

Руководство пользователя драйвера MEM



Содержание

Введение	3
1. Последовательность шагов работы с драйвером MEM	4
2. Расположение и проверка драйвера MEM	5
3. Настройка драйвера MEM в конфигураторе	6
3.1 Описание конфигурационного файла	6
3.2 Создание конфигурации	11
3.3 Открытие конфигурации	14
3.4 Запуск конфигуратора	15
3.5 Опция Перечитать конфигурацию	16
3.6 Опции «Подписать тег» и «Записать значение»	17
4. Включение и выключение драйвера MEM	20
4.1 Включение драйвера MEM	20
4.2 Выключение драйвера MEM	21
5. Настройка сигналов	22
5.1 Создание нового сигнала	22
5.2 Описание формата источника тегов драйвера MEM	23
6. Запуск ввода-вывода драйвера MEM	29
7. Перечень поддерживаемых драйвером типов данных	30
8. Описание статистики	31
9. Логирование	33
10. Описание спецификации расширенных функций Emicon 67, 70, 99	35
11. Порядок байт в данных, полученных драйвером	39
Приложение А	41
Лист регистрации изменений	42

Введение

Руководство пользователя для драйвера MEM содержит инструкции и справочную информацию по **использованию драйвера MEM** в программной платформе SCADA-системы «К-Систем» (далее по тексту - K-System SCADA) и предназначено **для технических специалистов**, занимающихся внедрением и настройкой K-System SCADA.

Драйвер MEM (Modbus Ethernet Master) - это драйвер ввода/вывода предназначенный для работы K-System SCADA по протоколу Modbus TCP.

Драйвер является частью K-System SCADA и не требует отдельной установки. Для работы с драйвером необходима установленная система K-System SCADA.



Для дополнительной информации по установке K-System SCADA рекомендуется обратиться к документу «Программная платформа SCADA-система «К-Систем». Руководство системного администратора».

1. Последовательность шагов работы с драйвером MEM

1. [Включить драйвер MEM.](#)
2. Сохранить и перезапустить проект.
3. [Запустить систему ввода - вывода.](#)
4. [Открыть конфигуратор.](#)
5. [Создать конфигурацию.](#)
6. [Добавить шлюз \(главное устройство\).](#)
7. [Добавить ведомое устройство.](#)
8. [Добавить блок данных.](#)
9. Сохранить конфигурацию.
10. Убедится в расположении файла конфигурации в свойствах драйвера MEM в полях CfgFileName и Dir.
11. [Открыть конфигурационный файл через блокнот.](#)
12. Заполнить конфигурационный файл.
13. Сохранить конфигурационный файл.
14. [Прописать Source у тегов драйвера MEM.](#)
15. [Перечитать конфигурацию.](#)
16. Возврат в конфигуратор.
17. [Запуск работы конфигуратора.](#)

**Совет!**

Опция *Перечитать конфигурацию* необходима при изменении конфигурационного файла в процессе работы системы K-System SCADA.

2. Расположение и проверка драйвера MEM

Перед началом использования необходимо убедиться, что все файлы драйвера MEM расположены в требуемой директории.

Для работы драйвера MEM необходимы три файла:

- drvmem.dll (файл библиотеки);
- Default.mem (конфигурационный файл);
- DrvMEM.lib (файл дескриптор драйвера).



Внимание! Конфигурационный файл является основным способом редактирования и хранения конфигурации. Название конфигурационного файла может быть изменено, однако расширение конфигурационного файла для драйвера MEM - «**.mem**».

3. Настройка драйвера MEM в конфигураторе

3.1 Описание конфигурационного файла

Конфигурационный файл **Default.mem** см. рисунок 3.1, следует располагать в директории K-System SCADA по следующему пути: **<путь к K-System SCADA>\bin\Drivers**.

```
<?xml version="1.0" encoding="utf-8"?>
<MBMasterTCP ConfigVersion="1" DrvMEMVersion="1.3.1.15">
  <Gateways>
    <Gateway0
      Name=""
      Wait=""
      Host1="" |
      Host2=""
      TimeOut=""
      NumPort1=""
      NumPort2=""
      Host1Enabled=""
      Host2Enabled=""
      TimeDataChange=""
      CountConnection=""
      PauseBetweenMess=""
      ReconnectionTimeMs=""
      FirstConnectionTimeMs="">
    <SlaveDevice0
      Name=""
      Active=""
      Address=""
      DataBlockFailureTimeoutMs="">
    <Block0
      Kind=""
      Name=""
      Type=""
      Begin=""
      Count=""
      KindExt=""
      DataType=""
      Interval=""
      ByteOrder=""/>
    </SlaveDevice0>
  </Gateway0>
</Gateways>
</MBMasterTCP>
```

Рисунок 3.1 - Конфигурационный файл

Параметры шлюза (главного устройства)	
Name	Наименование контроллера
Wait	Время ожидания ответа от устройства, мс
Host 1	Основной интерфейс
Host 2	Резервный интерфейс
TimeOut	Timeout подключения к устройству, мс
NumPort1	Номер порта основного интерфейса
NumPort2	Номер порта резервного интерфейса
Host1Enabled	Активность основного интерфейса
Host2Enabled	Активность резервного интерфейса
TimeDataChange	Интервал срабатывания события DataChange, мс
CountConnection	Количество подключений
PauseBetweenMess	Пауза между запросами в каждом подключении, мс
ReconnectionTimeMs	Время с момента обрыва подключения, в течение которого предыдущий активный хост считается приоритетным
FirstConnectionTimeMs	Время с момента активации ввода-вывода, в течение которого основной хост считается приоритетным
Параметры ведомого устройства	
Name	Название ведомого устройства
Active	Активность ведомого устройства
Address	Адрес устройства в сети Modbus
DataBlockFailureTimeoutMs	Время на которое отключается ведомое устройство при неуспешном опросе любого блока, мс
Параметры блока данных	
Kind	Тип работы блока (читать и писать)
Name	Название блока
Type	Тип блока по стандарту ModBus (1,2,3,4)
Begin	Начальный адрес блока
Count	Количество регистров в блоке, задаётся в конфигураторе
KindExt	Тип протокола (0-Standard ModBus, 1-Emicon)
DataType	Тип данных блока
Interval	Периодичность обработки (опроса) блока, мс
ByteOrder	Порядок байт в полученном сообщении

Актуальное описание полей конфигурационного файла:

	Внимание! Структура конфигурационного файла была обновлена. В таблице ниже приведены устаревшие параметры.
---	---

Устаревший параметр	Актуальный параметр	Комментарий
Adress	Address	
Host1Enebled	Host1Enabled	
Host2Enebled	Host2Enabled	
PauseBetwinMess	PauseBetweenMess	
End		Избыточный параметр блока данных; убран из конфигурационного файла
ReversByte	ByteOrder	Поле ByteOrder принимает значение от 0 до 5 (см.раздел “Порядок байт в данных, полученных драйвером”)

В конфигурационном файле содержится свойство *DataType*, отвечающее за тип данных блока.

Данные блока в конфигураторе драйвера MEM выводятся в соответствии с размером регистра и типом данных блока. Например,(см. рисунок 3.2) для типа данных *Single (float)* отводятся два регистра (по стандарту Modbus), но само значение находится в одном занимаемом регистре. Поэтому конфигуратор показывает данные через один регистр.

drivers\test.mem

Name	Single	Название блока	
DataType	Single	Тип данных блока, задается в конфигурационном файле	
type	us Holding Registers (Регистры входа)	Тип блока по стандарту Modbus (3,4)	
Begin	400350	Начальный адрес блока	
Count	10	Кол-во регистров в блоке, задается в конфигурационном файле	
Kind	Читать и писать	Тип работы блока (0,1,2,3)	
KindExt	Стандартный Modbus		
Interval	500	Периодичность обработки (опроса) блока	
ByteOrder	LittleEndian (DC BA)	Порядок байт в полученном ответе	

TikMes	203	Количество запросов блока у устройства	
TikMesGet	203	Количество ответов от устройства	
TikMesErr	0	Количество ошибок или ошибочных ответов	
TikMesTimeOut	0	Количество срабатываний TimeOut	
Last time read	11:16:18.858	Время последнего чтения блока	
Quality	GOOD	Качество данных блока	
Address	**Word (uint16_t)**	**Byte (uint8_t)**	**SmallInt (**Single (float)**
0 : 400350 : 1		1111111101111111	-3.296485227E38
1 : 400351		1111111111111111	
2 : 400352 : 2		0111111101111111	3.402823466E38
3 : 400353		1111111111111111	
4 : 400354 : 3		0011101011010100	0.001617431641
5 : 400355		0000000000000000	
6 : 400356 : 4		0111111101111111	3.402823466E38
7 : 400357		1111111111111111	

Рисунок 3.2 - Конфигуратор (пример отображения типа данных «Single»)

Для каждого блока отображаются абсолютный (по Modbus) и относительный (внутри блока) регистры, см. рисунок 3.3.

