

Modbus-устройства

Modbus-устройства

Содержание

- 1 Введение
- 2 Modbus-протокол. Общая информация
- 3 Регистрация Modbus-устройств в Tekon OPC-сервер
 - ◆ 3.1 Тестовый опрос устройства
 - ◆ 3.2 Тестовая запись значения регистра
 - ◆ 3.3 Особенности представления значений Modbus-регистров
 - ◆ 3.4 Как зарегистрировать битовую маску ошибок
- 4 Настройка SCADA
- 5 Демонстрационный пример

Настройка Modbus в АСУД.SCADA может быть выполнена двумя способами:

- регистрация устройства непосредственно в Tekon OPC-сервер
 - ◆ это новый способ доступный с версии АСУД.SCADA 2.6.3
 - ◆ целесообразно использовать на всех новых объектах где добавляются Modbus-устройства
 - ◆ при этом варианте не используется БД original.gdb все настройки выполняются непосредственно в OPC-сервере
 - ◆ поддерживается практически полный функционал настройки, аналогичный ASUDBase
 - ◇ исключение - возможность выполнить запись в регистр по другому адресу (отличному от адреса чтения) доступна пока только при регистрации в ASUDBase
 - ◆ опрос осуществляется непосредственно OPC-сервером, программа KCSLogger не нужна
- регистрация устройства в программе ASUDBase
 - ◆ целесообразно использовать, если у вас уже есть Modbus-устройства, зарегистрированные в программе ASUDBase
 - ◆ опрос осуществляется программой KCSLogger

Данная статья ориентируется на настройку Modbus-устройств непосредственно в OPC-сервере **версии 2.7.2 (и выше)** и не рассматривает возможность настройки устройств в программе **ASUDBase**

Если, по каким-то причинам вы планируете выполнять настройку устройств в программе **ASUDBase**, либо в АСУД.SCADA **2.5.x и ранее**, а также настройку Modbus с концентратором **КЦС-М** смотри [предыдущую редакцию статьи](#).

АСУД-248 позволяет реализовать взаимодействие с **ЛЮБЫМИ** устройствами, поддерживающими протокол Modbus.

Modbus-устройства могут быть подключены к концентраторам:

- КЦС-IPM / КУН-IPM
- КЦС-М (есть ограничения в типе поддерживаемых устройств, подробнее см. [тут](#))
- напрямую по компьютерной сети.

КЦС-IPM / КУН-IP

Тракт передачи данных:

Устройство Modbus -- RS-232/485 -- КЦС-IPM / КУН-IP -- Компьютерная сеть -- АРМ-диспетчера

Поддерживаются спецификации Modbus-RTU, Modbus TCP

Чтение данных одиночное, групповое

Запись данных одиночная

Число подключаемых устройств до 255* на каждом интерфейсе

* - подробнее см. спецификацию подключаемого устройства и интерфейса подключения.

Внимание:

- Требуется наличие USB-ключа защиты АСУД.SCADA.
В ключе защиты ПО АСУД.SCADA должно быть прописано число **драйверов EXT-модуля**, соответствующее общему числу подключаемых приборов.
Для тестового опроса устройства в конфигураторе OPC-сервера драйвер не требуется.
- Следует придерживаться следующего правила при адресации Modbus-устройств: устройства, подключаемые к одному IP-концентратору должны иметь уникальные адреса (RS-идентификаторы).

Прямое подключение по компьютерной сети

Тракт передачи данных:

Устройство Modbus с Ethernet -- Компьютерная сеть -- АРМ-диспетчера

Остальное аналогично случая подключения к КЦС-IPM / КУН-IP, рассматриваемому далее

Modbus - это коммуникационный протокол широко применяемый в промышленности. Он определяет правила пересылки данных при взаимодействии устройств.

Мы можем реализовать диспетчеризацию и управления практически любого устройства, если оно поддерживает данный протокол.

