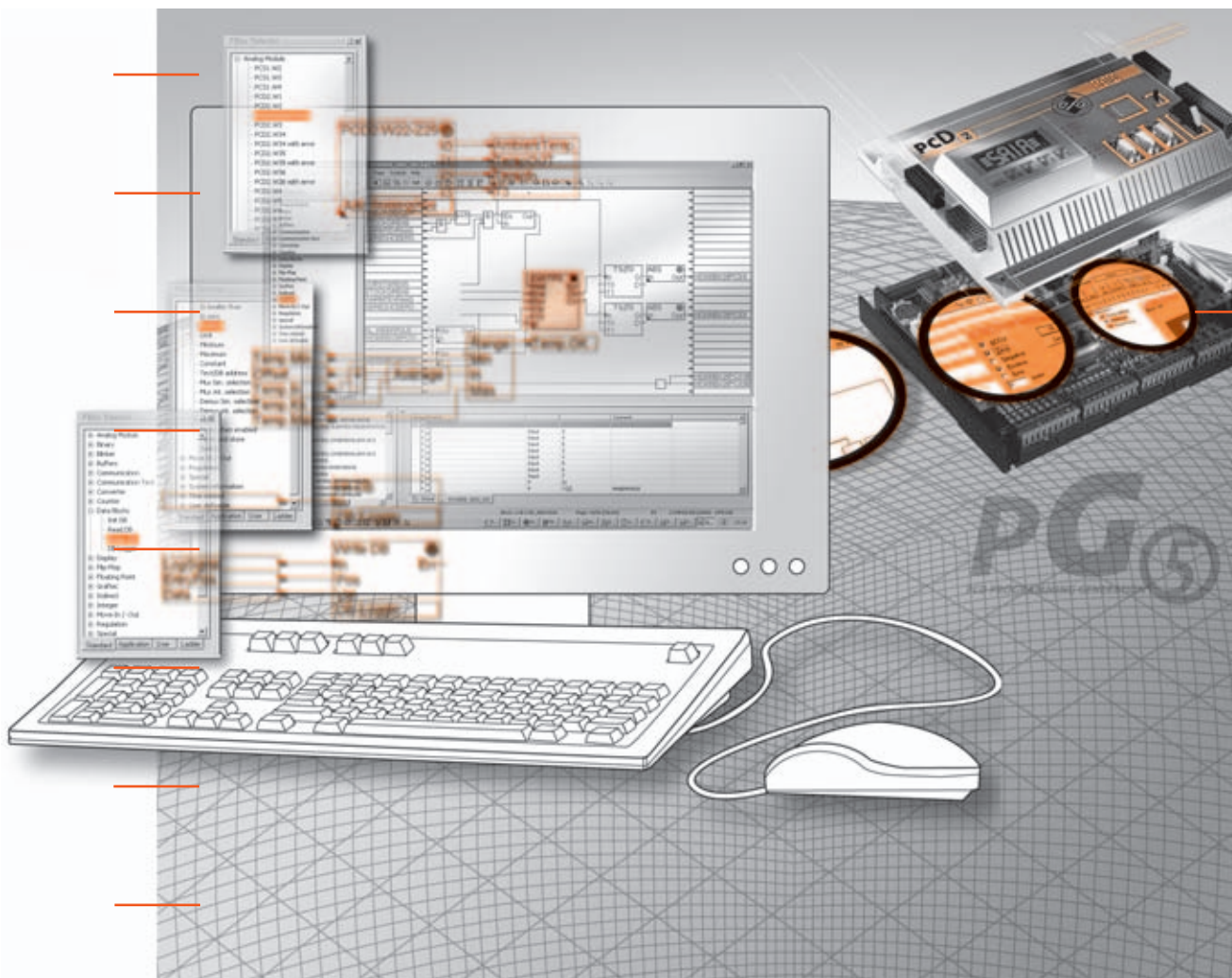




# Визуальное программирование контроллеров SAIA®PCD

Удобное и эффективное решение задач



## Преимущества от использования стандартных библиотек PG5

- **Многokратное упрощение программирования** за счет применения функциональных блоков..
- **Функции контроля, сравнения, коммуникаций, временных измерений и т.п.** параметрируются, а не программируются.
- **Встроенные окна отладки позволяют сократить работу по выставлению параметров сложных функциональных блоков.** Можно получить доступ к операндам и результату их работы, значительно повышая точность параметрирования.
- **Автоматическая проверка версии библиотек и своевременное их обновление.** После обновления функции могут сосуществовать в одном проекте с предыдущими версиями.
- **Проверка синтаксиса позволит избежать ошибок в программе.**

## Характеристики стандартной библиотеки функций PG5

- **250 функций разделены на 21 группу.**
- **Объединяет все базовые функции** такие, как логические операции, таймеры, счетчики и основные математические функции.
- **Комплексные математические и логические операции** в дополнительных функциональных блоках.
- **Поддерживают все специальные функции операционной системы контроллеров.**
- **Коммуникационные функциональные блоки** позволяют организовать фирменную промышленную сеть SAIA®S-BUS поверх Ethernet TCP/IP или RS-485/232, а также для разработки специфических протоколов.
- **Функциональные блоки для считывания или записи данных,** полученных через аналоговые каналы имеют функцию прямого параметрирования для входных каналов или предустановки входных элементов таких, как Pt100, Ni100.
- **Временные функции,** включая таймеры или функции, построенные на встроенных часах реального времени.
- **Файлы помощи,** встроенные в каждый блок.

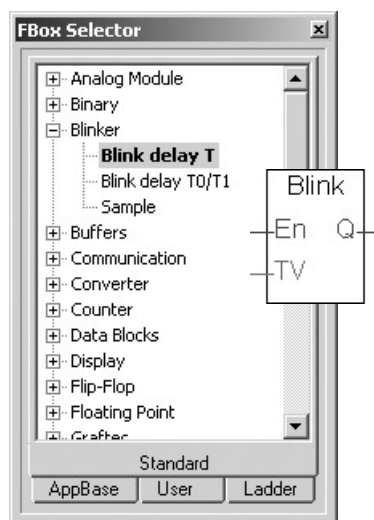
## Общие характеристики функциональных блоков

### Содержание

Общие характеристики функциональных блоков .....	2
КОPLA .....	3
Логические операции .....	4
Переключатели и счетчики .....	5
Операции с целыми числами .....	6
Команды переноса .....	7
Операции с числами с плавающей точкой .....	8
Буферы .....	9
Конверторы, регуляторы и блоки данных .....	10
Функции счета времени, часы и блинкеры .....	11
Сети и коммуникации .....	12
Системная группа и специальная группа .....	14
Группа для работы с дисплеем и аналоговая группа .....	15
Технические характеристики и данные для оформления заказа .....	16

### Разделение на группы

Все функциональные блоки разделены на группы: Стандартные (Standard), Специальные (AppBase) и Произвольные (User - определяются разработчиком). Дополнительные библиотеки для отопления, вентиляции и кондиционирования воздуха или для модемной связи хранятся в группе специальные (Application).