Last time read	11:07:46.904	Время последнего чтения блока
Quality	GOOD	Качество данных блока
Address	**Word (uint16_t)**	**Byte (uint8_t)**
0 : 400100 : 1		0000000000000001
1 : 400101		0000000000000001
2 : 400102 : 2		0000000000000101
3 : 400103		0000000000000000
4 : 400104 : 3		0000000000000111
5 : 400105		0000000000000000
6 : 400106 : 4		0000000000010000
7 : 400107		0000000000000000
8 : 400108 : 5		0000000000010000
9 : 400109		0000000000000000
10 : 400110 : 6		0001000100010001
11 : 400111		1111111111111111
12 : 400112 : 7		0000000000000000
13 : 400113		0000000000000000
14 : 400114 : 8		1100110011001100
15 : 400115		1011101110111011
16 : 400116 : 9		1010101010101010
17 : 400117		1101110111011101
18 : 400118 : 10		1110111011101110
19 : 400119		0001111111111111

Рисунок 3.3 - Конфигуратор (пример отображения поля «Address»)

Поле *DataType* в конфигурационном файле может принимать следующие значения:

Тип данных	Значения в конфигурационном файле драйвера MEM
Smallint	varsmallint, smallint, 3
Integer	varinteger, integer, 4
Single	varsingle, single, 5
Double	vardouble, double, 6
Currency	Varcurrency, currency, 7
Date	vardate, date, 11
Boolean	varboolean, boolean, 8
Byte	varbyte, byte, 18
Word	varword, word, 19
LongWord	varlongword, longword, 20
Int64	varint64, int64, 20
ShortInt	varshortint, shortint, 16

В зависимости от выбранного типа блока данных (*Type*) задается смещение адресации (по стандарту Modbus), см. ниже таблицу соответствия диапазонов адресов регистров и функций согласно официальному стандарту Modbus.

Тип регистра/ Описание	Функция	Код функции Modbus	Смещение адресации (диапазон адресов)
Coils Дискретные выходы, битовые значения, чтение	01 (Coil Status)	1	0
Discrete Inputs Дискретные входы (только чтение)	02 (Input Status)	2	100001
Holding Registers Регистры хранения (16-бит, чтение/запись)	04 (Holding Register)	3	400001
Input Registers Входные регистры (16-бит, только чтение)	03 (Input Registers)	4	300001

Драйвер MEM поддерживает 3 и 4 тип блоков регистра, см. таблицу ниже. Это означает, корректную работу драйвера с функциями чтения/записи регистров:

- Чтение Input Registers — функция 4;
- Чтение Holding Registers — функция 3;
- Запись Holding Registers — функции 6 (один регистр) и 16 (несколько регистров).

			№ Функции:		3	4	6	16
Тип регистра :	Функция:	Номер регистра DEC	Размер	Доступ				
3	Регистры ввода (Input Registers)	300001- 365535	16-битное слово	только чтение		Read Multiple		
4	Регистры хранения (Holding Registers)	400001- 465535	16-битное слово	чтение и запись	Read Multiple		Write	Write Multiple

В драйвере MEM адреса регистров приводятся в шестизначном формате, несмотря на классический Modbus, где адреса регистров чаще приводятся в пятизначном формате. Шестизначный формат используется для расширения нумерации, чтобы иметь большой адресный диапазон.

	Внимание! Драйвер MEM поддерживает регистры только в шестизначном формате! Поддержка пятизначных регистров в разработке.
---	--

В конфигураторе драйвера MEM отображаются типы данных Pascal, и их соответствие в C++, см. таблицу ниже:

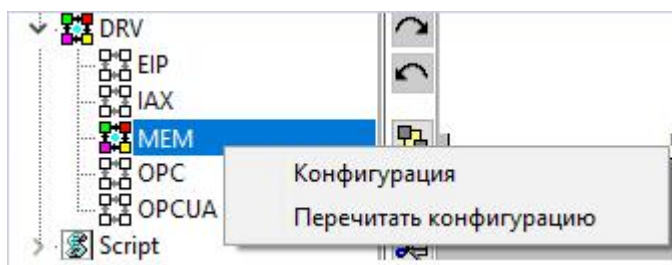
Тип данных Pascal	Тип данных C++	Описание
Word	uint16_t	Беззнаковое целое число (16 бит)
Byte	uint8_t	Беззнаковое целое число (8 бит)
SmallInt	int16_t	Знаковое целое число (16 бит)
Single	float	Число с плавающей точкой (32 бита)
LongWord / Cardinal	uint32_t	Беззнаковое целое число (32 бита)
LongInt	int32_t	Знаковое целое число (32 бита)
Double / Real	double	Число с плавающей точкой (64 бита)
ShortInt	int8_t	Знаковое целое число (8 бит)
Int64	int64_t	Знаковое целое число (64 бита)
Boolean	bool	Логический тип (true/false)

	Внимание! При смене типа данных тега в SCADA-системе необходимо перезапустить проект.
---	--

Драйвер MEM работает с данными, размер которых больше 16 бит, т.е. минимальная длина регистра для адресации равна 16 бит (WORD).

3.2 Создание конфигурации

Для того чтобы создать конфигурационный файл необходимо запустить **Конфигуратор**, нажав ПКМ в дереве объектов на объект *MEM* во вкладке *DRV*, как представлено на рисунке ниже.



В открывшемся окне «Конфигуратор» добавляем устройство, нажав на кнопку «+», см. рисунок 3.4.

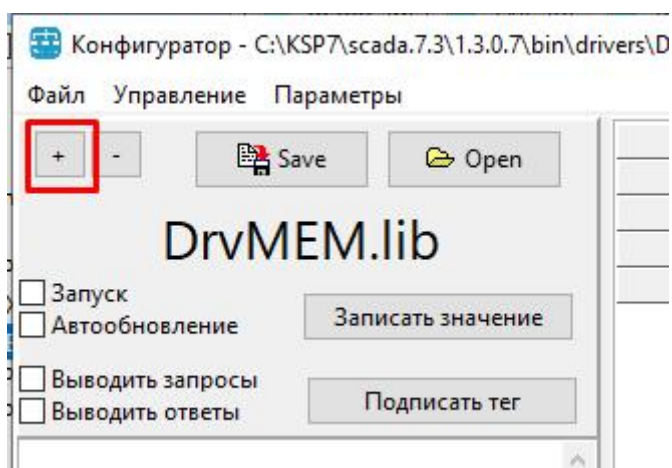


Рисунок 3.4 - Конфигуратор (Добавление шлюза (главного устройства))

Далее необходимо добавить ведомое устройство. Для этого нажмите ПКМ на добавленный шлюз и выберите «Добавить», см. рисунок 3.5.

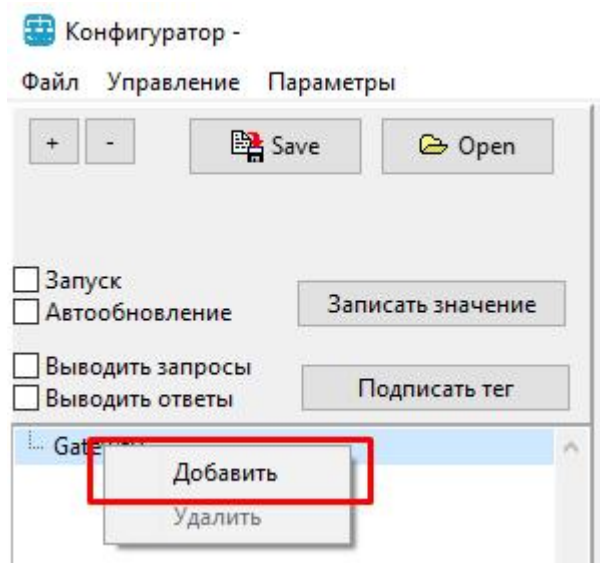


Рисунок 3.5 - Конфигуратор (Добавление ведомого устройства)

Далее необходимо добавить блок данных. Для этого нажмите ПКМ на добавленное ведомое устройство и выберите «Добавить», см. рисунок 3.6.

Данные действия после сохранения (нажатием на кнопку ) создают пустой файл конфигурации, содержащий только структуру.

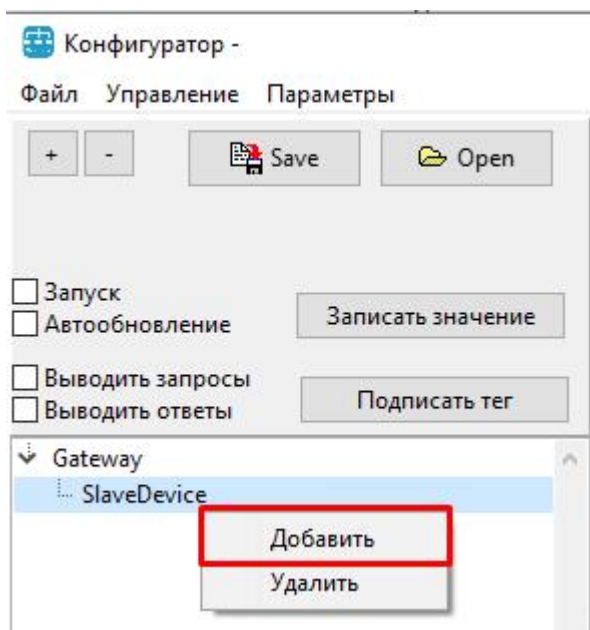


Рисунок 3.6 - Конфигуратор (Добавление блока данных)

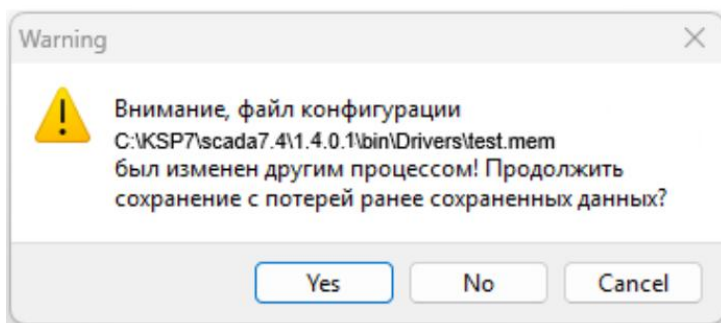
	Внимание! Необходимо сохранить созданный пустой конфигурационный файл в директорию K-System SCADA по следующему пути: <путь к K-System SCADA>\bin\Drivers\.
---	--

После сохранения файла, его дальнейшее редактирование возможно в ручном режиме в текстовом редакторе «Блокнот».

