.

Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**.

STATUS	QU	QD	PO/AO	Описание
130	TRUE	TRUE	POS/ANG <sup>4</sup>	Режим ожидания (автоматическое управление)
131	FALSE	FALSE	POS/ANG	Режим ожидания (ручное управление)
132	TRUE	FALSE	POS/ANG	Ручное управление (открытие)
133	FALSE	TRUE	POS/ANG	Ручное управление (закрытие)
134	TRUE	FALSE	POS/ANG	Зафиксировано одиночное нажатие на входе S1
135	FALSE	TRUE	POS/ANG	Зафиксировано одиночное нажатие на входе S2
136	TRUE	TRUE	PI/AI	Автоматический режим управление (IN=TRUE)
137	TRUE	TRUE	DBL_POS1/DBL_ANG1	Зафиксировано двойное нажатие на входе S1
138	TRUE	TRUE	DBL_POS2/DBL_ANG2	Зафиксировано двойное нажатие на входе S2
139	FALSE	FALSE	POS/ANG	Режим ожидания (принудительно)

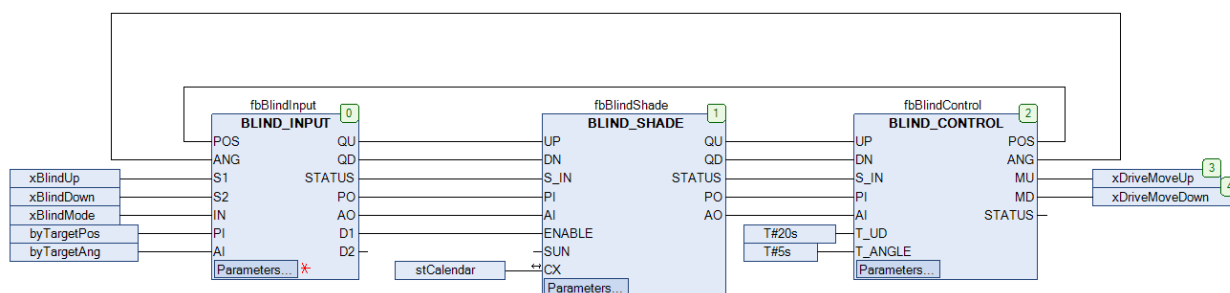
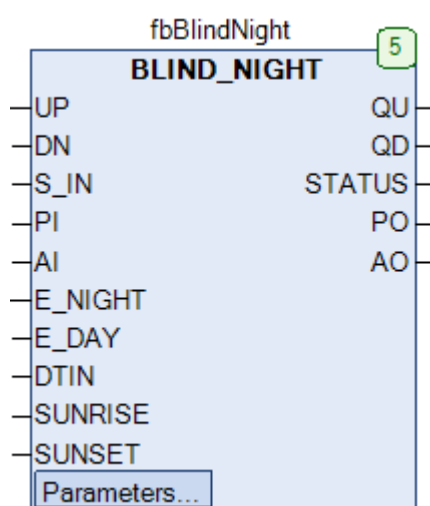


Рис. 7.5. Пример работы с ФБ **BLIND\_INPUT** на языке CFC

<sup>4</sup> Если **MASTER\_MODE = TRUE**, то выходы **PO/AO** сохраняют свои последние значения.

## 7.6. BLIND\_NIGHT

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	UP	BOOL	Команда ручного управления (открытие).
	DN	BOOL	Команда ручного управления (закрытие).
	S_IN	BYTE	ESR-код предыдущего модуля.
	PI	BYTE	Требуемая степень открытия жалюзи (0 – полностью закрыто, 255 – полностью открыто).
	AI	BYTE	Требуемый угол наклона ламелей (0 – вертикально, 255 – горизонтально).
	E_NIGHT	BOOL	TRUE – включить обработку наступления ночи.
	E_DAY	BOOL	TRUE – включить обработку наступления дня.
	DTIN	DT	Текущие дата и время.
	SUNRISE	TOD	Время восхода.
	SUNSET	TOD	Время заката.
<b>Выходы</b>	QU	BOOL	Сигнал для ИМ (открытие).
	QD	BOOL	Сигнал для ИМ (закрытие).
	STATUS	BYTE	ESR-код.
	PO	BYTE	Уставка положения жалюзи (0 – полностью закрыто, 255 – полностью открыто).
	AO	BYTE	Уставка угла наклона ламелей (0 – вертикально, 255 – горизонтально).
<b>Параметры</b>	SUNRISE_OFFSET	TIME	Задержка активации относительно рассвета.
	SUNSET_OFFSET	TIME	Задержка активации относительно заката.
	NIGHT_POSITION	BYTE	Уставка положения жалюзи для ночи.
	NIGHT_ANGLE	BYTE	Уставка угла наклона ламелей для ночи.