Есть несколько модификаций данного протокола:

- Modbus RTU
- Modbus TCP
- Modbus ASCII (в настоящее время не поддерживается в АСУД-248)

Само слово "Modbus" - ничего не говорит об интерфейсе между устройствами.

Протокол Modbus может работать поверх интерфейсов RS-485/RS-232, компьютерной сети и других.

С точки зрения доступа к данным устройство Modbus представляет собой таблицу, примерно следующего вида

Адрес регистра	Атрибуты	Параметр	Значение
1	R/W, Целое число	Уставка температуры	15
2	R, Целое число * 10	Температура наружного воздуха	-156
301	R, Дробное число	Частота сети	49,8

где,

- Адрес регистра - уникальный идентификатор параметра, по которому осуществляется запрос значения параметра
- Атрибуты - атрибуты параметра
 - ◆ R - параметр можно читать
 - ◆ W - параметр можно записывать
 - ◆ Целое число - параметр представляет собой целое число
 - ◆ Целое число * 10 - параметр следует разделить на 10 для получения реального значения
 - ◆ Дробное число - параметр представляет собой дробное число
- Параметр - описание параметра
- Значение - текущее значение параметра

Внимание!

Таблица должна быть отражена в технического документации конкретного устройства.
Таблица не универсальна для разного типа Modbus-устройств.
Без наличия таблицы настроить устройство не получится.

Для некоторых типов Modbus-устройств, например: свободно программируемых контроллеров, таблица может зависеть от проекта, зашитого в контроллер.

Т.е. может быть сформирована в штатном ПО контроллера только после создания файла-проекта (см. например [PIXEL](#)).

Все регистры Modbus-устройства доступные на чтение / запись условно хранятся в 4-х таблицах:

Таблица	Функция чтения	Тип данных	Разрешены Чтение (R) / Запись (W)	Функция записи
Регистры флагов (Coils)	0x1	0 или 1	R / W	0x5
Дискретные входы (Discrete Inputs)	0x2	0 или 1	R	-
Регистры хранения (Holding Registers)	0x3	числовой/строковый	R / W	0x10 или 0x6 (см. параметр AW далее)
Регистры ввода (Input Registers)	0x4	числовой/строковый	R	-

Каждый регистр:

- это ячейка размером в 2-байта.
- имеет уникальный адрес.

В зависимости от типа Modbus-устройства адресное пространство таблиц может быть объединено, т.е. регистр с адресом 0 в таблице HoldReg, аналогичен (полностью соответствует) адресу 0 в таблице InpReg (это можно понять либо из документации, либо при тестовом опросе устройства).

Если вы затрудняетесь с тем, какой тип регистра указывать - используйте HoldReg.

Иногда в документации на устройство можно увидеть, что адреса регистров обозначаются 6-ти значными числами: логическими адресами.

Старшая цифра логического адреса при этом обозначает тип регистра, а остальные цифры - физический адрес регистра.

Старшая цифра логического адреса	Начальный логический адрес	Тип данных
0	000001	Coil
1	100001	Disclnp
3	300001	InpReg
4	400001	HoldReg

Как правило, при таком указании логических адресов считается что базовый регистр устройства = 1 (т.е. из адреса еще нужно будет вычесть единицу)

Например: логический адрес = 40568

- Тип регистра 4 = HoldReg
- Физический адрес регистра = 568 - 1 = 567

Замечание!

При регистрации устройства единицу все время вычитать не нужно, вы просто указываете в настройках устройства значение базового регистра = 1.

Для определения значения, хранящегося в регистре, необходимо, исходя из документации, корректно указать тип данных.