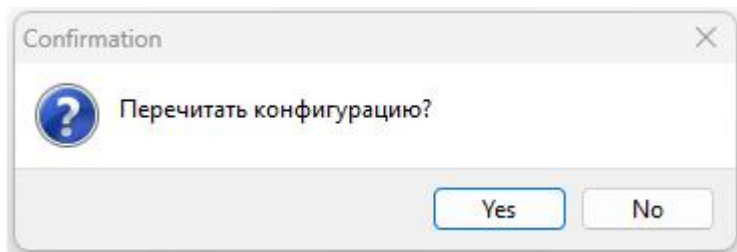
Так как возможен запуск нескольких экземпляров драйвера MEM, при многократном запуске конфигуратора, процесс сохранения файла конфигурации осуществляется согласно следующему сценарию.

Поведение при сохранении конфигурации в измененный другим процессом файл:

- При нажатии «Yes» файл конфигурации перезаписывается.
- При нажатии «Cancel» сохранение файла конфигурации отменяется.



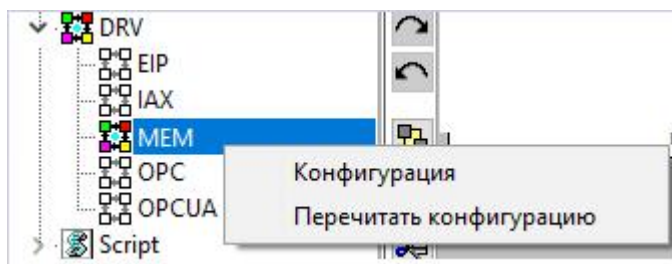
Поведение при нажатии кнопки «No»:



- При нажатии «Yes» конфигурация перечитывается.
- При нажатии «No» перечитывание конфигурации и сохранение файла конфигурации отменяется.

3.3 Открытие конфигурации

Для того чтобы открыть конфигурацию необходимо запустить **Конфигуратор**, нажав ПКМ в дереве объектов на объект *MEM* во вкладке *DRV*, как представлено на рисунке ниже.



	Внимание! При соблюдении расположения всех файлов драйвера <i>MEM</i> в одной папке по следующему пути: <путь к K-System SCADA>\bin\Drivers\., Конфигуратор сразу откроет конфигурацию, находящуюся по указанному пути.
---	---

Иначе, в открывшемся окне **«Конфигуратор»** выбрать пункт меню *Файл - Открыть конфигурацию* (см. рисунок 3.7) в открывшейся директории выбрать

требуемый файл с расширением *.mem или воспользоваться кнопкой *Open* (см. рисунок 3.8).

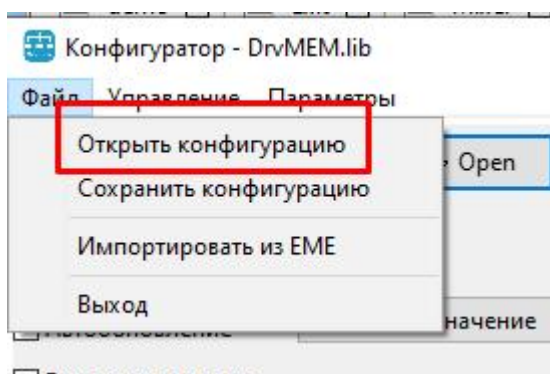


Рисунок 3.7 - Открытие конфигурации через меню - *Файл*

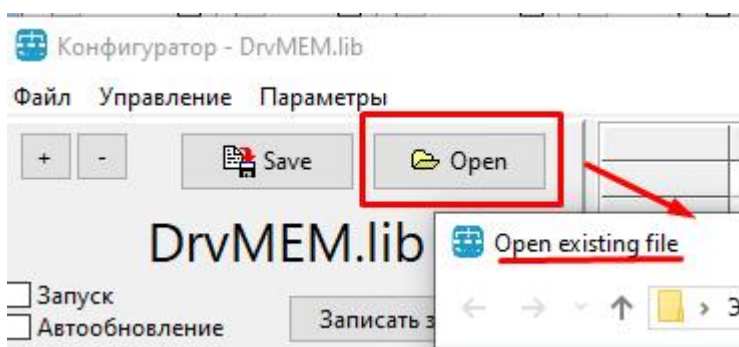


Рисунок 3.8 - Открытие конфигурации через кнопку - *Open*

3.4 Запуск конфигуратора

После открытия конфигурации необходимо нажать на чекбоксы *Запуск* и *Автообновление*, см. рисунок 3.9.

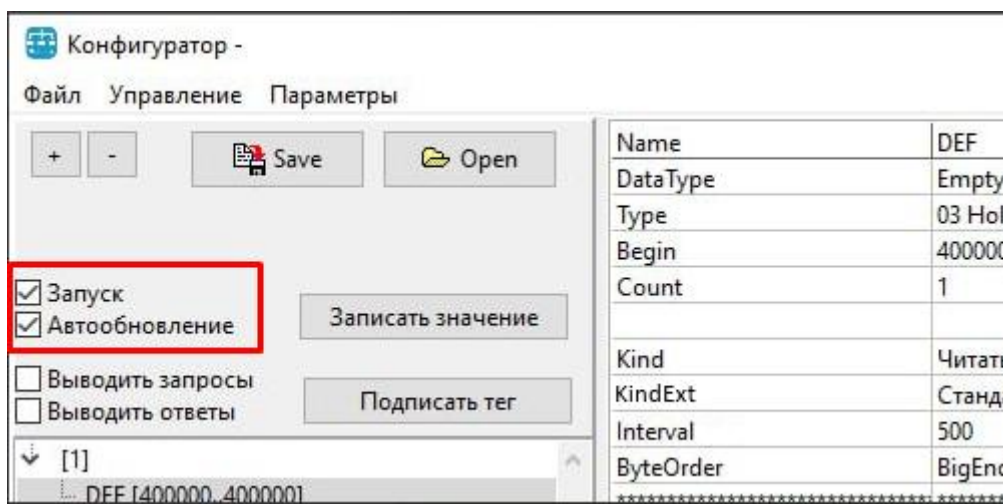


Рисунок 3.9 - Окно конфигуратора

Для вывода запросов к устройству и ответов от устройства, информации о подключённом хосте, ошибках подключения в поле логов конфигуратора, нажмите на чекбоксы *Выводить запросы* и *Выводить ответы*, см. рисунок 3.10.

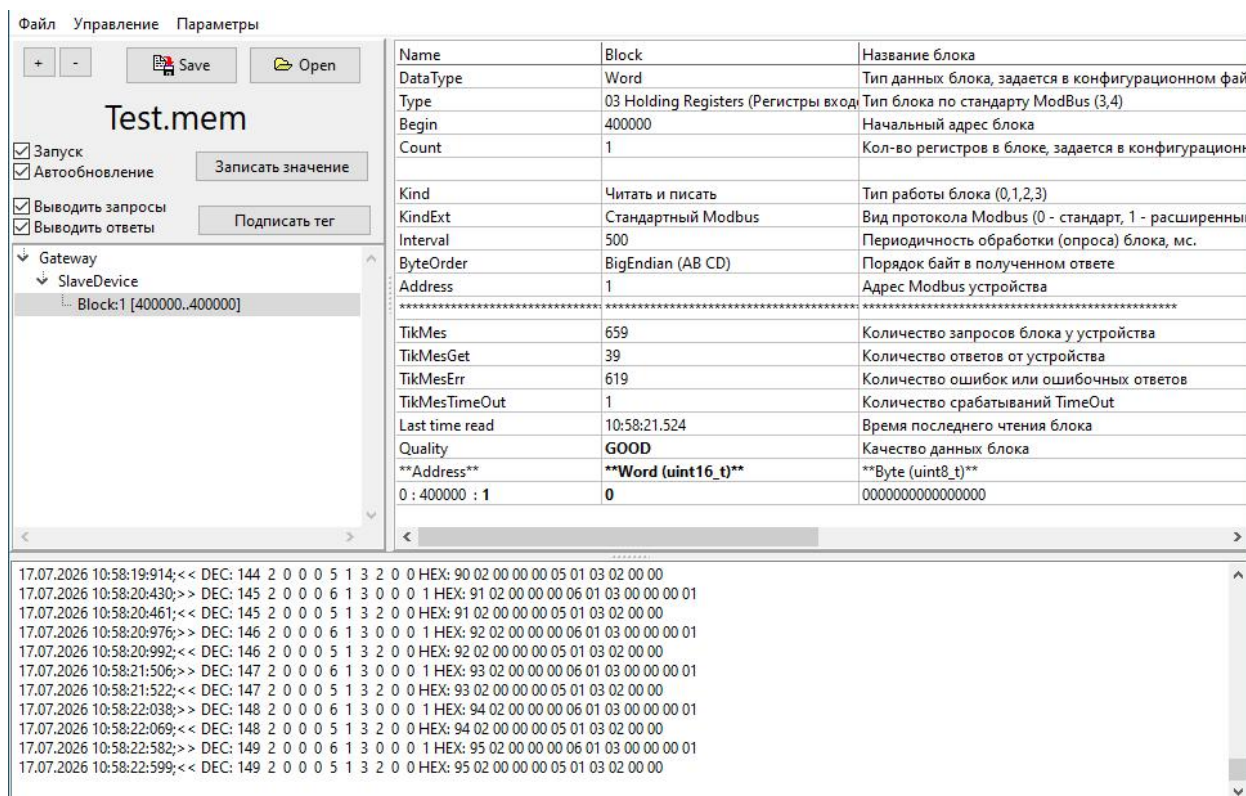


Рисунок 3.10 - Информационное поле окна конфигуратора

3.5 Опция Перечитать конфигурацию

После внесения изменений в конфигурационный файл в Блокноте, существует возможность в SCADA перечитать конфигурацию без сохранения и перезагрузки проекта.

