

Инструкция

**Организация чтения архивных файлов из памяти
ОВЕН ПЛК.**

2009

Содержание

Содержание	2
Введение	3
1 Настройка ПЛК.....	3
1.1 Архивация данных	3
1.1.1 Добавление архивации в проект	3
1.1.2 Настройка параметров архивации	4
1.2 Организация передачи исторических данных с ПЛК по 20-ой функции Modbus 7	
2 Настройка Lectus OPC.....	11
2.1 Чтения файла архива.....	11
2.1.1 Добавление исторического Modbus узла	11
2.1.2 Добавление переменных в исторический Modbus узел.....	14
2.2 Предоставление данных HDA-клиентам	16
3 Чтение исторических данных HDA-клиентом	19
Приложение	24
3.1 Форматы запросов и ответов при обмене по 20-ой функции Modbus.....	24

Введение

Настоящий документ является инструкцией по организации передачи архивных данных из энергонезависимой памяти ОВЕН ПЛК в SCADA-системы поддерживающие стандарт OPC HDA1.20. Передача архивных данных реализуется при помощи OPC-сервера Lectus OPC.

Для организации обмена архивными (историческими) данными необходимо настроить ОВЕН ПЛК и Lectus OPC на работу по 20-ой функции ModBus.

1 Настройка ПЛК

В данном проекте ПЛК выполняет следующие функции:

- 1) Архивация данных;
- 2) Передача архивных данных в качестве Slave-устройства.

1.1 Архивация данных

1.1.1 Добавление архивации в проект

Архивация данных в ОВЕН ПЛК осуществляется специально созданным модулем **«Archiver»**. Добавление модуля «Archiver» в PLC_Configuration представлено на рисунке 1.1.

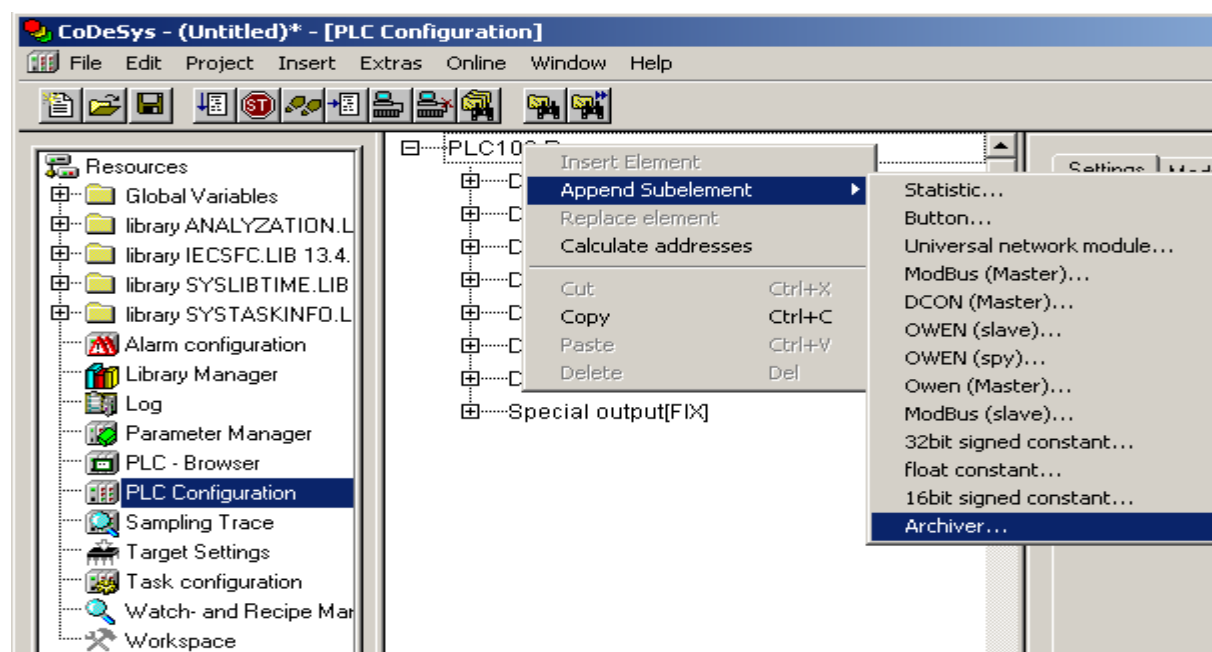


Рисунок 1.1 - Добавление модуля «Archiver»