Рис. 7.6. Внешний вид ФБ **BLIND\_NIGHT** на языке CFC



Функциональный блок **BLIND\_NIGHT** представляет собой интерфейсный модуль для автоматического закрытия жалюзи и штор на ночь. На вход **DTIN** подается текущее системное время, на входы **SUNRISE** и **SUNSET** – время восхода и заката солнца для заданной широты и долготы (можно использовать выходы блока **SUN\_TIME** из библиотеки **OSCAT\_BASIC**). Параметры **SUNRISE\_OFFSET** и **SUNSET\_OFFSET** определяют задержку активации ФБ относительно времени наступления рассвета и заката соответственно.

Входы **UP** и **DN** используются для ручного открытия и закрытия жалюзи/штор. Значения входов передаются на выходы **QU** и **QD**. К моменту наступления ночи оба этих входа должны иметь значения **TRUE** для перевода блока в автоматический режим управления. Если в течение ночи хотя бы один из входов примет значение **FALSE**, то это будет восприниматься как переход в режим ручного управления.

Входы **E\_NIGHT** и **E\_DAY** включают обработку наступление ночи и дня.

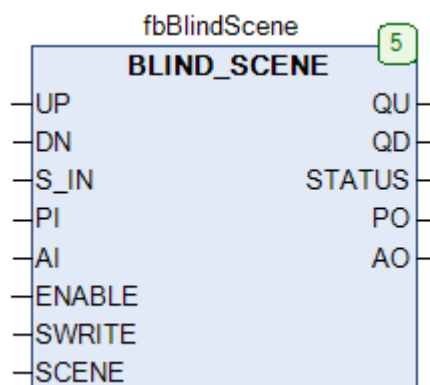
Если **E\_NIGHT = TRUE**, то при наступлении ночи (текущее время > **SUNSET + SUNSET\_OFFSET**) на выходы **PO** и **AO** передаются значения параметров **NIGHT\_POSITION** и **NIGHT\_ANGLE**. Это позволяет автоматически опускать жалюзи/шторы в ночное время и закрывать ламели.

Если **E\_DAY = TRUE**, то при наступлении дня (текущее время > **SUNRISE + SUNRISE\_OFFSET**) на выходы **PO** и **AO** передаются значения входов **PI** и **AI**. Тоже самое происходит в режиме ручного управления (независимо от значения **E\_DAY**).

Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**. Ночью **STATUS = 141**, в течение дня и в режиме ручного управления **STATUS = S\_IN**.

## 7.7. BLIND\_SCENE

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	UP	BOOL	Команда ручного управления (открытие).
	DN	BOOL	Команда ручного управления (закрытие).
	S_IN	BYTE	ESR-код предыдущего модуля.
	PI	BYTE	Требуемая степень открытия жалюзи (0 – полностью закрыто, 255 – полностью открыто).
	AI	BYTE	Требуемый угол наклона ламелей (0 – вертикально, 255 – горизонтально).
	ENABLE	BOOL	Разрешение на активацию сценария.
	SWRITE	BOOL	Передний фронт – сохранить текущие значения как сценарий с номером SCENE.
	SCENE	BYTE	Номер выбранного сценария (0...15).
<b>Выходы</b>	QU	BOOL	Сигнал для ИМ (открытие).
	QD	BOOL	Сигнал для ИМ (закрытие).
	STATUS	BYTE	ESR-код.
	PO	BYTE	Уставка положения жалюзи (0 – полностью закрыто, 255 – полностью открыто).
	AO	BYTE	Уставка угла наклона ламелей (0 – вертикально, 255 – горизонтально).

Рис. 7.7. Внешний вид ФБ **BLIND\_SCENE** на языке CFC

Функциональный блок **BLIND\_SCENE** представляет собой интерфейсный модуль для выбора сценария освещения. Блок может хранить до 16-ти сценариев. Для управления сценариями входы **UP** и **DN** одновременно должны иметь значение **TRUE**. Вход **SCENE** определяет номер сценария (**0...15**). Если вход **ENABLE** имеет значение **TRUE**, то по переднему фронту на входе **SWRITE** значения входов **PI** и **AI** сохраняются в виде сценария с номером **SCENE**. При выборе сценария (**ENABLE = TRUE, SCENE = x**) на выходы **PO** и **AO** будут переданы соответствующие сохраненные значения **PI** и **AI**. Выходы **QU** и **QD** при этом будут активны, чтобы следующий блок схемы работал в автоматическом режиме.

Для блокировки сценария следует при нужном значении **SCENE** и деактивированном входе **ENABLE** подать импульс по переднему фронту на входе **SWRITE**. В этом случае при выборе данного

сценария выходы блока будут определяться значение входов **UP**, **DN**, **PI** и **AI**. Также аналогичное поведение выходов будет проявляться для любого сценария при **ENABLE = FALSE** или в том случае, если любой из входов **UP/DN** имеет значение **FALSE**.