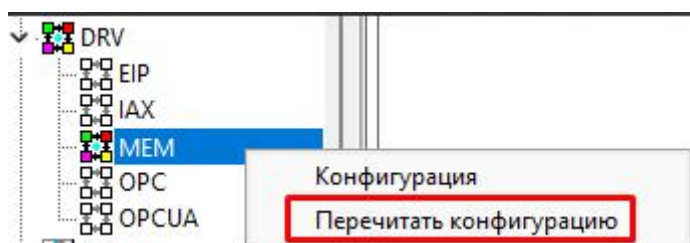


Рисунок 3.11 – Опция драйвера «Перечитать конфигурацию»

Опцию драйвера **Перечитать конфигурацию** возможно использовать без остановки ввода-вывода. Внести изменения в конфигурационный файл, сохранить его и закрыть. После редактирования конфигурационного файла через Блокнот,

необходимо заново открыть его в конфигураторе. Галочки «Запуск» и «Автообновление» можно оставить включенными — они не влияют на процедуру замены файла и не требуют снятия или повторной установки. Далее воспользоваться опцией **«Перечитать конфигурацию»**, см. рисунок 3.11.



Внимание! Для корректной работы опции **«Перечитать конфигурацию»** в свойствах драйвера MEM необходимо указать полный путь до файла конфигурации.

FileName	DrvMEM.dll
CfgFileName	C:\<путь к K-System SCADA>\bin\Drivers\тест мем.mem
Author	Sasha K
ShortName	MEM
Dir	C:\<путь к K-System SCADA>\bin\Drivers\
Enabled	true

3.6 Опции «Подписать тег» и «Записать значение»

Опции **«Подписать тег»** и **«Записать значение»** используются для проверки возможности записи в регистры тегов.

Инструкция по использованию:

1. Запустите конфигуратор MEM драйвера без открытия приложения SCADA.
2. Откройте конфигурацию, загруженную на ПЛК.
3. Запустите драйвер MEM в конфигураторе.
4. После подключения к ПЛК выберите нужный тег двойным щелчком ЛКМ.
5. Выполните подписку выбранного тега по кнопке **«Подписать тег»**, см. рисунок 3.12 .
6. Используйте кнопку **«Записать значение»** для записи тестового значения в выбранный тег, см. рисунок 3.14.
7. Проверьте, успешно ли значение записалось в регистр.

Данный процесс позволяет убедиться в корректности записи данных в теги ПЛК без запуска SCADA-приложения.

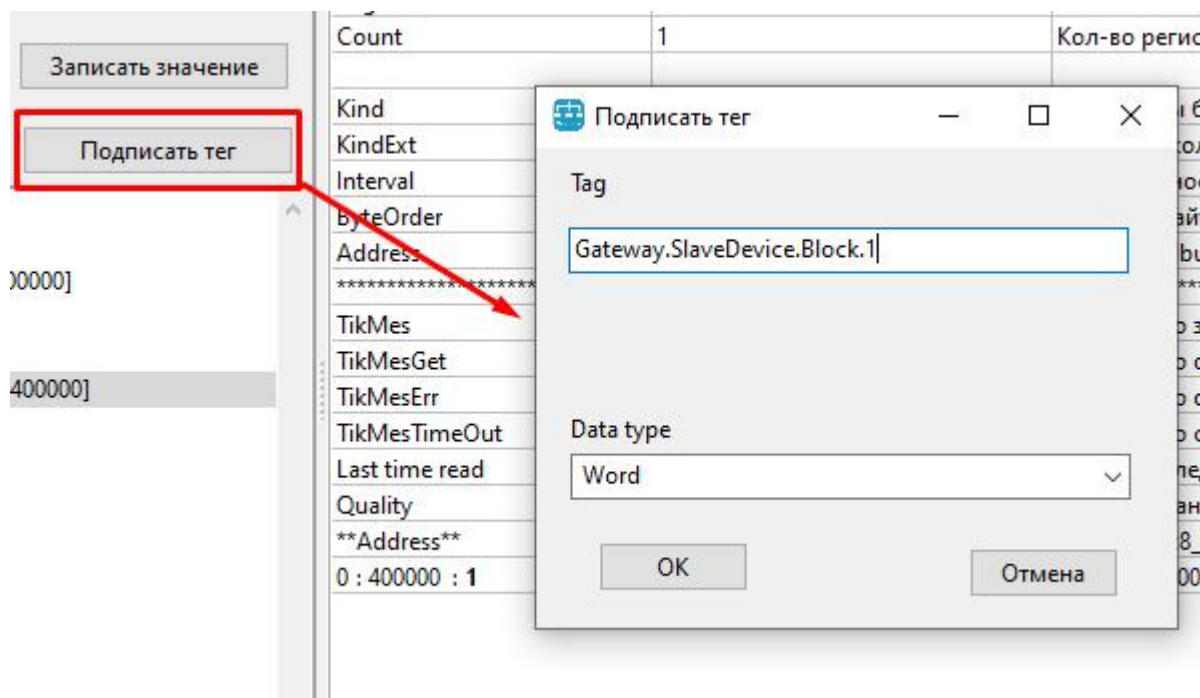


Рисунок 3.12 – Опция конфигулятора «Подписать тег»

При применении опции «**Подписать тег**» в информационном поле конфигулятора отображается запись, содержащая следующие данные: номер тега, качество подписки, время подписки и текущее значение тега, см. рисунок 3.13.



Рисунок 3.13 – Пример информационного поля конфигулятора – запись о подписке тега

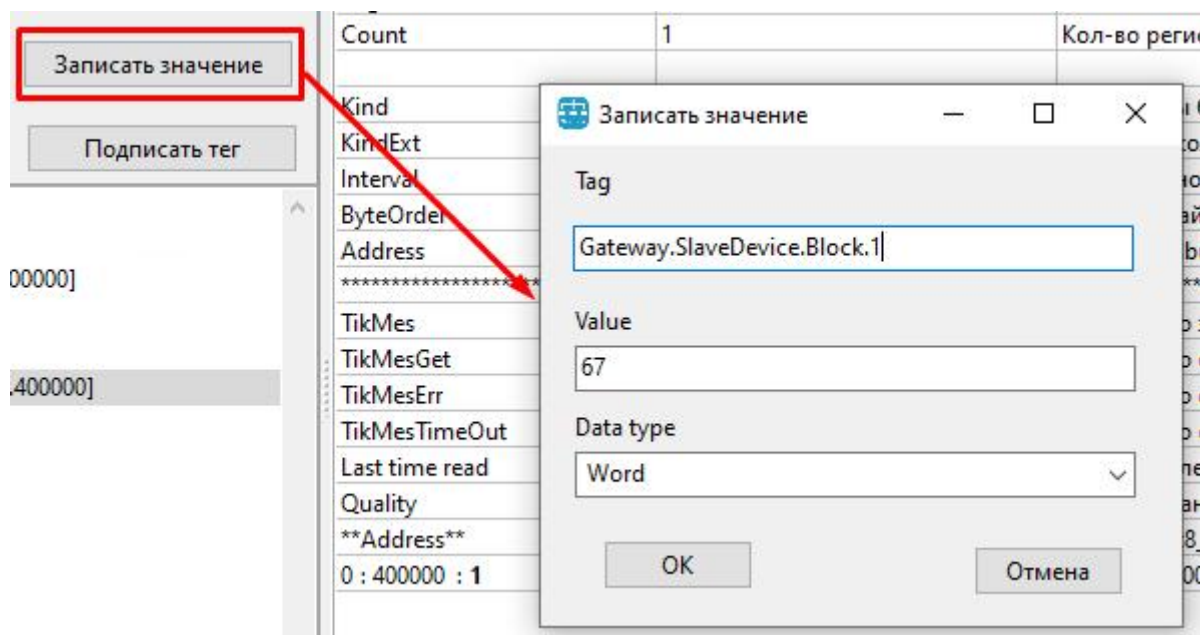


Рисунок 3.14 – Опция конфигулятора «Записать значение»

При применении опции «**Записать значение**» в информационном поле конфигулятора отображается запись, содержащая следующие данные: номер тега, качество подписки, время подписки и текущее записанное значение тега, см. рисунок 3.15.