Поддерживаются следующие типы данных:

Название	Альтернативное название	Число регистров	Диапазон значений
----------	-------------------------	-----------------	-------------------

BIT		1	0 или 1
WORD	Unsigned word	1	Целые числа 0 .. 65535
CHAR	Signed byte, Short Integer	1	Целые числа -128..127
SHORT	Signed word, Small Integer	1	Целые числа -32 768 ... 32 767
FLOAT		2	1.5x10 ⁻⁴⁵ ... 3.4x10 ³⁸
DWORD	Unsigned long	2	Целые числа 0..4294967295
INTEGER	Signed long	2	Целые числа -2147483648..2147483647
BIT4		2	0 или 1
DOUBLE		4	5.0 x 10 ⁻³²⁴ .. 1.7 x 10 ³⁰⁸
STRING		указывается (запрашиваемое число регистров будет равно половине указанному)	строка данных
INT64	Big Integer		Целые числа -2 ⁶³ ... 2 ⁶³ -1

1 регистр = 16 бит
2 регистра = 32 бита
4 регистра = 64 бита

Также важен формат представления числа в регистре. Он определяет порядок перестановки полу-байт в регистрах.

Обычное значение поля Формат:

- Для типов данных в 1 регистр - 1
 - Для типов данных в 2 регистра - 4 (или значение 2)
 - Для типов в 4 регистра - 14 (или значение 12)
- Для данного типа существуют дополнительные значения формата с 11 до 14:
- ♦ формат с 1 до 4 - дополнительно меняют местами левые-правые 2-регистра
 - ♦ формат с 11 до 14 - аналог 1 - 4, но не выполняется перестановка.

Если при успешном опросе устройства, вы получаете в регистрах непонятные значения, попробуйте поменять Формат.

Комментарий по полю Формат (при первом чтении данный раздел можно пропустить).

Рассмотрим пример влияния значения поля Формат на вычисленный результат (см. программу **KCS DEMO**).

Предположим мы считываем целочисленное значение типа INTEGER (целое число, занимающее 2 регистра = 4 байта данных), по начальному адресу, например: 2054 (0x0806). Пусть нам известно, что в этом регистре должно храниться значение = 0x12345678 (значение в 16-ричной кодировке или 305419896 в привычной нам 10-ричной кодировке, преобразование вы можете выполнить в калькуляторе Windows).

Запрос данных (в шестнадцатиричной кодировке):
>> 01 03 08 06 00 02 26 6A

Значения регистров передаются начиная с указанного адреса, по два байта на регистр, старший байт каждого регистра передаётся первым.

Ответ устройства
36 26 16 06
<< 01 03 04 12 34 56 78 81 07

где:

- 0x12 0x34 - первый регистр (адрес 2054); байты 3, 2
- 0x56 0x78 - второй регистр (адрес 2055); байты 1, 0

Полученные данные последовательность байт	Формат	Правило перестановки	Результат	Комментарий
12 34 56 78	1	1_0_3_2	0x56781234	
12 34 56 78	2	0_1_2_3	0x78563412	обратный порядок младший байт вперед
12 34 56 78	3	2_3_0_1	0x34127856	
12 34 56 78	4	3_2_1_0	0x12345678	прямой порядок старший байт вперед

Т.е. для регистров устройства данного типа корректное значение формата - 4.

Предположим мы считываем целочисленное типа WORD, которое хранится в 1-м регистре (2 байта данных) по адресу, например: 2. Пусть мы значение что значение - 0x08.

Запрос данных (в шестнадцатиричной кодировке):
>> 01 03 00 02 00 01 25 CA

Ответ устройства
16 06
<< 01 03 02 00 08 B9 82

Полученные данные последовательность байт **Формат** **Правило перестановки** **Результат**

00 08 1 1_0 0x0008
00 08 2 0_1 0x0800

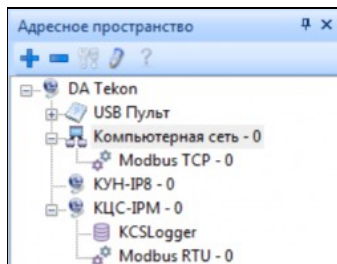
Т.е. для регистров устройства данного типа корректное значение формата - 1.

Добавьте Modbus-устройство в конфигурацию OPC-сервера.