Ниже описана зависимость основных входов и выходов блока:

UP	DN	ENABLE	SWRITE	SCENE	QU	QD	PO	AO	Описание
TRUE	TRUE	FALSE	FALSE	-	TRUE	TRUE	PI	AI	Ручное управление
-	-	TRUE	TRUE	x	-	-	PI[x]	AI[x]	Сохранение сценария
-	-	FALSE	TRUE	x	-	-	PI	AI	Блокировка сценария
TRUE	TRUE	TRUE	FALSE	x	TRUE	TRUE	PI[x]	AI[x]	Выбор сценария

Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**. Вход **S\_IN** содержит ESR-код, полученный от другого модуля. Он транслируется на выход **STATUS** в тех случаях, когда блок не имеет собственных сообщений.

STATUS	Описание
160...175	Выбран режим 1...16
176	Сохранение сценария
S_IN	Передача сообщения от другого блока

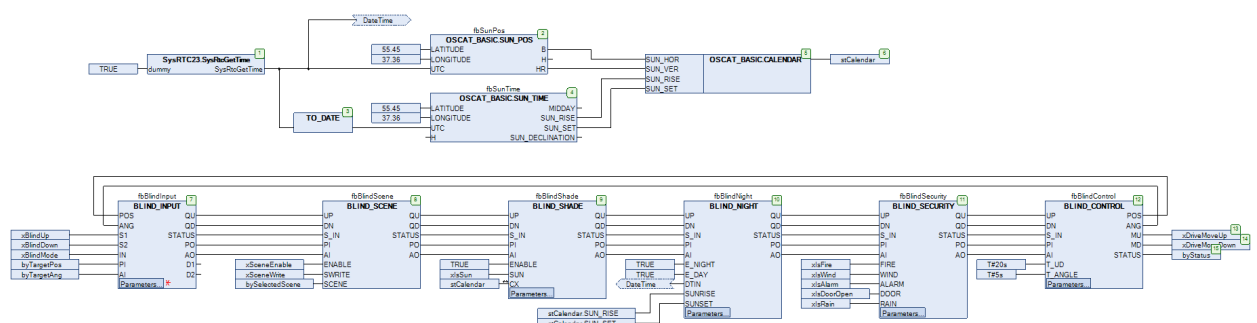
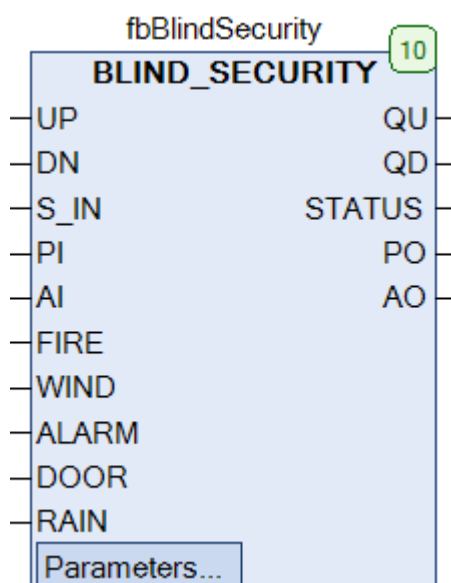


Рис. 7.8. Пример использования ФБ **BLIND\_SCENE** совместно с другими модулями библиотеки на языке CFC (рисунок хорошо масштабируется)

## 7.8. BLIND\_SECURITY

Тип модуля: ФБ	Переменная	Тип	Описание
Входы	UP	BOOL	Команда ручного управления (открытие).
	DN	BOOL	Команда ручного управления (закрытие).
	S_IN	BYTE	ESR-код предыдущего модуля.
	PI	BYTE	Требуемая степень открытия жалюзи (0 – полностью закрыто, 255 – полностью открыто).
	AI	BYTE	Требуемый угол наклона ламелей (0 – вертикально, 255 – горизонтально).
	FIRE	BOOL	TRUE – пожар.
	WIND	BOOL	TRUE – сильный ветер.
	ALARM	BOOL	TRUE – обнаружено проникновение.
	DOOR	BOOL	TRUE – открыта входная дверь.
	RAIN	BOOL	TRUE – идет дождь.
Выходы	QU	BOOL	Сигнал для ИМ (открытие).
	QD	BOOL	Сигнал для ИМ (закрытие).
	STATUS	BYTE	ESR-код.
	PO	BYTE	Уставка положения жалюзи (0 – полностью закрыто, 255 – полностью открыто).
	AO	BYTE	Уставка угла наклона ламелей (0 – вертикально, 255 – горизонтально).
Параметры	ALARM_UP	BOOL	Реакция на проникновение (TRUE – открытие).
	WIND_UP	BOOL	Реакция на ветер (TRUE – открытие).
	RAIN_UP	BOOL	Реакция на дождь (TRUE – открытие).