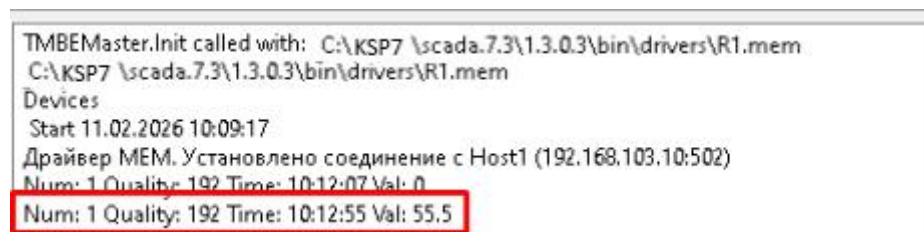


Рисунок 3.15 – Пример информационного поля конфигулятора – информация о записанном значении в теге

4. Включение и выключение драйвера MEM

После запуска K-System SCADA в режиме редактирования проекта и открытия проекта в дереве объектов на вкладке *DRV* отображается объект *MEM*.

По умолчанию драйвер выключен.

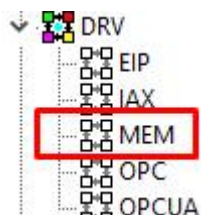


Рисунок 4.1 – Выключенный драйвер в дереве объектов

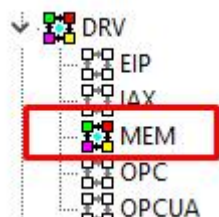


Рисунок 4.2 – Включенный драйвер в дереве объектов

4.1 Включение драйвера MEM

Включить драйвер можно двумя способами:

- Щелкнуть на объект *MEM*. В открывшихся **Параметрах объекта**, на вкладке *Свойства объекта* изменить параметр *Enabled* на *True*.
- Щелкнуть по вкладке **Инструменты** в строке меню. Перейти в *Проект* и в открывшемся окне *Параметры проекта* во вкладке **Драйверы** (см. рис. 4.3) изменить параметр драйвера MEM с *Выкл.* на *Вкл.*

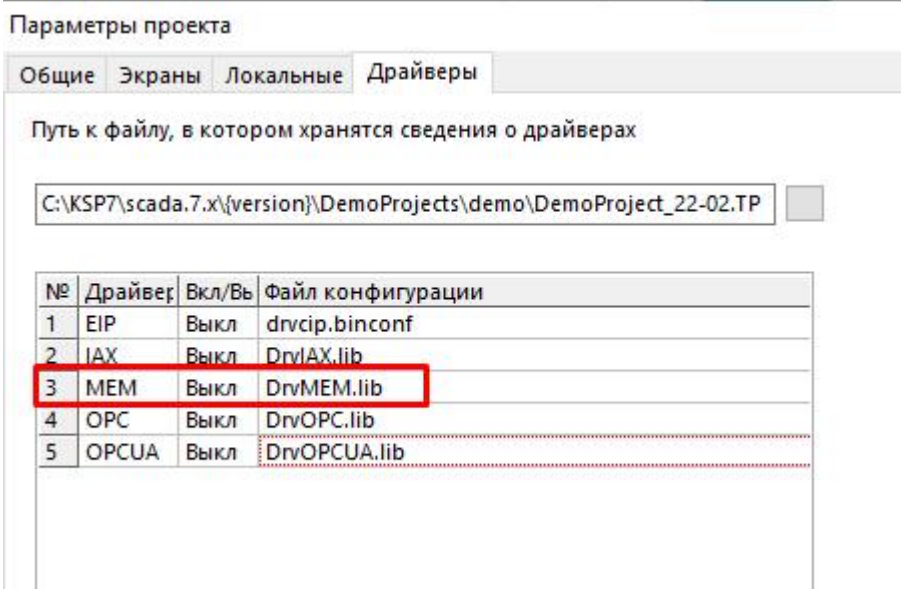


Рисунок 4.3 - Параметры драйвера

Сохранить изменения в проекте и перезапустить проект.

После перезапуска проекта, значок объекта *MEM* будет подсвечен , что соответствует запущенному драйверу MEM.

Убедиться в расположении файла конфигурации в свойствах драйвера *MEM* (см. рис. 4.4). В поле:

– *CfgFileName* - <путь к K-System SCADA>\bin\Drivers\<конфиг. файл>.

– *Dir* - <путь к K-System SCADA>\bin\Drivers\.

Свойства объекта	
MEM [TDLib] Множ. ☒	
Name	Driver Modbus TCP/IP
Tag	0
Desc	MEM
Pref	MEM
Path	Prj.DRV.MEM
IsLoad	True
IsInit	False
FileName	DrvMEM.dll
CfgFileName	C:\<путь к K-System SCADA>\bin\Drivers\тест мем.мем
Author	Sasha K
ShortName	MEM
Dir	C:\<путь к K-System SCADA>\bin\Drivers\
Enabled	True

Рисунок 4.4 - Свойства объекта MEM

4.2 Выключение драйвера MEM

Выключить драйвер можно двумя способами:

- Дважды щелкнуть на объект *MEM*. В открывшихся **Параметрах объекта**, на вкладке *Свойства объекта* изменить параметр *Enabled* на *False*.
- Щелкнуть по вкладке **Инструменты** в строке меню. Перейти в *Проект* и в открывшемся окне *Параметры проекта* во вкладке **Драйвера** изменить параметр драйвера MEM с *Вкл.* на *Выкл.*

Сохранить изменения в проекте и перезапустить проект.

5. Настройка сигналов

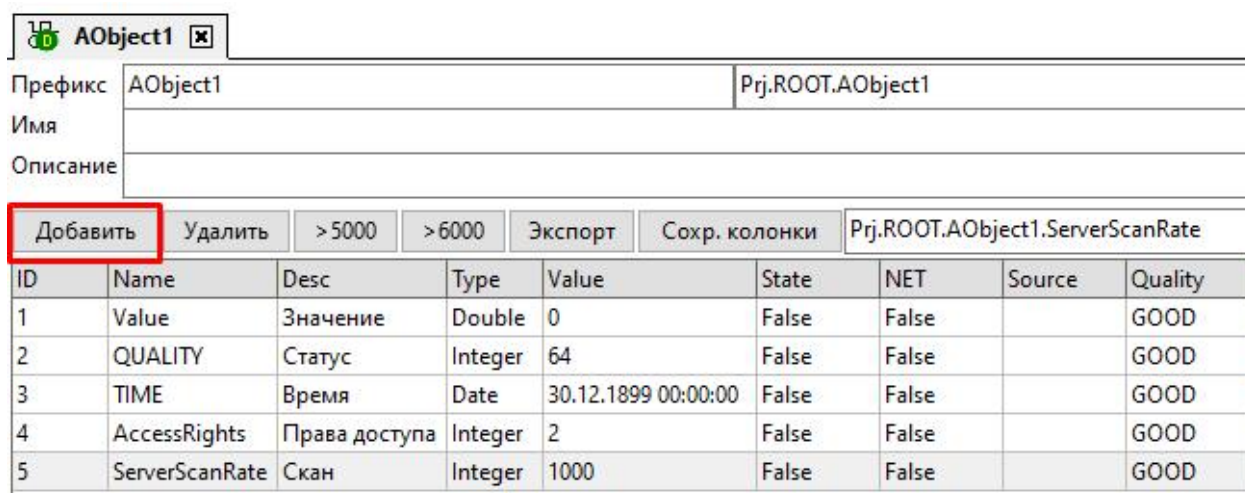
Теги из драйвера записываются внутрь полей сигнала (раздел ROOT в дереве проекта).

5.1 Создание нового сигнала

5.1.1 Для создания нового сигнала необходимо в дереве объектов щёлкнуть ПКМ на объекте *ROOT* и выбрать пункт *Сигнал*. В результате будет создан сигнал.

5.1.2 Для редактирования свойств сигнала необходимо дважды щёлкнуть на объекте сигнала.

5.1.3 Добавить теги сигнала возможно двумя способами: путем нажатия на кнопку *Добавить* в окне сигнала либо при помощи импорта тегов.



ID	Name	Desc	Type	Value	State	NET	Source	Quality
1	Value	Значение	Double	0	False	False		GOOD
2	QUALITY	Статус	Integer	64	False	False		GOOD
3	TIME	Время	Date	30.12.1899 00:00:00	False	False		GOOD
4	AccessRights	Права доступа	Integer	2	False	False		GOOD
5	ServerScanRate	Скан	Integer	1000	False	False		GOOD

Рисунок 5.1 - Пример окна сигнала AObject1

При нажатии на кнопку *Добавить* отображается окно добавления свойства сигнала (см. рисунок 5.1).

В открывшемся окне заполнить требуемые поля и выбрать корректно тип тега (в соответствии с п. 5.2).

После создания заполнить параметр **Source** в соответствии с описанием форматов тегов (см. п. 5.2).

После добавления тегов необходимо сохранить и перезапустить программу.