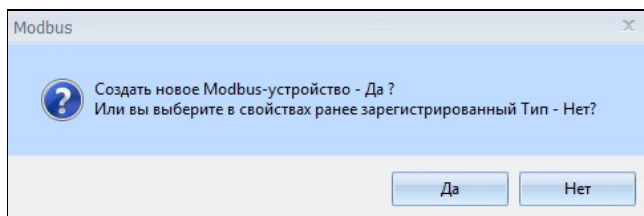
Если Modbus-устройство было зарегистрировано в программе ASUDBase, вы можете создать конфигурационный файл сервера, нажав в окне настройки Modbus-устройства в ASUDBase кнопку Экспорт.

Если устройство подключено

- напрямую по компьютерной сети
 - ◆ Добавьте DA \ Компьютерная сеть \ Modbus TCP
- через KЦC-IPM \ КУН-IPM
 - ◆ Добавьте DA \ Концентратор KЦC-IPM или КУН-IPM \ Modbus RTU
- через сторонний конвертер интерфейса
 - ◆ Добавьте DA \ IP-RS преобразователь \ Modbus RTU



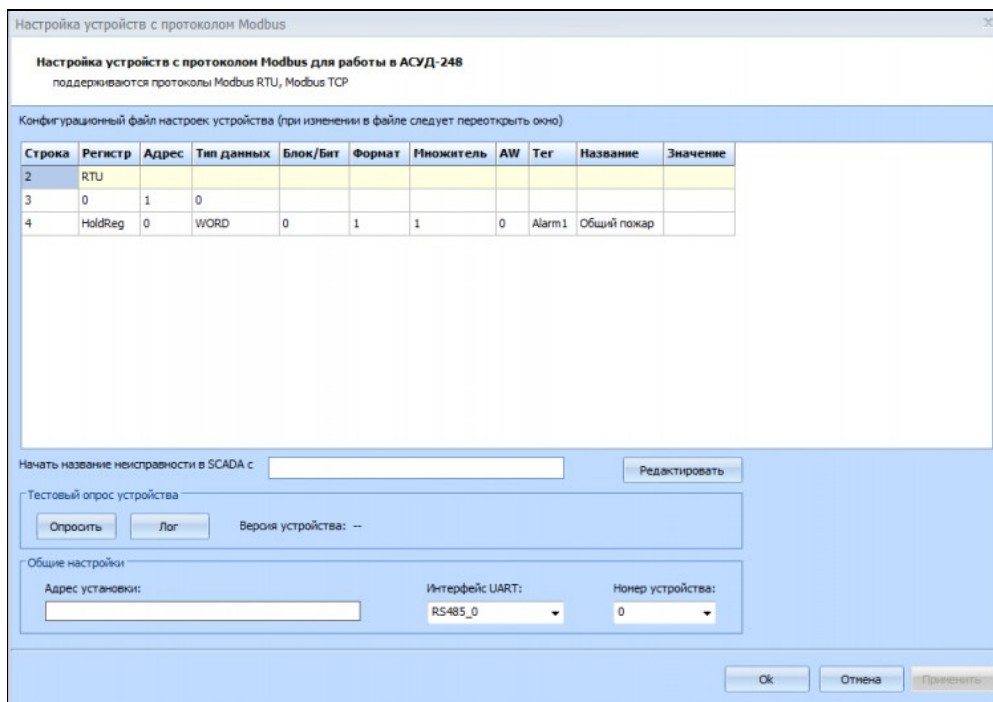
Выполните двойной клик на созданном устройстве.



Нажмите Да, если вы создаете новое устройство с новым описанием регистров. Нажмите Нет - если вы создаете устройство, для которого ранее уже создавали таблицу регистров.

Если вы выбрали Да - следует указать в латинице имя устройства, например: pixel

Далее выберите в свойствах созданный ранее конфигурационный файл устройства и нажмите два раза на само устройство, для того, чтобы открыть окно настройки.



На рисунке отображен пример окна настройки устройства, подключенного к концентратору KЦC-IPM, КУН-IPM.