Рис. 7.9. Внешний вид ФБ **BLIND\_SECURITY** на языке CFC

Функциональный блок **BLIND\_SECURITY** представляет собой интерфейсный модуль для автоматического открытия/закрытия жалюзи при наступлении определенных событий. На входы

**FIRE/WIND/ALARM/DOOR/RAIN** заводятся дискретные сигналы от датчиков пожара/ветра/проникновения/открытия входной двери/дождя (приоритет обработки датчиков соответствует перечисленной последовательности). При детектировании сигнала датчика происходит воздействие на выходы **QU/QD** (открытие или закрытие жалюзи). Параметры **ALARM\_UP/WIND\_UP/RAIN\_UP** позволяют определить действие, которое произойдет при сигнале от датчика проникновения/ветра/дождя: **TRUE** – открытие жалюзи, **FALSE** – закрытие. При наличии сигнала от датчиков пожара или открытия входной двери всегда будет выполняться открытие жалюзи.

При отсутствии сигналов от датчиков блок работает в режиме ручного управления: значения входов **UP/DN/S\_IN/PI/AI** передаются на выходы **QU/QD/STATUS/PO/AO**.

Выходы блока должны подключаться напрямую к входам блока [BLIND CONTROL](#).

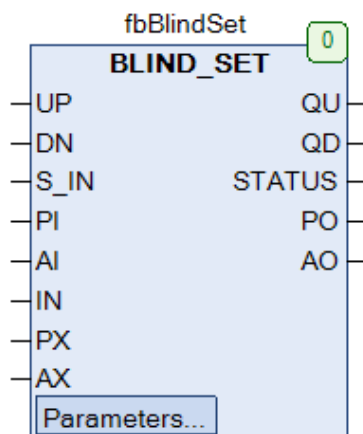
Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**. Вход **S\_IN** содержит ESR-код, полученный от другого модуля. Он транслируется на выход **STATUS** в тех случаях, когда блок не имеет собственных сообщений.

<b>STATUS</b>	<b>Описание</b>
111	Сигнал от датчика пожара
112	Сигнал от датчика ветра
113	Сигнал от датчика проникновения
114	Сигнал от датчика открытия входной двери
115	Сигнал от датчика дождя
S_IN	Передача сообщения от другого блока

Пример работы с блоком приведен на рис. 7.8.

## 7.9. BLIND\_SET

Тип модуля: ФБ	Переменная	Тип	Описание
<b>Входы</b>	UP	BOOL	Команда ручного управления (открытие).
	DN	BOOL	Команда ручного управления (закрытие).
	S_IN	BYTE	ESR-код предыдущего модуля.
	PI	BYTE	Требуемая степень открытия жалюзи в автоматическом режиме (0 – полностью закрыто, 255 – полностью открыто)
	AI	BYTE	Требуемый угол наклона ламелей в автоматическом режиме (0 – вертикально, 255 – горизонтально).
	IN	BOOL	TRUE – ручной режим (может быть заблокирован состоянием UP, DN).
	PX	BYTE	Требуемая степень открытия жалюзи в ручном режиме (0 – полностью закрыто, 255 – полностью открыто).
	AX	BYTE	Требуемый угол наклона ламелей в ручном режиме (0 – вертикально, 255 – горизонтально).
<b>Выходы</b>	QU	BOOL	Сигнал для ИМ (открытие).
	QD	BOOL	Сигнал для ИМ (закрытие).
	STATUS	BYTE	ESR-код.
	PO	BYTE	Уставка положения жалюзи (0 – полностью закрыто, 255 – полностью открыто).
	AO	BYTE	Уставка угла наклона ламелей (0 – вертикально, 255 – горизонтально).
<b>Параметры</b>	OVERRIDE_MANUAL	BOOL	TRUE – состояние UP, DN не может заблокировать переход в ручной режим).
	RESTORE_POSITION	BOOL	TRUE – при выходе из ручного режима восстановить предыдущее состояние на время RESTORE_TIME.
	RESTORE_TIME	BOOL	Время, на которое будет восстановлено предыдущее состояние при выходе из ручного режима управления.
<b>Используемые модули</b>	OSCAT_BASIC.T_PLC_MS		

Рис. 7.10. Внешний вид ФБ **BLIND\_SET** на языке CFC

Функциональный блок **BLIND\_SET** представляет собой интерфейсный модуль для ручного управления жалюзи. Если вход **IN** имеет значение **FALSE**, то значения входов **UP/DN/S\_IN/PI/AI** передаются на выходы **QU/QD/STATUS/PO/AO**. Если входы **UP** и **DN** имеют значение **TRUE** (т. е. блок работает в режиме автоматического управления), и при этом вход **IN** принимает значение **TRUE**, то блок переходит в режим ручного управления. Если параметр **OVERRIDE\_MANUAL** имеет значение **TRUE**, то переход в режим ручного управления происходит независимо от значения входов **UP/DN**. В режиме ручного управления входы **QU** и **QD** имеют значение **TRUE**, а на выходы **PO/AO** передаются значения входов **PX/AX**. Если параметр **RESTORE\_POSITION** имеет значение **TRUE**, то после выхода из режима ручного управления (при **IN = FALSE**) выходы **PO/AO** на время **RESTORE\_TIME** принимают значения, которые имели до перехода на ручное управление. По истечению времени **RESTORE\_TIME** на выходы **PO/AO** начинаются передаваться значения входов **PI/AI**.