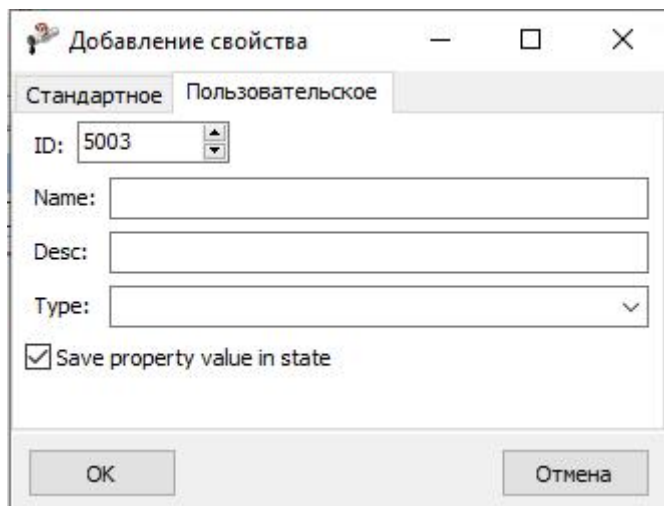


Рисунок 5.2 – Окно добавления свойств сигнала AObject1

5.2 Описание формата источника тегов драйвера MEM

Для подписки на тег в драйвере MEM необходимо:

1. В поле *тип* (Type) сигнала прописать необходимый тип данных. тип.
2. В поле *источник* (Source) сигнала указать источник в формате:

MEM:[Gateway].[Device].[Block].[Register]{Byte}

где:

Gateway - имя шлюза (главного устройства) к которому подключены ведомые устройства;

Device - имя ведомого устройства;

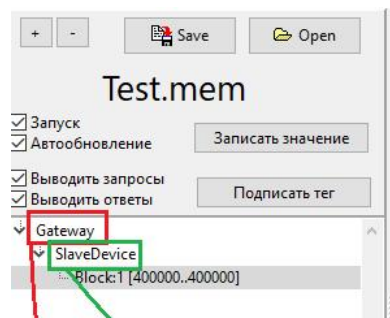
Block - имя блока в ведомом устройстве;

Register - относительный адрес в блоке;

Byte (опционально) - номер бита в регистре, предикатом указать символ амперсанта (&). Нумерация битов начинается с 0.



Пример: MEM:gateway.test.single.2 (обращение ко второму регистру блока "single" ведомого устройства test на шлюзе gateway)
()



ID	Name	Desc	Type	Value	State	NET	Source	Quality
1	Value	Значение	Double	0	False	False		GOOD
2	QUALITY	Статус	Integer	192	False	False		GOOD
3	TIME	Время	Date	17.07.2026 11:23:04	False	False		GOOD
4	AccessRights	Права доступа	Integer	2	False	False		GOOD
5	ServerScanRate	Скан	Integer	1000	False	False		GOOD
5001	test		Word	67	False	False	MEM:Gateway.SlaveDevice.Block.1	GOOD

MEM:Gateway.SlaveDevice.Block.1

Name	Block	Название блока
Data Type	Word	Тип данных блока, задается в конфигурационном файле
Type	03 Holding Registers (Регистры вход)	Тип блока по стандарту Modbus (3,4)
Begin	400000	Начальный адрес блока
Count	1	Количество регистров в блоке, задается в конфигурационном файле
Kind	Читать и писать	Тип работы блока (0,1,2,3)
KindExt	Стандартный Modbus	Вид протокола Modbus (0 - стандарт, 1 - расширенный Emicon)
Interval	500	Периодичность обработки (опроса) блока, мс.
ByteOrder	BigEndian (AB CD)	Порядок байт в полученном ответе
Address	1	Адрес Modbus устройства

TikMes	194	Количество запросов блока у устройства
TikMesGet	194	Количество ответов от устройства
TikMesErr	0	Количество ошибок или ошибочных ответов
TikMesTimeOut	0	Количество срабатываний TimeOut
Last time read	11:26:10.792	Время последнего чтения блока
Quality	GOOD	Качество данных блока
Address	**Word (uint16_t)**	**Byte (uint8_t)**
0 : 400000 : 1	67	0000000001000011

Рисунок 5.3 - Пример отображения типа данных «Word» в конфигураторе и в SCADA на примере тега
MEM:Gateway.SlaveDevice.Block.1

Драйвер позволяет осуществить множественную подписку на разные типы данных по одному источнику (Добавлена возможность указывать в блоке один тип данных, а спрашивать из SCADA другой. Смещение задается исходя из размера типа данных в блоке). Пример адресации на один и тот же регистр разными типами данных представлен в приложении А.

Префикс

AObject1

Pri.ROOT.AObject1

Имя

☐ Debug

Описание

Добавить

Удалить

> 5000

> 6000

Экспорт

Сохранить колонки

Pri.ROOT.AObject1.tag3

ID	Name	Desc	Type	Value	State	NET	Source	Quality	Time	NET Value
1	Value	Значение	Double	0	False	False		GOOD	02.07.2026 15:27:57	
2	QUALITY	Статус	Integer	192	False	False		GOOD	30.12.1899 00:00:00	
3	TIME	Время	Date	02.07.2026 15:27:57	False	False		GOOD	30.12.1899 00:00:00	30.12.1899 00:00:00
4	AccessRi	Провайд	Integer	2	False	False		GOOD	02.07.2026 15:27:57	
5	ServerSci	Скан	Integer	1000	False	False		GOOD	30.12.1899 00:00:00	
5001	tag1		Word	630	False	False	MEM:Dev1.Slave0.MCU1_Line_1.1	GOOD	02.07.2026 15:27:57	
5002	tag2		Integer	41288310	False	False	MEM:Dev1.Slave0.MCU1_Line_1.1	GOOD	02.07.2026 15:27:57	
5003	tag3		Int64	177331941198398070	False	False	MEM:Dev1.Slave0.MCU1_Line_1.1	GOOD	02.07.2026 15:27:57	

Рисунок 5.4 - Пример подписки на один и тот же регистр разными типами данных

Если устройство и тег указаны верно, параметр **Quality** перейдёт в состояние **UNCERTAIN**, что демонстрирует корректное обращение к конфигурации. В противном случае он перейдёт в состояние **SOURCE FAILURE**.



Внимание!

Адресация из SCADA к тегам в драйвере MEM начинается с 1 и не зависит от типа данных. Подразумевается, что данные в каждом блоке начинаются с первого регистра и идут подряд, т.е 1,2,3,4.

5	ServerSc	Скан	Integer	1000	False	False		GOOD
5001	test1		Word	67	False	False	MEM:Gateway.SlaveDevice.Block.1	GOOD
5002	test2		Word	42	False	False	MEM:Gateway.SlaveDevice.Block.2	GOOD
5003	test3		Word	314	False	False	MEM:Gateway.SlaveDevice.Block.3	GOOD

Name	Block	Название блока
DataType	Word	Тип данных блока, задается в конфигурационном файле
Type	03 Holding Registers (Регистры вход)	Тип блока по стандарту ModBus (3,4)
Begin	400000	Начальный адрес блока
Count	10	Кол-во регистров в блоке, задается в конфигурационном файле
Kind	Читать и писать	Тип работы блока (0,1,2,3)
KindExt	Стандартный Modbus	Вид протокола Modbus (0 - стандарт, 1 - расширенный Emicon)
Interval	500	Периодичность обработки (опроса) блока, мс.
ByteOrder	BigEndian (AB CD)	Порядок байт в полученном ответе
Address	1	Адрес Modbus устройства

TikMes	22	Количество запросов блока у устройства
TikMesGet	22	Количество ответов от устройства
TikMesErr	0	Количество ошибок или ошибочных ответов
TikMesTimeOut	0	Количество срабатываний TimeOut
Last time read	11:36:08.147	Время последнего чтения блока
Quality	GOOD	Качество данных блока
Address	**Word (uint16_t)**	**Byte (uint8_t)**
0 : 400000 : 1	67	0000000001000011
1 : 400001 : 2	42	000000000101010
2 : 400002 : 3	314	0000000100111010

Рисунок 5.5 - Пример адресации к тегам

5.2.1 Получение качества (Quality) блока данных

Для получения качества блока в драйвере MEM необходимо:

1. В поле *тип* (Type) сигнала прописать необходимый тип данных. тип.
2. В поле *источник* (Source) сигнала указать источник в формате:

MEM:[Gateway].[Device].[Block].[Qxxx]

где:

Gateway - имя шлюза (главного устройства), к которому подключены ведомые устройства;

Device - имя ведомого устройства;

Block - имя блока в ведомом устройстве;

Qxxx - где xxx либо пустой набор символов, либо любое число - в результате вернется качество блока Block.



Пример: MEM:gateway.test.single.Q2 (получение качества блока “single” ведомого устройства test на шлюзе gateway)

5.2.2 Префиксы

Свойство сигнала работает в следующих режимах:

- подписка (чтение и запись);
- принудительное чтение;
- принудительная запись;

Если значение в поле *Source* не содержит префикса, то произойдет **подписка (чтение и запись)** тега в драйвере и в результате этого значение в поле *Value* будет обновляться. Для однократной записи значения тега, на который оформлена подписка, в контроллер необходимо изменить значение в поле *Value* – значение будет отправлено на запись.

Префикс << в поле *Source* задает режим **принудительного чтения**. Значения в поле *Value* будут принудительно считываться и обновляться напрямую из контроллера независимо от других условий или настроек. Это позволяет всегда получать актуальные данные с устройства в реальном времени.



Пример: <<MEM:GW1.C1.R1.6

Префикс >> в поле *Source* задает режим **принудительной записи**. Значения из поля *Value* автоматически и обязательно записываются в контроллер независимо от других условий или ограничений, перезаписывая текущие значения устройства.



Пример: >>MEM:GW1.C1.R1.6

**Внимание!**

При удалении ранее заведенных подписок на теги необходим перезапуск K-System SCADA с сохранением базы тегов.