Для каждой ячейки доступна информационная подсказка (хинт).

Строки имеют следующий смысл:

1. название столбцов таблицы
2. тип протокола устройства [RTU, TCP]
Если вы подключаете устройство по RS-485, то почти всегда это RTU-протокол. Если по компьютерной сети - TCP.
3. поля конфигурации устройства
 1. 0 - базовый регистр, указывает сдвиг регистров при запросе (обычно 0 или 1)
 2. 1 - количество регистров читаемых в одном запросе, по умолчанию 1.
Большее значение параметра увеличивает скорость опроса устройства.
Но имеет смысл, если запрашиваемые регистры идут подряд.
Если в документации не указано сколько можно читать - попробуйте указать 10 и провести тестовой опрос устройства.
Если ошибок нет - оставляйте указанное значени. Иначе верните значение 1.
 3. 0 - не разрешать пропуски при чтении. Допустимые значения [0,1]
Некоторые устройства позволяют опрашивать несколько регистров, если они расположены не подряд.
Это также увеличивает скорость опроса устройства.
4. следующие строки - это собственно таблица адресов Modbus-регистров из документации на устройство

Для редактирования таблицы следует нажать Редактировать.

При этом по правому клику мышки, доступно меню добавления новых строк.

Описание столбцов настройки:

- Регистр - тип регистра из выпадающего меню
- Адрес - адрес регистра (в десятичной системе счисления)
Вы можете указывать адрес в 16-ричной кодировке, для этого добавьте в начале знак \$. Например: \$10, \$FA и т.п.
- Тип данных - тип указанный в документации на устройство
- Блок/Бит
 - ◆ для типа данных BIT указывается анализируемый бит (от 1 до 16) в слове. Бит 1 - младший, 16 - старший.
 - ◆ для типа данных BIT4 указывается анализируемый бит (от 1 до 32) в слове. Бит 1 - младший, 32 - старший.
 - ◆ для типа STRING - размер блока данных в байтах (обычно размер строки в документации на устройство указывается кратным двум байтам).
 - ◆ для других типов данных - не используется.
- Формат - формат данных (подробнее см. ранее)
- Множитель - множитель данных, по умолчанию 1.

Множитель учитывается для всех типов кроме BIT, BIT4, STRING

После чтения регистра:

Значение SCADA = значение регистра * множитель

Перед записью регистра:

Значение регистра = Значение SCADA / множитель

Минимальное значение Множителя 0,0001.

Внимание!

При указании в качестве Множителя значений 2, 3, 4, 5, 6, 7, 8, 9 - реальное значение множителя определяется по формуле

Множитель = 1 / (2 ^ Указанный_множитель)

- AW - данный параметр разрешает запись значений в регистр, например для изменения уставок устройства или управления устройством из SCADA.
Актуально **только** для регистров типа Coils / HoldReg.
Возможные значения:
0 - запись в регистр запрещена
1 - запись в регистр разрешена. Для регистров HoldReg будет использоваться функция \$10 для записи
6 - для регистров HoldReg будет использоваться функция \$06 для записи
10 - аналог значения 1. Для регистров HoldReg будет использоваться функция \$10 для записи
- Тега - имя тега, которое будет отображено в SCADA.
Имя тега должно быть уникально в рамках устройства.
В имени Тега не следует использовать символ точка.
- Название - наименование сигнала (неисправности), которое будет отображаться в журнале событий в программе SCADA.

Для сохранения изменений, нажать кнопку Сохранить.

Замечание!

Поле Версия устройства - не используется (всегда отображает ---)

Для выполнения тестового опроса устройства, сохраните настройки и нажмите кнопку Опросить. Будет выполнен опрос устройства с отображением полученных данных в столбце Значение

Кнопка Лог - отобразит лог последнего обмена данными с устройством. Расшировку кодов возможных ошибок см. в описании программы [KCS DEMO.EXE](#)

Вы также можете выполнить тестовый опрос одного регистра, нажав правой кнопкой мыши и выбрав меню Опросить.