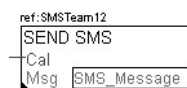
Функциональный блок устанавливается на поле программирования с помощью перетаскивания из поля группы на поле программирования. Основные группы объединяют не только обычные логические или арифметические действия, но и большое количество полезных системных функций.

### Разделение на типы данных

Каждый тип данных определяется своим цветом. Это делает программирование более наглядным и читабельным.

Бинарные данные	красный	
Целые числа	зеленый	
Числа с плавающей точкой	желтый	

Текстовые данные вводятся в поле функционального блока.



Входы, которые активируются переходом состояния, отмечены треугольником.

Статический вход:  
Вход по изменению состояния:



## KOPLA в FUPLA

### Окно параметров с режимом просмотра

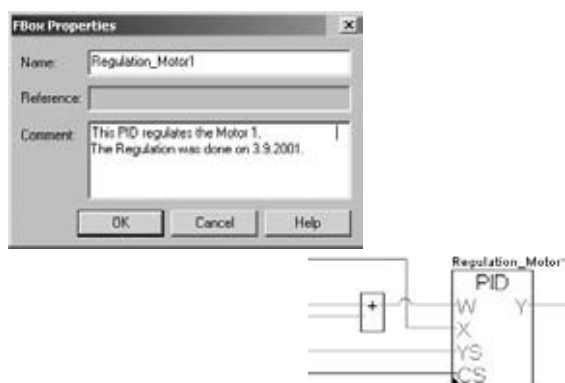
Для предотвращения ошибок при установке связей функциональных блоков, в нижней части блока находится треугольник, с помощью которого можно перейти к окну настройки.



Окно настройки служит не только для установки параметров функциональных блоков, но оно позволяет также наблюдать изменение каждого параметра в процессе работы. Если это необходимо, параметры могут быть изменены без остановки контроллера. Например, можно видеть параметры работы контроллера и оптимизировать коэффициенты ПИД регулятора.

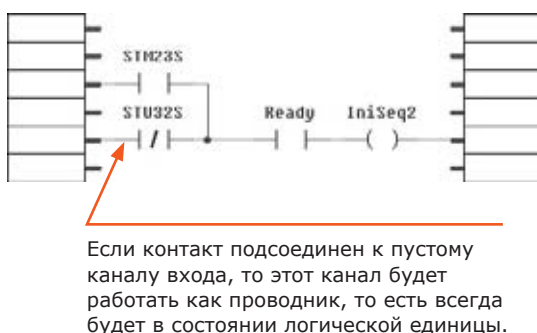
### Документирование

Для улучшения документирования и комментирования программы, каждый функциональный блок имеет имя и поле для комментария. Имя не служит для идентификации: оно может использоваться для связи блоков между собой. Например, все функциональные блоки, которые общаются через один интерфейс, могут иметь одно имя. Имя служит для справки и упрощает работу с функциональными блоками.



### KOPLA

Редактор KOPLA (контактно-релейная схема) является интегрированной частью пакета. В отличие от других графических программных оболочек, функциональные блоки и контактно-релейные элементы могут быть установлены в одном окне.



# Логические операции

## Логические операции

Стандартные логические операции: все функциональные блоки могут иметь от 2 до 10 входов



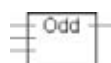
**И**



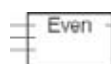
**Исключающее ИЛИ**



**ИЛИ**



**Нечетные входы**



**Четные входы**

В дополнение к базовым операциям данная группа имеет блоки со следующими функциями:

### Фронт



Функциональный блок, который распознает положительный фронт сигнала. На выходе блока появляется единица, когда вход меняет свое значение с нуля на единицу.

### Переместить



Логическое состояние входа транслируется на выход. Этот блок используется для прямой передачи значения входа контроллера на его выход.

### Установить в единицу



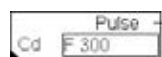
Выход данного блока всегда единица.

### Установить в ноль



Выход данного блока всегда ноль.

### Импульсный сигнал



Как только на адрес флага подается единица <sup>1)</sup>, блок начинает генерировать импульсы, которые длятся до конца циклического блока.

<sup>1)</sup> Выбор осуществляется между установкой и сбросом.

## Мультиплексоры

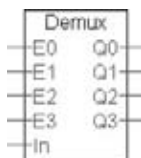
### Мультиплексор с двоичным управлением



Передаёт двоичное значение с входа I0-I7 на выход, когда соответствующий вход E0-E7 получает единицу.

Если единица приходит сразу на несколько входов E, то преимущество имеет номер входа с меньшим номером: вход E0 имеет максимальный приоритет, E7 - минимальный.

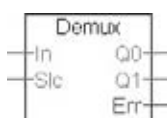
### Демультимплексор с двоичным управлением



Передаёт значение, когда соответствующий вход E0-E7 получает единицу.

Если на входе E0 - ноль, то соответствующий выход Q0 имеет также ноль.

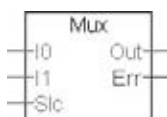
### Демультимплексор с управлением целым числом



Передаёт значение с входа In на один из выходов Q0-Q7. Выбор выхода осуществляется целым числом, поданным на вход Slc (Выбор).

Все невыбранные выходы имеют значение ноль. Выход Err (Ошибка) переходит в единицу, если на вход Выбор подается некорректное значение.

### Мультиплексор с управлением целым числом.



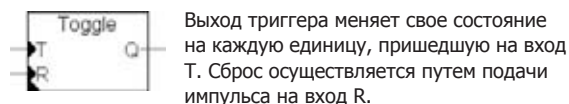
Передаёт бинарное значение с одного из входов E0-E7 на выход. Выбор входа осуществляется с помощью входа Slc (Выбор).

Значение входа Выбор соответствует номеру выбранного входа. На выходе Err (Ошибка) появляется единица, если на вход Выбор поступило некорректное значение.