6. Запуск ввода-вывода драйвера MEM

Перед запуском необходимо убедиться, что:

- проект запущен на основном сервере;
- контроллер подключен к сети;

Для **запуска ввода-вывода** K-System SCADA необходимо нажать на кнопку

▶ *Запуск/останов системы ввода/вывода.*

После успешного запуска системы ввода/вывода драйвера MEM, контроллер должен установить соединение с драйвером и начать передачу сообщений с обновлёнными значениями тегов. При этом, у обновлённых тегов параметр *Quality* должен перейти из состояния *UNCERTAIN* в состояние *GOOD*, что будет свидетельствовать о корректной передаче данных, см. рисунок 6.1.

5	ServerScd	Скан	Integer	1000	False	False		GOOD
5001	test1		Word	67	False	False	MEM:Gateway.SlaveDevice.Block.1	GOOD
5002	test2		Word	42	False	False	MEM:Gateway.SlaveDevice.Block.2	GOOD
5003	test3		Word	314	False	False	MEM:Gateway.SlaveDevice.Block.3	GOOD

Name	Block	Название блока
DataType	Word	Тип данных блока, задается в конфигурационном файле
Type	03 Holding Registers (Регистры ввода)	Тип блока по стандарту ModBus (3,4)
Begin	400000	Начальный адрес блока
Count	10	Кол-во регистров в блоке, задается в конфигурационном файле
Kind	Читать и писать	Тип работы блока (0,1,2,3)
KindExt	Стандартный Modbus	Вид протокола Modbus (0 - стандарт, 1 - расширенный Emicon)
Interval	500	Периодичность обработки (опроса) блока, мс.
ByteOrder	BigEndian (AB CD)	Порядок байт в полученном ответе
Address	1	Адрес Modbus устройства

TikMes	22	Количество запросов блока у устройства
TikMesGet	22	Количество ответов от устройства
TikMesErr	0	Количество ошибок или ошибочных ответов
TikMesTimeOut	0	Количество срабатываний TimeOut
Last time read	11:38:08.147	Время последнего чтения блока
Quality	GOOD	Качество данных блока
Address	**Word (uint16_t)**	**Byte (uint8_t)**
0 : 400000 : 1	67	0000000001000011
1 : 400001 : 2	42	0000000000101010
2 : 400002 : 3	314	0000000100111010

Рисунок 6.1 - Корректная передача данных после успешного соединения контроллера с драйвером MEM

7. Перечень поддерживаемых драйвером типов данных

В таблице ниже приведено описание поддерживаемых драйвером типов.

Наименование	Описание
Smallint	Знаковые целые числа размером 2 байта в диапазоне от - 32768 до 32767
Integer	Знаковые целые числа размером 4 байта в диапазоне от -2 147 483 648 до 2 147 483 647
Single	Числа с плавающей запятой размером 4 байта
Double	Числа с плавающей запятой размером 8 байта
Date	Дата и время с точностью до миллисекунд
Boolean	Позволяет хранить логическое значение true или false
ShortInt	Знаковые целые числа размером 1 байта в диапазоне от -128 до 127
Byte	Беззнаковые целые числа размером 1 байта в диапазоне от 0 до 255
Word	Беззнаковые целые числа размером 2 байта в диапазоне от 0 до 65535
LongWord	Беззнаковые целые числа размером 4 байта в диапазоне от 0 до 4294967295
Int64	Знаковые целые числа размером 8 байта в диапазоне от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807

8. Описание статистики

Для **просмотра статистики** необходимо щелкнуть на объект *MEM* в дереве объектов на вкладке *DRV*. В результате будет отображено окно раздела Статистика (см. рис. 8.1).

В разделе *Статистика* отображается общая статистика и статистика по блокам.

При нажатии на кнопку *Статистика* отображается **общая статистика**, включающая в себя следующие данные:

- количество тегов, на которое была оформлена подписка;
- количество обновлений тегов;
- количество обработанных обновлений тегов;
- частота событий обновления тегов;
- частота обновлений тегов.

При нажатии на кнопку *Теги драйвера* отображается **статистика по блокам данных** (для каждого блока), включающая в себя следующие данные (см. рис. 8.2):

- размер блока;
- количество обновлений блока;
- время последнего обновления блока;
- время между последними обновлениями блока в миллисекундах.

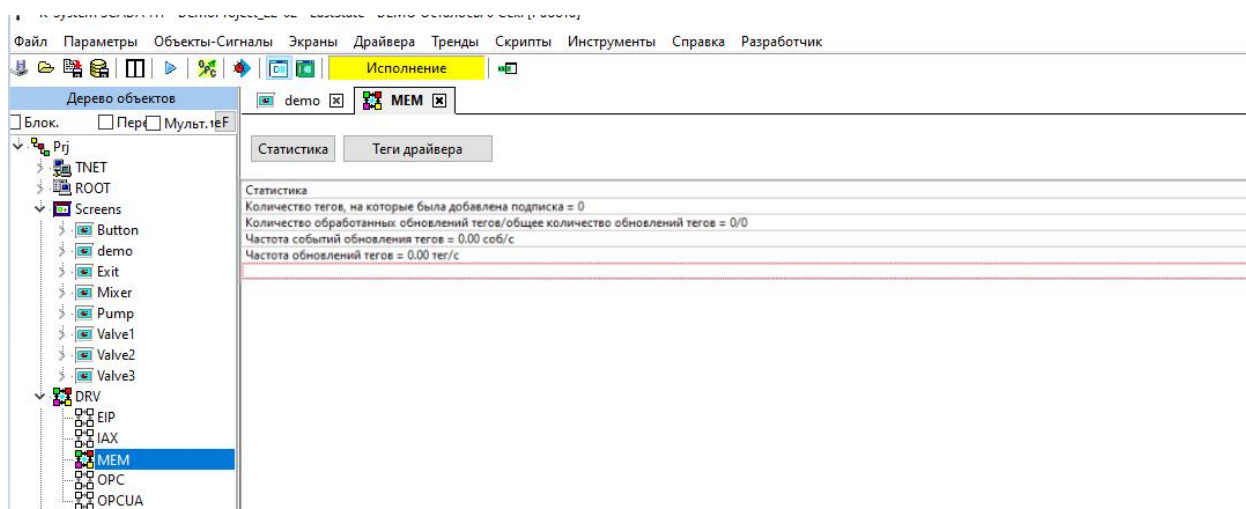


Рисунок 8.1 – Пример отображения общей статистики



9. Логирование

Список лог - сообщений в конфигураторе представлен в таблице ниже.

Сообщение	Описание
Сообщения работы сетевого обмена драйвера MEM:	
ReadBlok: Соединение с %Host% закрыто, принудительный разрыв	Произошла ошибка сети, которая привела к разрыву соединения с устройством.
ReadBlok: Таймаут при чтении с %Host%, разрыв соединения	Произошло превышение таймаута при чтении данных с устройства, которое привело к разрыву соединения с устройством.
ReadBlok: Нет данных в буфере от %Host%, разрыв соединения	Произошло нарушение целостности пакета, считанного от устройства.
Wrong buffer	Произошло нарушение целостности пакета, считанного от устройства.
Modbus slave при запросе чтения с блока %BlockName% передал отклик %FunctionCode%, код ошибки %ErrorCode%	Устройство сигнализирует об ошибке при попытке чтения данных.
Modbus slave при запросе чтения с блока %BlockName% передал отклик %FunctionCode%	Произошло нарушение целостности пакета, считанного от устройства.
ReadBlok: Исключение при чтении с %Host%, разрыв соединения	Возникла внутренняя ошибка драйвера тем, которая привела к разрыву соединения с устройством.
WriteBlokByReg: Соединение с %Host% закрыто, принудительный разрыв	Произошла ошибка сети, которая привела к разрыву соединения с устройством.
WriteBlokByReg: Таймаут при записи на %Host%, разрыв соединения	Произошло превышение таймаута при записи данных в устройство, которое привело к разрыву соединения с устройством.
WriteBlokByReg: Нет ответа от %Host%, разрыв соединения	Устройство не шлёт ответ на запрос записи, что привело к разрыву соединения.
Modbus slave при запросе записи в блок %BlockName% передал отклик %FunctionCode%, код ошибки %ErrorCode%	Устройство сигнализирует об ошибке при попытке чтения данных.
Modbus slave при запросе записи в блок %BlockName% передал отклик %FunctionCode%	Произошло нарушение целостности пакета, считанного от устройства.
Исключение при записи интерфейса 1 :%ErrorText%	Произошла ошибка при попытке записи блока.

Сообщение	Описание
Исключение при записи WriteBlokByReg :%ErrorText%	Произошла ошибка при попытке записи регистра.
Сообщения работы менеджера подключений драйвера MEM:	
Драйвер MEM. Нет подключения. Отсутствует разрешённый хост	Оба источника данных помечены как запрещённые (host1enabled = false, host2enabled = false).
Драйвер MEM. Установлено соединение с %HostId% (%HostAddress%:%Port%)	Установлено соединение.
Драйвер MEM. Попытка подключения к %HostId% (%HostAddress%:%Port%) не удалась -	Производится попытка восстановить соединение с основным подключением.
Драйвер MEM. Попытка подключения не удалась	Нет активного подключения; будут совершены попытки подключиться к первому появившемуся источнику
Драйвер MEM. %HostId% (%HostAddress%:%Port%) запрещён, производится обрыв соединения	Активное соединение было помечено как запрещённое, что привело к обрыву соединения.
Драйвер MEM. Соединение с %HostId% (%HostAddress%:%Port%) оборвалось	Активное соединение оборвалось.
Драйвер MEM. %HostId% (%HostAddress%:%Port%) запрещён, соединение с ним восстановлено не будет	Соединение, к которому были совершены попытки переподключения, было помечено как запрещённое.
Драйвер MEM. Соединение с %HostId% (%HostAddress%:%Port%) восстановлено	Соединение, к которому были совершены попытки переподключения, было восстановлено.
Драйвер MEM. Попытка переподключения с %HostId% (%HostAddress%:%Port%) не удалась	Производится попытка восстановить соединение с предыдущим активным подключением.
Драйвер MEM. Производится переключение на %HostId% (%HostAddress%:%Port%)	Соединение с предыдущим активным подключением восстановить не удалось, установлено соединение с другим источником.
Драйвер MEM. Попытка переподключения с %HostId% (%HostAddress%:%Port%) не удалась	Соединение с предыдущим активным подключением восстановить не удалось; будут совершены попытки подключиться к первому появившемуся источнику.