Если вы указали для регистра значение AW > 0, вы можете выполнить тестую запись непосредственно в окне настройки.

Для этого в рабочем режиме выберите строку и нажмите правой кнопкой мыши, выбрав меню Записать значение.

Для настроенных Modbus-регистров OPC-сервер создает следующие типы сигналов, которые будут доступны в SCADA:

- Дискретный сигнал - если регистр имеет тип BIT и AW = 0
 - ◆ в SCADA вы сможете использовать стандартные примитивы типа Дискретный сигнал или LED, Изображение
- Канал управления - если для регистра установлен параметр AW = 1, и название Тега оканчивается на _CTRL, например "Svet_CTRL"

- ◆ в SCADA вы сможете использовать стандартные примитивы типа Канал управления или Изображение
- ◆ при этом подразумевается, что значение регистра может быть только 0 или 1. В противном случае не следует использовать данный тип сигнала
- Числовое значение - для всех остальных случаев

Если регистр имеет тип BIT или BIT4, но вы не хотите, чтобы сигнал был представлен в SCADA, как дискретный датчик, вам следует:

- изменить тип с BIT на Word - если регистр может иметь только значения 0,1 (и не представляет собой битовую маску)
- изменить тип с BIT4 на DWord - если регистр может иметь только значения 0,1 (и не представляет собой битовую маску)
- в противном случае добавить к полю сокращение "_\$", т.е. например: вместо "Tag1" указать "Tag1_\$"

Если, исходя из описания устройства, регистр представляет собой битовую маску ошибок, как его регистрировать?

Например:

- 520.01 - авария насоса
- 520.02 - неисправность датчика температуры
- 520.03 - перепад давления

Запись типа 520.01 - означает, что для получения сигнала "Авария насоса" следует прочитать регистр 520 и проверить факт установки в значении регистра 1-го (младшего бита).

Т.е. в значении регистра 520 хранится несколько сигнальных (аварийных) событий. Допустим вы читаете регистр и получаете значение = 5.

Переводим это значение в двоичный код (например, с помощью Калькулятора Windows) и получаем значение 5 (DEC) = 101 (BIN), т.е. в регистре 520 установлены 1 и 3 бита, есть авария насоса и перепад давления.

Исходя из этого, следует определить, нужны ли в SCADA все ошибки по отдельности (отдельными примитивами типа Дискретный датчик) или только сигнал обобщенной ошибки с текстовым описанием активных ошибок?

Если все по отдельности - то для каждой ошибки делаете отдельную запись в таблице регистров:

- Адрес - одинаков
- Тип - BIT, если битовая маска лежит в одном Modbus-регистре или BIT4 - если в нескольких
- Блок/Бит - 1, 2, 3, и т.д. - тот бит, который отвечает за конкретную ошибку
- Название - Название ошибки
- В SCADA на каждую ошибку будет отображен свой Дискретный примитив

Если достаточно только обобщенной ошибки (т.е. факта того, что есть какая-то ошибка), то регистрируете только одну запись:

- Адрес - адрес битовой маски
- Тип - WORD или DWORD
- В SCADA используйте (подробнее см. ниже)
 - ◆ Переменную для отображения Аварийного сигнала обобщенной аварии или примитив LED-индикатор
 - ◆ Текстовое поле + Text-функцию для отображения описания активных ошибок

В АСУД.СCADA для визуализации данных Modbus-устройств можно использовать следующие примитивы:

- **Таблица Тегов - быстрая визуализация** в табличном виде в одном месте ситуационного плана без визуально-звуковой индикации:
 - ◆ числовых данных
 - ◆ состояний работы устройства (преобразование числа в надпись с помощью **Text-функций**)
 - ◆ расшифровка кодов ошибок (преобразование числа в надпись с помощью **Text-функций**)
- Цифровой индикатор, **стрелочный индикатор** - для отображения любых числовых значений (с возможностью указания граничных значений и визуально-звуковой индикации выхода за границы)
- Текстовое поле - для отображения любых числовых значений, а также преобразования числового значения в текстовое по заданному правилу с помощью **скрипта** или **Text-функции**. Кроме того с помощью данного примитива и **скрипта** можно выполнить **запись значения в регистр** Modbus-устройства.
- **Кнопка** - для **записи значения в регистр**.
- Дискретный датчик - визуализация аварийных событий.
- **Led-индикатор** - для отображения дискретных значений, в том числе выделение бита из маски.
- **Прибор учета** - если устройство было зарегистрировано в ASUDBase и вам нужна простая визуализация в табличной форме
- **Изображение** - визуализация аварийных событий
- **Набор изображений** - визуализация состояния устройства или **запись значения в регистр** Modbus-устройства.

Замечание! Для того чтобы реализовать возможность записи введенного пользователем значения в Modbus-устройство, необходимо чтобы:

- Регистр относится к типу **HoldReg** или **Coils**
- Значение поля **AW** (Allow Write) было > 0

Рассмотрим пример визуализации значения регистра, представляющего собой битовую маску ошибок, если этот регистр был настроен в OPC-сервере, как числовое значение (а не набор типов BIT).

Пример 1.

Необходимо визуализировать с помощью Дискретных примитивов состояние регистра "Авария", где значение регистра представляет собой битовую маску кодов аварий

Например, биты:

- 1 - авария насоса
- 2 - неисправность датчика температуры
- 3 - перепад давления

Создаем несколько переменных (по числу аварийных событий), например:

Var.1:

- Имя - 1
- Журнал событие - Авария насоса
- Журнал адрес - Котельная
- Задержка (с.) - 0
- Тип - ВЫРАЖЕНИЕ
- Выражение - BIT (X;1)
- Добавляем Тег регистра "Авария"

Var.2:

- Имя - 2
- Журнал событие - Неисправность ДТ
- Журнал адрес - Котельная
- Задержка (с.) - 0
- Тип - ВЫРАЖЕНИЕ
- Выражение - BIT (X;2)
- Добавляем Тег регистра "Авария"

Var.3:

- Имя - 3
- Журнал событие - Перепад давления
- Журнал адрес - Котельная
- Задержка (с.) - 0
- Тип - ВЫРАЖЕНИЕ
- Выражение - BIT (X;3)
- Добавляем Тег регистра "Авария"

Размещаем на ситуационном плане примитивы типа Дискретный датчик, в настройке Тег вручную указываем имена переменных VAR.1, VAR.2, VAR.3.

Пример 2. Необходимо визуализировать с помощью Дискретного датчика обобщенную аварию регистра "Авария" из предыдущего примера.

Иногда нет необходимости выделять каждую Аварии отдельным примитивом Дискретный датчиком, а например достаточно просто показать сигнал Обобщенной аварии. В этом случае,

Создаем переменную VAR.Total:

- Имя - Total
- Журнал событие - Авария
- Журнал адрес - Котельная
- Задержка (с.) - 0
- Тип - <> 0
- Добавляем Тег регистра "Авария"

Размещаем на ситуационном плане примитив типа Дискретный датчик, в настройке Тег вручную указываем VAR.Total.

Если значение тега "Авария" будет отлично от 0 (т.е. будет установлен флаг любой из аварий), то примитив Дискретный датчик перейдет в сигнальное состояние.

При этом расшифровку Аварий, удобно реализовать с помощью примитивов [Таблица Тегов](#) или Текстовое поле, связав их с тегом "Авария" (не переменной, а тегом!), и создав [Текст-функцию](#) (см. пример 3 Функция BITSTR в описании)

```
# BITSTR()
0=Норма
1=авария насоса
2=неисправность датчика температуры
3=перепад давления
```

Вы можете сами попробовать настроить в SCADA Modbus-устройство воспользовавшись эмулятором в программе [KUNIP.EXE](#) См. пример настройки в описании эмулятора.