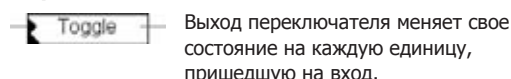
## Триггеры и счетчики

### Триггеры

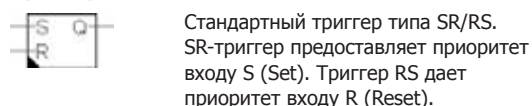
#### Триггер со сбросом



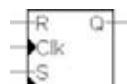
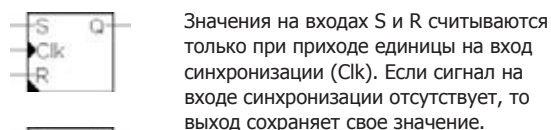
#### Переключатель



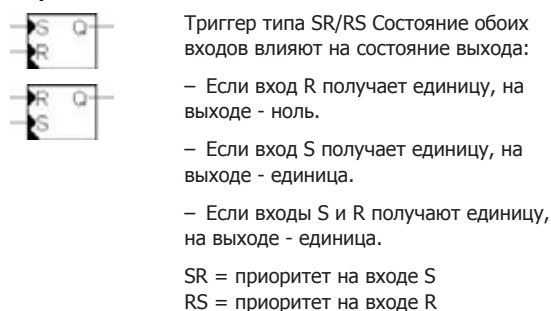
#### Триггер SR/RS типа



#### Триггер SR/RS с синхронизацией



#### Триггер SR/RS с динамическим переключением



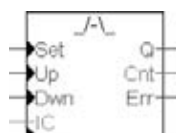
### Счетчики

Все контроллеры SAIA PCD имеют 31 бит для хранения данных счета. Таким образом, максимальное значение счета  $2^{31}-1$ . Если значение счета превысит это значение, на выходе Err счетчика появится единица.

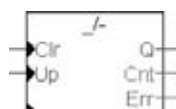
Контроллеры имеют различные типы счетчиков. Таймеры и счетчики могут использовать одни и те же адреса. Заводские установки резервируют адреса с 32 до 1599 для счетчиков. Эти установки можно изменить в разделе Software

Стандартные счетчики:

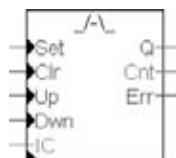
#### Счетчик вверх и вниз с предустановкой



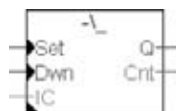
#### Счетчик вверх с обнулением



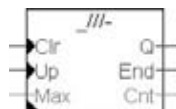
#### Счетчик вверх и вниз с предустановкой и обнулением



#### Счетчик вниз с предустановкой



#### Круговой счетчик с обнулением



## Операции с целыми числами

Целые числа могут использоваться в регистрах, блоках данных, таймерах и счетчиках.

Группа "Арифметика с целыми числами" объединяет функциональные блоки, которые позволяют выполнять операции с целыми числами. Целые числа могут принимать значение от -2147483648 до 2147483647.

Группа целочисленной арифметики состоит из трех частей:

### Арифметические операции с целыми числами

Базовые математические операции в рамках 32 бит.

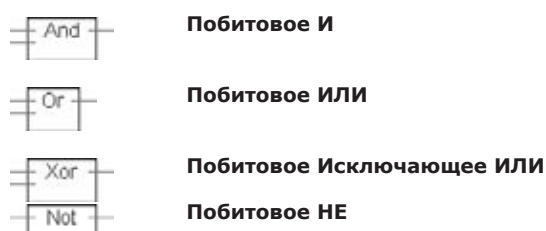


Эти операции выполняются между двумя целыми числами побитно. Например, операция "Побитовое И":


```

Вход   0000 0000 1111 1111
Вход   1111 0000 1111 0000
-----
Выход  0000 0000 1111 0000
    
```

Эти функциональные блоки выполняют действия в рамках 32 бит.



### Абсолютное значение

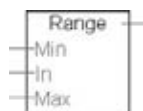
 На выходе блока выводится абсолютное значение входного сигнала.

### Среднее

 На выходе блока выводится среднее входных значений.

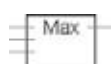
## Сравнение целых чисел

### Диапазон



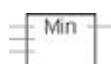
На выходе блока появляется единица, если число на входе попадает в диапазон между значениями Min и Max.

### Maximum



На выход подается максимальное значение, из всех поданных на вход.

### Minimum



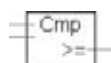
На выход подается минимальное значение, из всех поданных на вход.

### Лимит



На выход подается значение входа, ограниченное значениями Min и Max.

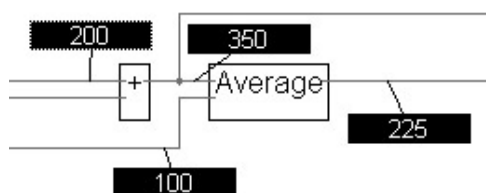
**Равен**  
**Больше**  
**Больше или равен**  
**Меньше**  
**Меньше или равен**  
**Равен нулю**



Двоичный выход будет иметь значение единица, если выполняется указанное условие для входов блока.

### Пример

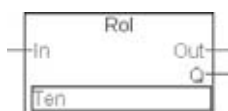
Сложение двух целых чисел с последующим вычислением среднего.



# Манипулирование целочисленными значениями

## Манипулирование целочисленными значениями

### Циклический сдвиг влево/вправо



Значение целого числа, подаваемого на вход, вращается влево или вправо. Степень вращения (количество бит) указывается в поле функционального блока. Результат (целое значение) подается на выход.

### Сдвиг влево/вправо



Работает так же, как циклический сдвиг. В дополнение к значению сдвига определяется заполнение освободившихся полей: нули или единицы.

### Переместить



Передаёт значение входа непосредственно на выход без изменения.

### Переместить при условии



Перемещает целое число с входа на выход, если на вход En подается единица. В противном случае, на выходе будет ноль.

### Переместить и сохранить



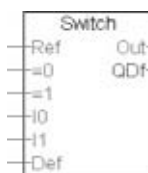
Перемещает целое число с входа на выход, если на вход Sto подается единица. Значение выхода сохраняется до следующей единицы на входе Sto.

### Адрес текстового блока или блока данных



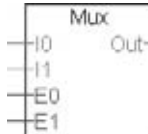
Выдает цифровое значение адреса текстового блока или блока данных.

### Переключить

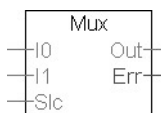


Передаёт значение входа In, если значение на входе =n равно значению Ref (ссылка). В том случае, если совпадений нет, то на выход подается значение входа Def (Базовые установки).

### Бинарно-целочисленный мультиплексор



Передаёт значение одного из входов на выход. Вход определяется с помощью двоичного или целочисленного значения.



## Группа битовых манипуляций

Функциональные блоки этой группы позволяют манипулировать как отдельными битами, так и их строками. Оставшиеся в регистре биты в результате этих операций не изменяются.