10. Описание спецификации расширенных функций Emicon 67, 70, 99.

Форматы запроса и ответа для расширенных функций заимствованы от стандартных функций протокола 3, 4, 16, 23 соответственно. Имеется несколько важных отличий:

- 1) Количество регистров для записи от 1 до 717.
- 2) Количество регистров для чтения от 1 до 719.
- 3) Данные в запросе и ответе имеют порядок байт Little Endian, а в регистрах заголовок Big Endian (в стандартных командах все Big Endian).
- 4) В ответе на функции чтения поле количество байт по прежнему однобайтовое и не учитывает переполнение. То есть если запрашивается 550 регистров $550 \times 2 = 1100_{dec} = 0x044C$, в байте означающем количество байт будет значение $0x4C$. Это значение может быть отброшено или же использовано для дополнительной проверки корректности заголовка ответа.

Функция 67- Расширенное чтение HOLDING регистров

Поле	Количество байт	Порядок байт	ОДЗ	Комментарий
Запрос:				
Address	1	-	0-255	Адрес подчиненного устройства (слейва/сервера). Для TCP используется произвольный, если иное явно не указано в конфигурации.
Function Code	1	-	67	Код Modbus функции
Start Register Num for Read	2	BE	0...65535	Номер стартового регистра для чтения.
Count for Read	2	BE	0...719	Количество регистров для чтения.
Ответ:				
Address	1	-	0-255	Адрес подчиненного устройства.
Function Code	1	-	67	Код Modbus функции.

Byte Count	1	-	(2 * Count for Read) MOD 256	Количество байт в ответе без учета переполнения.
Data	2 * Count for Read	LE (!)	-	Данные ответа.

Функция 70- Расширенная запись HOLDING регистров

Поле	Количество байт	Порядок байт	ОДЗ	Комментарий
Запрос:				
Address	1	-	0-255	Адрес подчиненного устройства (слейва/сервера). Для TCP используется произвольный, если иное явно не указано в конфигурации.
Function Code	1	-	70	Код Modbus функции.
Start Register Num for Write	2	BE	0...(65536 Count for Write)	Номер стартового регистра для записи. В сумме с количеством регистров не должен превышать значение 65536.
Count for Write	2	BE	0...717	Количество регистров для записи.
Data	2 * Count for Write	LE (!)	-	Данные для записи.
Ответ:				
Address	1	-	0-255	Адрес подчиненного устройства (слейва/сервера). Для TCP используется произвольный, если иное явно не указано в конфигурации.
Function Code	1	-	70	Код Modbus функции.
Start Register Num for Write	2	BE	0...(65536 Count for Write)	Номер стартового регистра для записи. В сумме с количеством регистров не должен превышать значение 65536.
Count for Write	2	BE	0...717	Количество регистров для

				записи.
Функция 99- Расширенное чтение и запись HOLDING регистров				
Поле	Количество байт	Порядок байт	ОДЗ	Комментарий
Запрос:				
Address	1	-	0-255	Адрес подчиненного устройства (слейва/сервера). Для TCP используется произвольный, если иное явно не указано в конфигурации.
Function Code	1	-	99	Код Modbus функции.
Start Register Num for Read	2	BE	0...65535	Номер стартового регистра для чтения.
Count for Read	2	BE	0...719	Количество регистров для чтения.
Start Register Num for Write	2	BE	0...(65536 Count for Write)	Номер стартового регистра для записи. В сумме с количеством регистров не должен превышать значение 65536.
Count for Write	2	BE	0...717	Количество регистров для записи.
Length	1	-	0-255	Длина передаваемого сообщения в байтах.
Data	2 * Count for Write	LE (!)	-	Данные для записи.
Ответ:				
Address	1	-	0-255	Адрес подчиненного устройства .
Function Code	1	-	99	Код Modbus функции.
Byte Count	1	-	(2 * Count for Read) MOD 256	Количество байт в ответе без учета переполнения
Data	2 * Count for Read	LE (!)	-	Данные ответа

Таблица: Modbus-регистры SLAVE- устройств.

Тип регистра	Диапазон адресов (5-значное число)	Функции
DISCRETE INPUTS (Дискретные входы)	10001-19999	
COILS (Дискретные выходы)	20001-29999	
INPUTS (16-битные входы)	30001-39999	04 – Чтение группы регистров (макс. 125 регистров)
HOLDING REGISTERS (16-битные выходы)	40001-49999	03 – Чтение группы регистров (макс. 125 регистров) 06 – Запись 1 регистра 16 – Запись групп регистров (макс. 125 регистров)
		67 расширенное чтение (макс. 719 регистров) 70 расширенная запись (макс. 717 регистров) 99 чтение и запись (макс. 719 для чтения/макс. 717 регистров для записи)

11. Порядок байт в данных, полученных драйвером

По умолчанию драйвер MEM имеет формат сообщений с порядком байт Big Endian. Поле *ByteOrder* из конфиг файла принимает значения, соответствующие полю Возможные значения в таблицах ниже для соответствующих типов. Также возможно изменение через конфигуратор путем двойного клика ЛКМ по значению поля *ByteOrder* с последующим сохранением конфигурационного файла и при необходимости обновления конфигурации в SCADA (см. раздел 3.5).

Kind	Читать и писать	Тип работы блока (0,1,2,3)
KindExt	Стандартный Modbus	Вид протокола Modbus (0 - стандарт, 1 - расширенный)
Interval	500	Периодичность обработки (опроса) блока, мс.
ByteOrder	BigEndian (AB CD)	Порядок байт в полученном ответе
Address	1	Адрес Modbus устройства

TikMes	0	Количество запросов блока у устройства
TikMesGet	0	Количество ответов от устройства
TikMesErr	0	Количество ошибок или ошибочных ответов
TikMesTimeOut	0	Количество срабатываний TimeOut
Last time read	00	Время последнего чтения блока
Quality	UNCERTAIN	Качество данных блока
Address	**Word (uint16_t)**	**Byte (uint8_t)**
0 : 400000 : 1		0000000000000000

Рисунок 11.1 – Контекстное меню выбора порядка байт

	Внимание!
	Конфигурация протокола «Расширенный Emicon» предусматривает, что значение ByteOrder по умолчанию должно быть равно 0, т.к. согласно спецификации данного протокола порядок байт в данных соответствует формату Big Endian.

Порядок байт значения из ПЛК относительно ABCD, A - старший байт:

Режим	Возможные значения	Результат	Формула	Описание
boBigEndian	0, bigendian, be, big-endian,	12 34	ABCD	Без изменений (по умолчанию)

Режим	Возможные значения	Результат	Формула	Описание
boLittleEndian	1, <i>littleendian,</i> <i>le, little-</i> <i>endian</i>	34 12	DCBA	Полная перестановка байт
boWordSwapLE	2, <i>wordswaple,</i> <i>wordswap</i> <i>little-endian</i>	34 12	CDAB	Престановка слов, BE в каждом
boWordSwapBE	3, <i>wordswapbe,</i> <i>wordswap</i> <i>big-endian</i>	34 12	BADC	прямой Порядок слов, LE в каждом

Приложение А

Пример адресации на один и тот же регистр разными типами данных

MEM			SCADA			
DBlock Int16			Source = MEM:Gateway:SlaveDevice:DBlock: 2			
Register		RawValue	Tag DataType			
Absolute addr	Relative addr		Register ID	Int 16	Int32	Int64
0	0	00000001	1			
1	1	00000001	2	00000010	00000010	00000010
2	2	00000010	3		00000001	00000001
3	3	00000001	4			00000010
4	4	00000010	5			00000001
5	5	00000001	6			
6	6	00000010	7			
7	7	00000010	8			
8	8	00000001	9			

MEM			SCADA			
DBlock Int32			Source = MEM:Gateway:SlaveDevice:DBlock: 2			
Register		RawValue	Tag DataType			
Absolute addr	Relative addr		Register ID	Int 16	Int32	Int64
0	0	00000001	1			
1		00000010				
2	1	00000001	2	00000010	00000010	00000010
3		00000010			00000001	00000001
4	2	00000001	3			00000010
5		00000010				00000001
6	3	00000001	4			
7		00000010				

MEM			SCADA			
DBlock Int64			Source = MEM:Gateway:SlaveDevice:DBlock: 2			
Register		RawValue	Tag DataType			
Absolute addr	Relative addr		Register ID	Int 16	Int32	Int64
0	0	00000001	1			
1		00000010				
2		00000001				
3		00000010				
4	1	00000001	2	00000010	00000010	00000010
5		00000010			00000001	00000001
6		00000001				00000010
7		00000010				00000001

[illegible]