Выход **STATUS** определяет состояние блока и совместим с **ESR-модулями** из библиотеки **OSCAT Basic**. Вход **S\_IN** содержит ESR-код, полученный от другого модуля. Он транслируется на выход **STATUS** в тех случаях, когда блок не имеет собственных сообщений.

<b>STATUS</b>	<b>Описание</b>
178	Режим ручного управления (IN = TRUE)
179	Выход из режима ручного управления, возврат к прежним значениям выходов на время RESTORE_TIME
S_IN	Передача сообщения от другого блока

## 7.10. BLIND\_SHADE

Тип модуля: ФБ	Переменная	Тип	Описание
Входы	UP	BOOL	Команда ручного управления (открытие).
	DN	BOOL	Команда ручного управления (закрытие).
	S_IN	BYTE	ESR-код предыдущего модуля.
	PI	BYTE	Требуемая степень открытия жалюзи (0 – полностью закрыто, 255 – полностью открыто).
	AI	BYTE	Требуемый угол наклона ламелей (0 – вертикально, 255 – горизонтально).
	ENABLE	BOOL	TRUE – разрешить работу блока.
	SUN	BOOL	TRUE – сейчас ясно.
Выходы	QU	BOOL	Сигнал для ИМ (открытие).
	QD	BOOL	Сигнал для ИМ (закрытие).
	STATUS	BYTE	ESR-код.
	PO	BYTE	Уставка положения жалюзи (0 – полностью закрыто, 255 – полностью открыто).
	AO	BYTE	Уставка угла наклона ламелей (0 – вертикально, 255 – горизонтально).
Входы-выходы	CX	<sup>1</sup>	Данные календаря.
Параметры	sunrise_offset	TIME	Задержка активации относительно рассвета.
	sunset_preset	TIME	Смещение для преждевременной активации относительно заката.
	direction	REAL	Направление, на которое выходит окно (180° – юг).
	angle_offset	REAL	Смещение направления относительно юга.
	slat_width	REAL	Ширина ламелей, мм.
	Slat_spacing	REAL	Расстояние между ламелями, мм.
	Shade_delay	TIME	Задержка активации.
	Shade_pos	BYTE	Положение жалюзи при затемнении.
Используемые модули	<sup>1</sup> OSCAT_BASIC.CALENDAR, OSCAT_BASIC.DEG, OSCAT_BASIC.RAD		

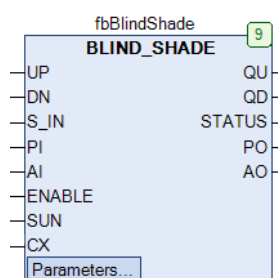


Рис. 7.11. Внешний вид ФБ BLIND\_SHADE на языке CFC

**Примечание переводчика:** в текущей версии библиотеки ФБ работает **некорректно** (**пруф**). Для корректной работы необходимо отредактировать код ФБ следующим образом:

```

4 IF UP AND DN AND enable AND sun_delay.Q AND cx.SUN_HOR > direction - angle_offset AND cx.SUN_HOR < direction + angle_offset AND
5 DT_TO_TOD(cx.LDT) > cx.SUN_RISE + sunrise_offset AND DT_TO_TOD(cx.LDT) < cx.SUN_SET - sunset_preset THEN
6

```

Рис. 7.12. Исправление исходного кода ФБ BLIND\_SHADE для обеспечения корректной работы



Функциональный блок **BLIND\_SHADE** представляет собой интерфейсный модуль для автоматического затемнения помещения с помощью жалюзи для защиты от ярких солнечных лучей. На вход-выход **CX** подается экземпляр структуры **OSCAT\_BASIC.CALENDAR**, который может быть получен с выхода ФБ **CALENDAR\_CALC** из библиотеки **OSCAT\_BASIC**. Структура содержит информацию о текущем системном времени, времени рассвета, заката и др.

Блок может работать в режиме затемнения или режиме ручного управления. В режиме затемнения жалюзи закрываются в то время суток, когда солнечный свет наиболее ярок и может создавать неудобства.