### Входное значение

0000 0000 0000 0000 0000 0000 0000 0000  
31 0

### Переместить биты Q0-Q7 на значение

X XXXX XXX | ←

### Результат

0000 0000 0000 0000 000X XXXX XXXX 0000  
31 5 0

- Move In Bit
- Move In Digit
- Move In Nibbel
- Move In Byte
- Move In Word



Все блоки Move-In записывают один или несколько битов, цифр, полубайтов, байтов или слов по определенному адресу 32-битного регистра.

Точный адрес записи определяется значением Offset.

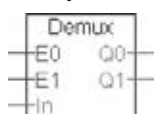
- Move Out Bit
- Move Out Digit
- Move Out Nibbel
- Move Out Byte
- Move Out Word



Все блоки Move-Out считывают один или несколько битов, цифр, полубайтов, байтов или слов по определенному адресу 32-битного регистра.

Точный адрес считывания определяется значением Offset.

### Бинарно-целочисленный демультиплексор



Передаёт значение входа на один из выходов. Выход определяется с помощью двоичного или целочисленного значения.



# Числа с плавающей точкой

Числа с плавающей точкой могут принимать значения от  $2,710505E-20$  до  $-9,223371E+18$  и операции с ними обозначаются в редакторе функциональных блоков (FUPLA) желтыми линиями.

Необходимо отметить, что все участвующие в операции числа должны иметь один и тот же формат. Все контроллеры используют единый формат чисел с плавающей точкой - быстрая плавающая точка. Группа функциональных блоков "Конвертеры" преобразовывают значения в формат IEEE.

Результаты операций с числами с плавающей точкой могут храниться в регистрах или блоках данных.

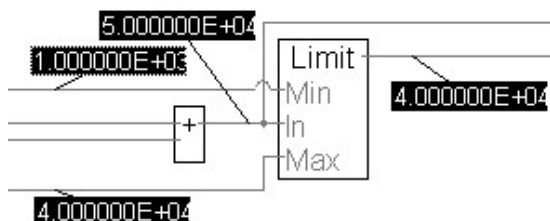
## Арифметические действия с числами с плавающей точкой

Приводим основные функциональные блоки для выполнения математических действий

-  **Сложение**
-  **Вычитание**
-  **Умножение**
-  **Деление**
-  **Квадратный корень**
-  **Натуральная экспонента**
-  **Натуральный логарифм**
-  **Синус**
-  **Косинус**
-  **Арктангенс**

### Пример

Два числа с плавающей точкой складываются, а результат - 50000, "обрезается" функциональным блоком Limit с максимальным значением 40000.



## Другие действия с числами с плавающей точкой

### Переместить



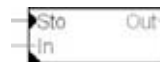
Пересылает входное значение непосредственно на выход. Целое значение также пересылается на выход.

### Переместить при условии



При наличии единицы на входе En (Разрешить), входное значение пересылается на выход. В противном случае на выходе будет ноль.

### Переместить с сохранением

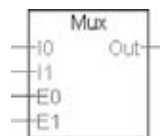


При прохождении положительного фронта на вход Sto (Сохранить), входное значение будет сохранено на выходе до прохождения следующего положительного фронта.

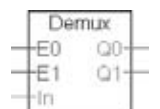
### Мультиплексор



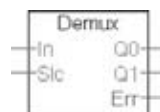
Передаёт значения выбранного входа на выход. Вход может быть выбран двоичным или целым значением.



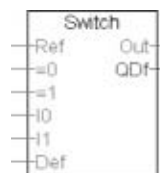
### Демультимплексор



Передаёт значение входа на выбранный выход. Выход может быть выбран двоичным или целым значением.



### Переключить

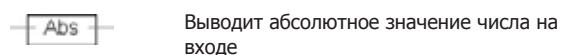


Передаёт значение входа In, если значение на входе =n равно значению Ref (ссылка). В том случае, если совпадений нет, то на выход подается значение входа Def (Базовые установки).

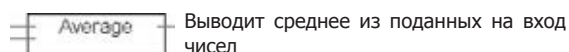
# Числа с плавающей точкой Буфер

## Сравнение чисел с плавающей точкой

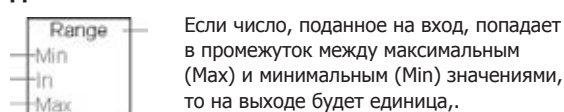
### Абсолютное значение



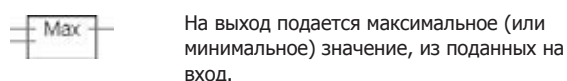
### Среднее



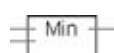
### Диапазон



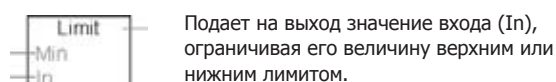
### Maximum



### Minimum



### Limit



### Равен

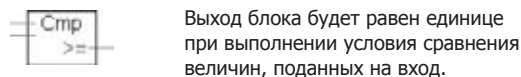
### Больше

### Больше или равен

### Меньше

### Меньше или равен

### Равен нулю



## Операции с буфером

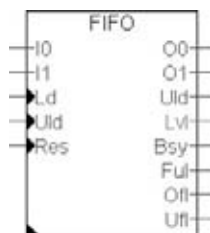
Функциональные блоки этого семейства дают возможность сохранять различных типов данных. Размер буфера может устанавливаться разработчиком.

### Сокращения, которые используются далее:

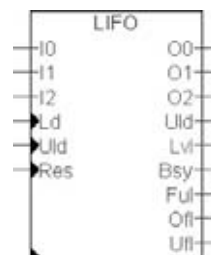
- In Входное значение, которое загружается в буфер
- Ld Загрузить значение в буфер
- Uld Выгрузить значение из буфера
- Res Сброс
- Out Выход: последнее выгруженное значение
- Uld Выгрузить: показывает, что значение было выгружено из буфера
- Lvl Уровень: количество значений в буфере
- Bsy Занято: как минимум, одно значение находится в буфере
- Ful Буфер полностью заполнен
- Off Буфер переполнен (Ful + 1)
- Ufl Пуст: выдается на команду Выгрузить при пустом буфере

FIFO - функциональный блок управляет буфером по принципу "Первый зашел/первый вышел". LIFO – по принципу "Последний зашел/первый вышел".

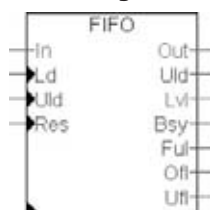
### FIFO Binary



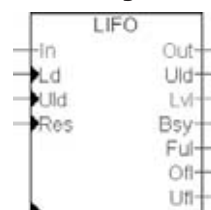
### LIFO Binary



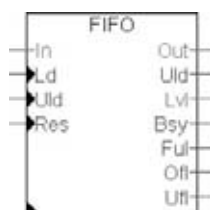
### FIFO Integer



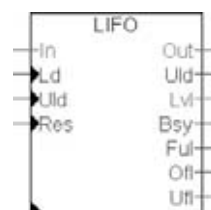
### LIFO Integer



### FIFO Float



### LIFO Float



## Конвертеры, регуляторы и блоки данных

Различные типы данных не могут участвовать в вычислениях (например, двоичное значение не может быть сложено с целочисленным непосредственно). Поэтому в пакете существует группа функциональных блоков, которые позволяют преобразовывать данные в нужный тип.

### Конвертер целочисленных значений

#### Число с плавающей точкой – в целое

Этот функциональный блок преобразовывает число с плавающей точкой в целое и наоборот.

#### Целое в двоичное быстро

#### Целое в двоичное реверсивно быстро

#### Целое в двоичное побитно со сдвигом

#### Целое в битовую последовательность

#### Целое в двоичное 1-8

#### Целое в двоичное 1-24

Все эти функциональные блоки передают последовательность битов из значения на входе (целое) на выход. Количество и последовательность этих битов зависит от выбора блока. "Быстро" значит, что результат не выводится через цифровые выходы, а, вместо этого, записывается во флаг или выход напрямую.

### Конвертер двоичных чисел

#### Двоичное в целое 1-8

#### Двоичное в целое 1-24

#### Двоичное в целое быстро

#### Двоичное в целое реверсивно быстро

#### Двоичный счет 1-8

#### Двоичный счет быстро

#### Битовая последовательность в целое

#### Битовая последовательность в целое со сдвигом

Все эти функциональные блоки конвертируют двоичные значения в целочисленные. "Быстро" значит, что результат не записывает в целочисленный выход, а сохраняется в регистре.

### Конвертер форматов чисел с плавающей точкой

#### Из IEEE- в PCD-формат

#### Из PCD- в IEEE-формат

Обратите внимание, что все контроллеры оперируют с числами в формате Motorola FFP. Эти функциональные блоки преобразуют IEEE формат в Motorola и обратно.

### Конвертер VCD чисел

#### Конвертер VCD чисел

#### VCD в целое

#### VCD в целое быстро

#### VCD в целое реверсивно быстро

#### Целое в VCD

#### Целое в VCD реверсивно быстро

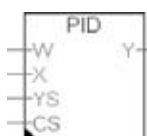
#### Целое в VCD быстро

Эти функциональные блоки преобразуют VCD-значения в целочисленные различными способами. "Быстро" значит, что результат не записывается в целочисленный выход, а будет скопирован в регистр напрямую.

### Регуляторы

Каждый контроллер имеет встроенный ПИД-регулятор

#### PID



ПИД-регулятор с настраиваемым значением холодного старта и разрешением:

T0 (время несущей) должно быть в десять раз больше, чем время выполнения цикла программы, но не меньше 80 мс.

### Блоки данных

Блоки данных нужны для хранения большого количества информации. Чаще всего, их используют для периодического сохранения данных для вывода на дисплей или принтер.

Блок данных – это 32-битный массив данных, которые располагаются в пользовательской памяти.

#### Init DB



Инициализирует выбранный блок данных и заполняет его нулями. Инициализация происходит при изменении входного сигнала (In) с нуля на единицу.

#### DB Logger



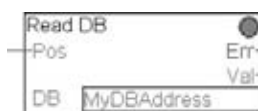
Записывает входное значение в блок данных. Блок данных автоматически ставит метки и выдает единицу на выход при заполнении.

#### Write DB




Записывает входные значения (Val) в блок данных. Вход Pos определяет адрес записанного значения.

#### Read DB



Считывает элементы блока данных. Вход Pos определяет адрес необходимого элемента. Результат выводится как целое значение на выход Val.

# Таймеры

Контроллеры PCD имеют различные типы таймеров. Таймеры и счетчики делят одно адресное пространство. Заводские настройки резервируют первые 31 адрес для таймеров. Оставшиеся адреса (до 1599) зарезервированы для счетчиков. Однако это соотношение вы можете изменить в настройках .

Функциональные блоки таймеров всегда оперируют с временной разверткой 100 мс. С помощью программных настроек можно уменьшить частоту развертки таймеров до 10 мс. Однако это будет иметь значение только для таймеров, используемых в макроассемблере. На таймеры, используемых в FUPLA это не повлияет.

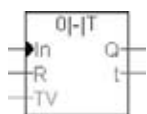
## Таймеры: базовые функции



**Включить задержку**



**Выключить задержку**



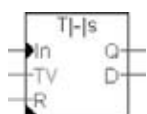
**Выключить задержку со сбросом**



**Включить/выключить задержку**

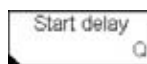
## Таймеры: специальные функции

### Задержка с сохранением



После получения единицы на вход In, блок выжидает время, указанное на входе TV, и выдает единицу на выход Q. Состояние выхода будет сохраняться до получения единицы на вход R.

### Задержка при перезагрузке



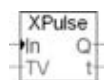
Когда включается контроллер (холодный запуск), состояние выхода Q остается без изменений (Ноль) в течение времени, которое задается в окне настроек функционального блока. После истечения времени задержки выход постоянно находится в положении Единица.

### Импульс



Когда на вход In приходит единица, выход Q начинает генерировать импульсы с длительностью, заданной на входе TV. Если входной сигнал переходит в ноль, выход Q также переходит в ноль (вне зависимости от значения TV).

### Одиночный импульс




Когда на вход In приходит единица, на выход Q подается импульс с длительностью, заданной на входе TV. Цикл повторяется с приходом очередной единицы на вход In.

### Длительность цикла



Измеряет длительность заданного числа циклов. Максимальное значение сохраняется.

## Таймеры: функции времени

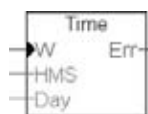
Практически все контроллеры SAIA PCD (кроме PCD1.M110) имеют встроенные часы реального времени. Эти часы могут быть настроены с помощью PG5 ⇒ Online Configurator . Кроме того, они содержат календарь по числам и неделям.

### Считать время (аппаратные часы)



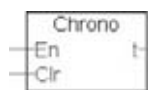
Считывает время и дату со встроенных часов контроллера и выводит их через два целочисленных выхода.

### Записать время (аппаратные часы)



Записывает время в часы реального времени контроллера. Удобно использовать при наличии сигнала точного времени.

### Хронометр



Эта функция служит для измерения временных интервалов.