Режим затемнения активируется при одновременном выполнении следующих условий:

- предыдущий ФБ системы управления работает в автоматическом режиме: входы **UP** и **DN** ФБ **BLIND\_SHADE** имеют значение **TRUE**;
- получено разрешение на работу ФБ **BLIND\_SHADE**: вход **ENABLE** имеет значение **TRUE**;
- солнечный свет падает в окно: азимут солнца находится в пределах (**direction** – **angle\_offset**...**direction** + **angle\_offset**), где **direction** – направление, на которое выходит окно (0 – север, 90 – восток, 180 – юг, 270 – запад), а **angle\_offset** – смещение для этого направления;
- солнце уже взошло и еще не зашло: текущее системное время находится в пределах (время рассвета + **sunrise\_offset**...время заката – **sunset\_preset**). Время рассвета и время заката считываются с входа-выхода **CX**;
- есть сигнал от датчика солнца: вход **SUN** имеет значение **TRUE**. Если вход **SUN** принимает значение **FALSE**, то блок продолжает работать еще в течение времени **Shade\_delay** – это позволяет избежать ложных отключений блока из-за проплывающих облаков.

Ниже приведено графическое представление параметров **direction** и **angle\_offset**. Параметр **direction** определяет направление, на которое выходит окно (0 – север, 90 – восток, 180 – юг, 270 – запад) – это позволяет закрывать жалюзи только в то время, когда солнце светит в данном направлении. Параметр **angle\_offset** представляет собой смещение для этого направления, которое позволяет учесть толщину стены и ширину окна, чтобы избежать ненужных закрытий. На рис. 7.12 **direction** = 135°, **angle\_offset** = 65°.

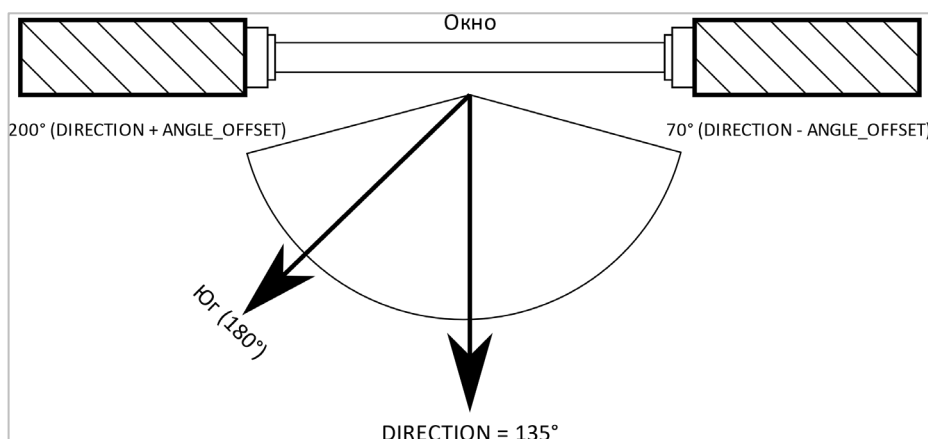


Рис. 7.13. Графическое представление параметров **direction** и **angle\_offset**

В режиме затемнения выходы блока имеют следующие значения:

- выходы **QU** и **QD** имеют значение **TRUE**, чтобы следующий блок системы работал в автоматическом режиме;
- на выход **PO** передается значение параметра **Shade\_pos** (установка положения жалюзи при затемнении);
- значение выхода **AO** рассчитывается таким образом, чтобы обеспечить оптимальный угол наклона ламелей ширины **slat\_width** и интервалом между планками **Slat\_spacing**;
- Выход **STATUS** имеет значение **151**.

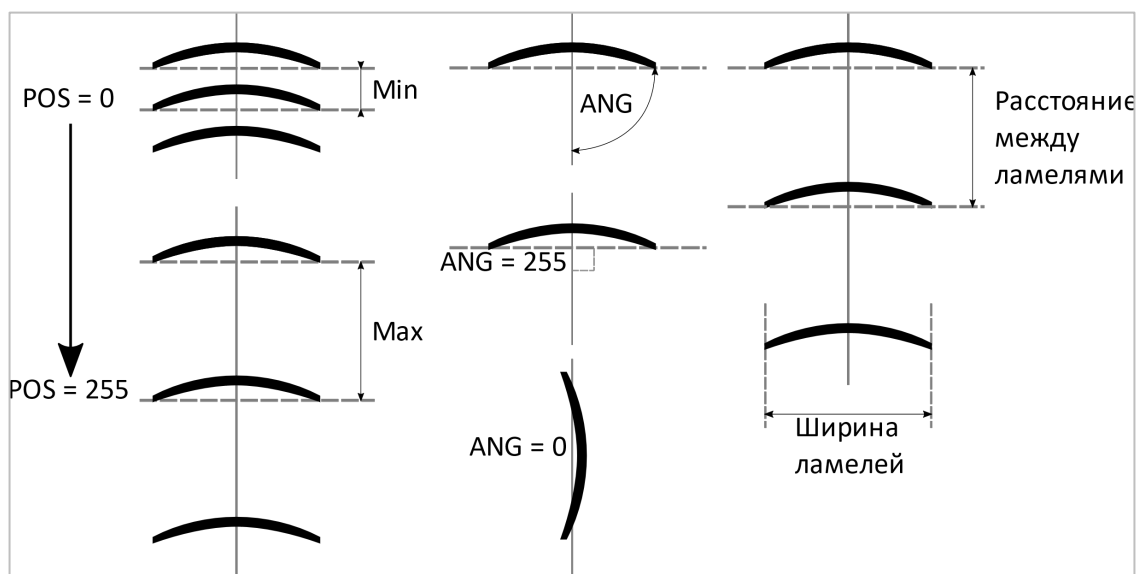


Рис. 7.14. Схема управления жалюзи (I)

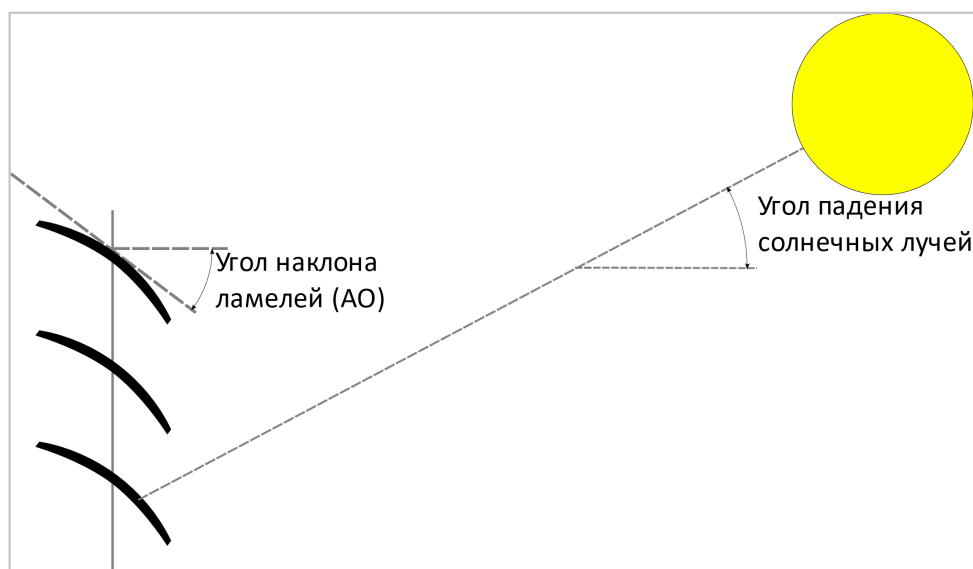
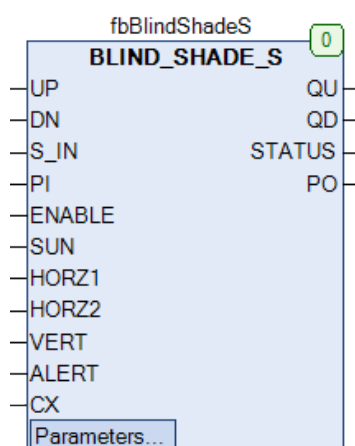


Рис. 7.15. Схема управления жалюзи (II)

В режиме ручного управления на выходы **QU/QD/PO/AO/STATUS** передаются значения входов **UP/DN/PI/AI/S\_IN**. Пример использования ФБ приведен на рис. 7.5 и 7.8.

## 7.11. BLIND\_SHADE\_S

Тип модуля: ФБ	Переменная	Тип	Описание
Входы	UP	BOOL	Команда ручного управления (открытие).
	DN	BOOL	Команда ручного управления (закрытие).
	S_IN	BYTE	ESR-код предыдущего модуля.
	PI	BYTE	Требуемая степень открытия штор (0 – полностью закрыто, 255 – полностью открыто).
	ENABLE	BOOL	TRUE – разрешить работу блока.
	SUN	BOOL	TRUE – сейчас ясно.
	HORZ1	REAL	Мин. азимут солнца для режима затемнения.
	HORZ2	REAL	Макс. азимут солнца для режима затемнения.
	VERT	REAL	Макс. высота солнца над горизонтом для режима затемнения.
	ALERT	BOOL	TRUE – принудительное открытие жалюзи.
Выходы	QU	BOOL	Сигнал для ИМ (открытие).
	QD	BOOL	Сигнал для ИМ (закрытие).
	STATUS	BYTE	ESR-код.
	PO	BYTE	Уставка положения штор (0 – полностью закрыто, 255 – полностью открыто).
Входы-выходы	CX	<sup>1</sup>	Данные календаря.
Параметры	sunrise_offset	TIME	Задержка активации относительно рассвета.
	sunset_preset	TIME	Смещение для преждевременной активации относительно заката.
	Shade_delay	TIME	Задержка активации.
	Shade_pos	BYTE	Положение штор при затемнении.
Используемые модули	<sup>1</sup> OSCAT_BASIC.CALENDAR		

Рис. 7.16. Внешний вид ФБ **BLIND\_SHADE\_S** на языке CFC

Функциональный блок **BLIND\_SHADE\_S** представляет собой интерфейсный модуль для автоматического затемнения помещения с помощью штор для защиты от ярких солнечных лучей. На вход-выход **CX** подается экземпляр структуры **OSCAT\_BASIC.CALENDAR**, который может быть

получен с выхода ФБ **CALENDAR\_CALC** из библиотеки **OSCAT\_BASIC**. Структура содержит информацию о текущем системном времени, времени рассвета, заката и др.