Часы реального времени могут синхронизироваться через полевою шину SAIA S-BUS. Функциональный блок "Broadcast Clock" (из семейства коммуникационных блоков) синхронизирует часы реального времени всех подчиненных станций в сети.

## "Мигалки"

### Мигалка T



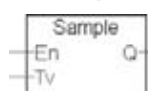
Выход Q начинает выдавать импульсы пока на вход En (разрешить) подается единица. Длительность пульсации определяет значение на входе TV (с шагом 0,1 с).

### Мигалка T0/T1



Выход Q начинает выдавать импульсы пока на вход En (разрешить) подается единица. Длительность импульса и паузы определяют значения на входе T0 и T1 (с шагом 0,1 с).

### Семплер

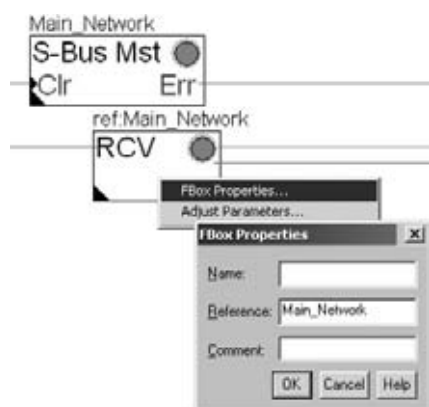


Пока на входе En (разрешить) находится единица, выход Q генерирует один импульс на цикл. Длительность пульсации определяет значение на входе TV (с шагом 0,1 с).

## Шины, сети и коммуникации

### Коммуникации: SASI

Аббревиатура SASI переводится как "Назначить последовательный интерфейс". Для того, чтобы использовать последовательный интерфейс контроллера, необходимо настроить режим его работы. Как правило, эти функциональные блоки устанавливаются в начале программы (перед блоками "Послать" или "Получить").



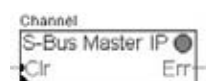
Назначение блоков SASI ("Назначить последовательный интерфейс"), SEND ("Послать") и RECEIVE ("Получить") легко понять из их названия. Функциональные блоки одной сети или шины используют одни и те же имена.

#### SASI SBus Master



Инициализирует последовательный интерфейс как ведущий ("Master") в шине SAIA S-BUS. После этого данный контроллер может считывать и записывать данные в подчиненные станции ("Slave").

#### SASI SBus Master IP



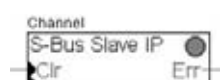
Инициализирует последовательный интерфейс как ведущий ("Master") в шине SAIA S-BUS поверх Ethernet TCP/IP. После этого данный контроллер может считывать и записывать данные в других мастерах и подчиненные станции ("Slave").

#### SASI SBus Slave



Инициализирует последовательный интерфейс как ведущий ("Slave") в шине SAIA S-BUS.

#### SASI SBus Slave IP

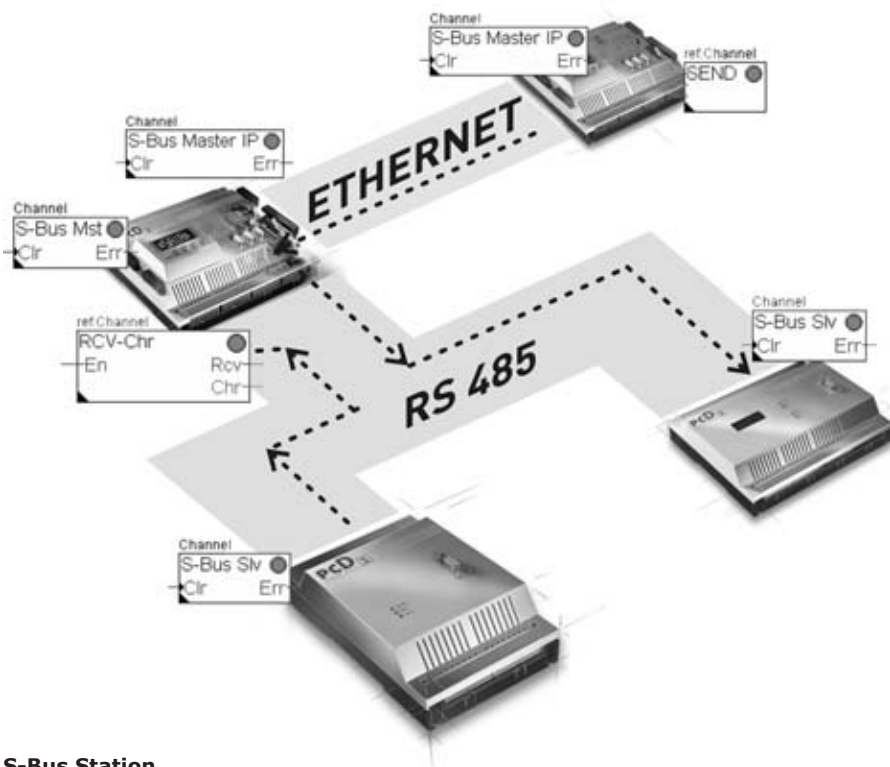


Инициализирует последовательный интерфейс как ведущий ("Slave") в шине SAIA S-BUS поверх Ethernet TCP/IP.

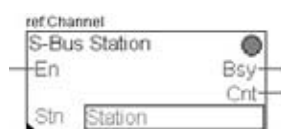
#### SASI S-Bus Extended



Похож на функциональный блок "S-Bus Master", но с дополнительными параметрами.



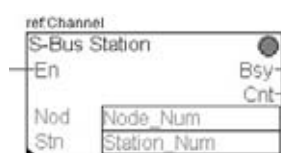
#### S-Bus Station



Этот функциональный блок предлагает основные функции по управлению станцией (на усмотрение разработчика).

Применение данного блока оказывает влияние на реакцию станции на команды функциональных блоков SEND ("Послать") и RECEIVE ("Получить").

#### S-Bus Station IP



Такой же блок, но для сети Ethernet TCP/IP.

#### SASI RIO



Инициализирует порт для связи с распределенными системам сбора данных (Удаленные модули ввода/вывода - RIO).

#### SASI Diagnostic



Предоставляет данные диагностики всех портов, открытых с помощью блоков SASI (на усмотрение разработчика).

#### SASI Mode D

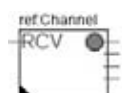


Инициализирует последовательный интерфейс в режиме дуплекса в связях типа "Точка-точка".

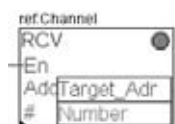
Эти функциональные блоки позволяют управлять ресурсами контроллеров через полевую шину SAIA S-BUS. Эти команды не являются циклическими, но регулярный опрос ресурса можно организовать программно.

### Коммуникации: Прием

Данные функциональные блоки считывают ресурсы контроллеров (такие как, флаги, таймеры, счетчики, регистры, блоки данных, входы и выходы) с подчиненных станций. В именах блоков, "Быстро/Quick" показывает, что эти команда позволяет получить несколько параметров одного контроллера в одной посылке и сразу сохранить их в памяти. Блок типа "Много/Multiple" позволяет одновременно считывать один параметр (например, вход 32) с нескольких контроллеров.



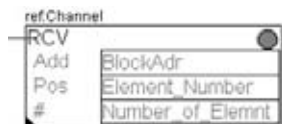
**Получить двоичное число**  
**Получить целое**  
**Получить десятичное**



**Получить двоичное Быстро**  
**Получить целое Быстро**  
**Получить десятичное Быстро**



**Получить двоичное Много**  
**Получить целое Много**  
**Получить десятичное Много**



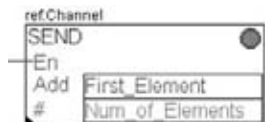
**Получить блок данных**

### Коммуникации: Передача

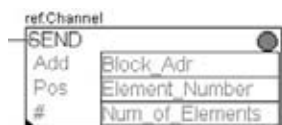
Данные функциональные блоки записывают ресурсы контроллеров (такие как, флаги, таймеры, счетчики, регистры, блоки данных и выходы) в подчиненные станции. В именах блоков, "Быстро/Quick" показывает, что эта команда позволяет послать несколько параметров в один контроллер в одной посылке и сразу сохранить их в памяти.



**Передать двоичное**  
**Передать целое**  
**Передать десятичное**



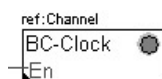
**Двоичное Быстро**  
**Целое Быстро**  
**Десятичное Быстро**



**Передать блок данных**



**Передать команду**



**Синхронизировать время**

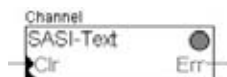
Рассылает время на все станции сети SAIA®S-Bus.

### Блоки для работы с текстом

Текстовый режим (так называемый MC-mode) служит для передачи символов ASCII и их последовательностей через серийный интерфейс, не используя специальный протокол. Этот режим используется в следующих случаях:

- При написании собственного протокола;
- При посылке символов в принтер или на дисплей.

#### SASI Text



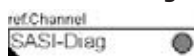
Инициализирует один из последовательных каналов (называемых каналами) в текстовом режиме. Используется, когда необходимо послать телеграмму без специального протокола.

#### SASI Char



То же, но со специальными дополнительными параметрами.

#### SASI Text Diagnostic



Применение этого блока необязательно. Он предоставляет удобные функции диагностики при работе через последовательные интерфейсы.

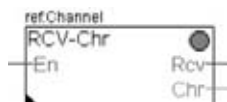
Все следующие функциональные блоки требуют наличие SASI-блоков в начале файла.

#### Edit value



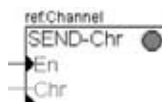
Отображает меню редактирования на подключенном дисплее. Позволяет редактировать значение, записанное в регистре, с помощью внешнего дисплея.

#### Receive Character



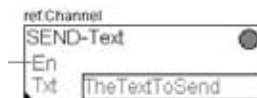
Получает ASCII-символ с помощью последовательного интерфейса. На выход блока подается целое число в формате ASCII.

#### Transmit Character



Передает от 1 до 4 ASCII-кодов через указанный интерфейс.

#### Transmit Text



Передает текст через указанный интерфейс.

## Системные блоки

### Системные блоки

Системные блоки считывают и записывают наиболее нужные данные о системе. Эти функциональные блоки могут пригодиться для проверки корректности настройки системы или программного обеспечения.

#### Read device type

Считывает тип контроллера

Тип	ASCII	Число
PCD1	D1	1
PCD2	D2	2
PCD4	D4	4
PCD6	D6	6

#### Read Firmware version

Считывает номер версии операционной системы

#### Read CPU type

Считывает тип процессора в контроллере

Контроллер	ASCII	Число
PCD1		
PCD2.M1..	M1_	10
PCD4.M11..	M11	11
PCD4.M24..	M24	24
PCD6.M1..	M1_	10
PCD6.M54..	M54	54

#### Read system counter

Считывает системное время. Системное время работает с точностью 0,001 секунды. Удобно для точной регистрации временных интервалов.

#### Read S-Bus station

Считывает номер станции (контроллера) в сети SAIA S-BUS.

#### Write S-Bus station

Перезаписывает номер станции в сети SAIA S-BUS в ходе выполнения программы. Если программа хранится в ПЗУ, то новый номер будет записан в базовое ОЗУ контроллера.

#### Read S-Bus PGU parameter

Считывает все параметры шины SAIA S-BUS, хранящиеся в контроллере.

#### Read user program name

Считывает имя программы, связанное с контроллером (процессором).

### Специальные блоки

Данные блоки выполняют специальные функции операционной системы.

#### System selftest

Этот функциональный блок запускает самотестирование контроллера. Если тест не выполняется, то, на соответствующем выходе блока (от 0 до 11), появится единица. К тому же, на выходе "Общая ошибка" появится единица при ошибке в любом тесте.

- 0 Доступная память ОЗУ
- 1 ОЗУ для рабочей программы
- 2 Контрольная сумма рабочей программы
- 3 Не используется
- 4 Часы реального времени
- 5 Последовательные интерфейсы
- 6 Контрольная сумма операционной системы
- 7 Не используется
- 8 Системная память ОЗУ
- 9 Расширение памяти
- 10 Доступная память
- 11 Не используется

#### XOB Diagnostic

Этот блок сохраняет нужную информацию в случае возникновения исключительной ситуации (XOB). На каждую такую ситуацию нужно предусмотреть отдельный блок.

#### Execute XOB

Немедленно запускает исключительный организационный блок XOB17-XOB19. Практически применяется для синхронизации заданий многопроцессорных систем.

#### Battery

Показывает состояние батареи. Ошибка исчезает через 2 секунды после установки (новой) батареи.

#### Install cyclic task

Этот блок запускает циклический процесс при получении единицы на вход "Enable". Процесс будет повторяться с частотой X.

#### Software Watchdog

Работает только с определенным типом контроллеров. Для информации обратитесь к подробной документации на контроллер.

#### Hardware Watchdog

Реле сторожевого таймера размыкается при возникновении ошибки. Питание контроллера должно подаваться через это реле.

#### Использование EEPROM

Процессоры некоторых контроллеров (PCD1, PCD2 и PCD4.M170) могут размещать информацию в ППЗУ (EEPROM). Это можно использовать при написании программ. Содержимое этих регистров сохранится после отключения питания микросхемы.