Блок может работать в режиме затемнения или режиме ручного управления. В режиме затемнения шторы закрываются в то время суток, когда солнечный свет наиболее ярок и может создавать неудобства. Режим затемнения активируется при одновременном выполнении следующих условий:

- предыдущий ФБ системы управления работает в автоматическом режиме: входы **UP** и **DN** ФБ **BLIND\_SHADE\_S** имеют значение **TRUE**;
- получено разрешение на работу ФБ **BLIND\_SHADE\_S**: вход **ENABLE** имеет значение **TRUE**;
- солнечный свет падает в окно: азимут солнца находится в пределах (**HORZ1...HORZ2**), а высота солнца над горизонтом не превышает значения **VERT**;
- солнце уже взошло и еще не зашло: текущее системное время находится в пределах (время рассвета + **sunrise\_offset**...время заката – **sunset\_preset**). Время рассвета и время заката считываются с входа-выхода **CX**;
- есть сигнал от датчика солнца: вход **SUN** имеет значение **TRUE**. Если вход **SUN** принимает значение **FALSE**, то блок продолжает работать еще в течение времени **Shade\_delay** – это позволяет избежать ложных отключений блока из-за проплывающих облаков.

В режиме затемнения выходы блока имеют следующие значения:

- выходы **QU** и **QD** имеют значение **TRUE**, чтобы следующий блок системы работал в автоматическом режиме;
- на выход **PO** передается значение параметра **Shade\_pos** (уставка положения штор при затемнении). Если значение **Shade\_pos** превышает заданную степень открытия штор **PI**, то на выход **PO** передается значение **PI**;
- Выход **STATUS** имеет значение **151**.

В режиме ручного управления на выходы **QU/QD/PO/AO/STATUS** передаются значения входов **UP/DN/PI/AI/S\_IN**. Если сигнал открытия входной двери **ALERT** имеет значение **TRUE**, то происходит открытие штор (**QU = TRUE, QD = FALSE, PO = PI, STATUS = 152**). Обработка сигнала открытия двери имеет приоритет над режимом затемнения и режимом ручного управления.

Ниже приведен пример использования ФБ совместно с ФБ [BLIND CONTROL S](#):

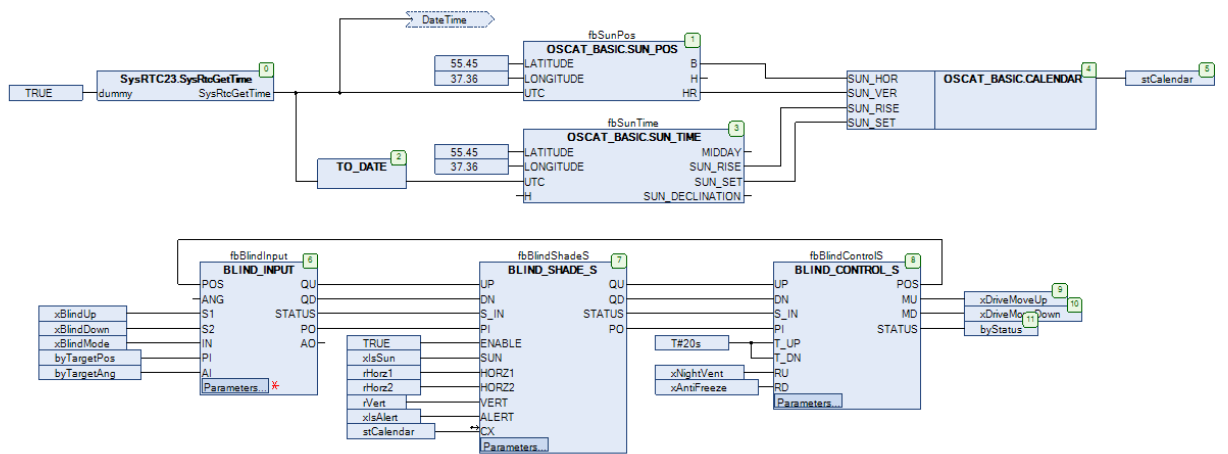


Рис. 7.17. Пример работы с ФБ **BLIND\_SHADE\_S** на языке CFC (рисунок хорошо масштабируется)