## Аналоговые модули и визуализация

### Аналоговые каналы: основные блоки

Базовые функциональные блоки считывают/записывают значения аналоговых сигналов и передают их без изменения (например, без передискретизации или линеаризации).

Аналоговые сигналы преобразуются в цифровой формат (АЦП):

8 бит	0...255
10 бит	0...1023
12 бит	0...4095

За цикл процессор контроллера преобразует значение одного канала в плате ввода/вывода. При включении контроллера все значения устанавливаются в ноль.

**PCD2.W1**  
**PCD2.W2**  
**PCD2.W4**  
**PCD2.W5**  
**PCD4.W1**  
**PCD4.W3**  
**PCD4.W4**  
**PCD4.W8**  
**PCD6.W1**  
**PCD6.W3**  
**PCD6.W4**

### Аналоговые каналы: блоки с расширенными функциями

Блоки данного семейства обладают расширенными функциями:

- Выбор различного типа датчика со встроенной функцией линеаризации сигнала данного сенсора;
- Срабатывание ошибки (с индикацией) при выходе сигнала за указанный диапазон;
- Термин "С ошибкой/with error" в названии функционального блока говорит о том, что он не только отображает код ошибки, но и позволяет передать его другим блокам.

**PCD2.W3**  
**PCD2.W34**  
**PCD2.W34 with error**  
**PCD2.W35**  
**PCD2.W35 with error**  
**PCD2.W36**  
**PCD2.W36 with error**  
**PCD2.W220-Z29**

Блоки ...W500 и ...W600 имеют расширенные функции:

- Функция фильтра;
- Масштабирование различных диапазонов;
- Измерение гистерезиса и наличие логического выхода "Значение </> X".

**PCD2.W6**  
**PCD4.W600**  
**PCD4.W500**  
**PCD4.W5 Temperature**  
**PCD4.W5 Status**

### Блоки для работы с дисплеем

**PCD2.F510** Посылает значение на дисплей PCD2.F5... Плата PCD2.F5.. может отображать до шести цифр (от -99 999 до 999 999). Кроме того, может выводить несколько предустановленных фраз в сочетании с двумя цифрами (0-99):

- Текст "Hlp" + 2 цифры;
- Текст "Err" + 2 цифры;
- 2 цифры без текста;
- Время

#### PCD2.F510 Text

Выводит на дисплей предустановленные слова:

- =SAIA=
- =PCd2=
- HELP
- Error
- Чистый дисплей.

#### D120 Module

Выводит на дисплей PCD7.D120 целочисленное значение. Значение будет отображаться только в том случае, если на вход "En" подается единица. Таким образом, несколько разных функциональных блоков могут отображать свое значение по очереди.

Данный дисплей может быть подключен к обычному дискретному выходу. Это дает возможность обойтись без последовательного интерфейса.

#### D120 Module Duplex

То же, но с функцией подключения двух модулей последовательно.

#### D12 Module D14 Module

Оба эти дисплеи остались из предыдущего PCA семейства контроллеров.

### Текстовые дисплеи

Структура меню в текстовых дисплеях не программируется в FUPLA. Для этого служит специальный пакет - HMI-редактор, который позволяет быстро сформировать меню и который является частью пакета. Некоторые функциональные блоки (например, из семейства HEAVAC) могут передавать свои данные в HMI-редактор. Это означает, что параметры этих блоков могут быть отображены на дисплее.

Текст может быть передан на дисплей с помощью функциональных блоков из семейства передачи текстовых блоков.

# Технические характеристики и информация для оформления заказа

## Технические характеристики

Тип контроллера	Все функциональные блоки работают со всеми типами контроллеров (за исключением блоков, связанных с часами реального времени и сторожевым таймером)
Требования к системе	При определенных обстоятельствах некоторые функции требуют дополнительных компонентов, например, наличие последовательного порта, дисплея, памяти, сторожевого таймера, часов реального времени и т.п.

## Информация для заказа

Тип	Описание
<b>PCD8.P59 000 M9</b>	<b>Стандартные библиотека PG5</b> , основные библиотеки которого описаны в данном документе. Являются частью каждого пакета PG5.  Дополнительные библиотеки:
<b>PCD9.M59 BAS M5</b>	<b>Базовые модемные функциональные блоки Modem-Basis</b> Инициализация и диагностика, определяемые профили, хранение списка телефонных номеров, защита паролем, установка связи, передача или получение данных по событию или расписанию, связь между контроллерами или контроллером и компьютером (SCADA), поддержка шины SAIA S-BUS по телефонным сетям.
<b>PCD9.M59 PAG M5</b>	<b>Modem-Pager (блоки для работы с пейджером)</b> Передача одного или нескольких сообщений на пейджер по событию или расписанию. Поддерживаются TAP и другие национальные протоколы.
<b>PCD9.M59 SMS M5</b>	<b>Modem-SMS (блоки для работы с SMS)</b> Передача одного или нескольких сообщений SMS по событию или расписанию. Поддерживаются UCP и TAP. Получение сообщений и управление контроллером с помощью SMS.
<b>PCD9.M59 DTM M5</b>	<b>Modem-DTMF (блоки для работы с модемом DTMF)</b> Поддерживается получение DTMF-сигналов для передачи инструкций по телефону, то есть управление контроллера можно организовать с помощью телефона с тональным режимом.
<b>PCD9.B59 HVCM5</b>	<b>PG5-HEAVAC</b> Библиотека из более, чем 150 функциональных блоков для решения задач по отоплению, вентиляции и кондиционированию воздуха в инженерных системах. Объекты HEAVAC библиотеки могут быть импортированы в HMI-редактор для отображения на дисплее.
<b>PCD9.C59 EIB M5</b>	<b>EIB-библиотека</b> Функциональные блоки для коммуникации с шиной EIB.

### Saia-Burgess Controls AG

Bahnhofstrasse 18  
CH-3280 Murten / Schweiz

Telefon 026 / 672 72 72  
Telefax 026 / 672 74 99

E-mail: [pcd@saia-burgess.com](mailto:pcd@saia-burgess.com)  
Homepage: [www.saia-burgess.com](http://www.saia-burgess.com)  
Support: [www.sbc-support.ch](http://www.sbc-support.ch)

### InCoSystems Ltd

Россия, 129090, Москва,  
Грохольский пер., 13, строение 2,  
офис 306.

Тел: (495) 974-1223, 937-5778.  
Факс: (495) 974-1224

E-mail: [info@incosystems.ru](mailto:info@incosystems.ru)  
[www.incosystems.ru](http://www.incosystems.ru)