1.1.2 Настройка параметров архивации

Параметры архивации должны соответствовать поставленной задаче. В данном примере архивация производится по времени (Archive Mode = By timer), данные сохраняются в файл в текстовом виде удобном для чтения пользователя (Archive Type = ASCII only) с периодом архивации 5 секунд (Period of Archiving = 5) (Рисунок 1.2).

Полное описание всех параметров модуля «Archiver» и способы работы с ним представлено в документе «PLC_Configuration_OWEN.pdf», находящимся на Диске ПЛК в разделе «Документация».

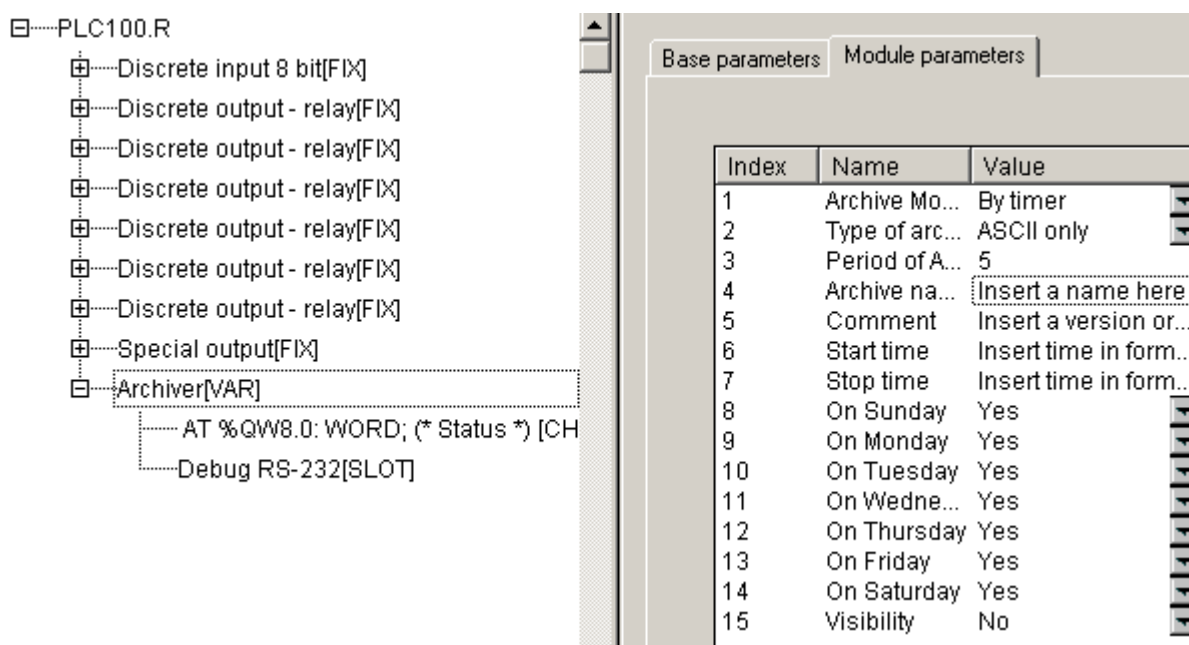


Рисунок 1.2 – Настройка параметров архивации модуля «Archiver»

Архивацию данных необходимо производить в файл, для этого измените подмодуль интерфейса на «File Output» (Рисунок 1.3.).

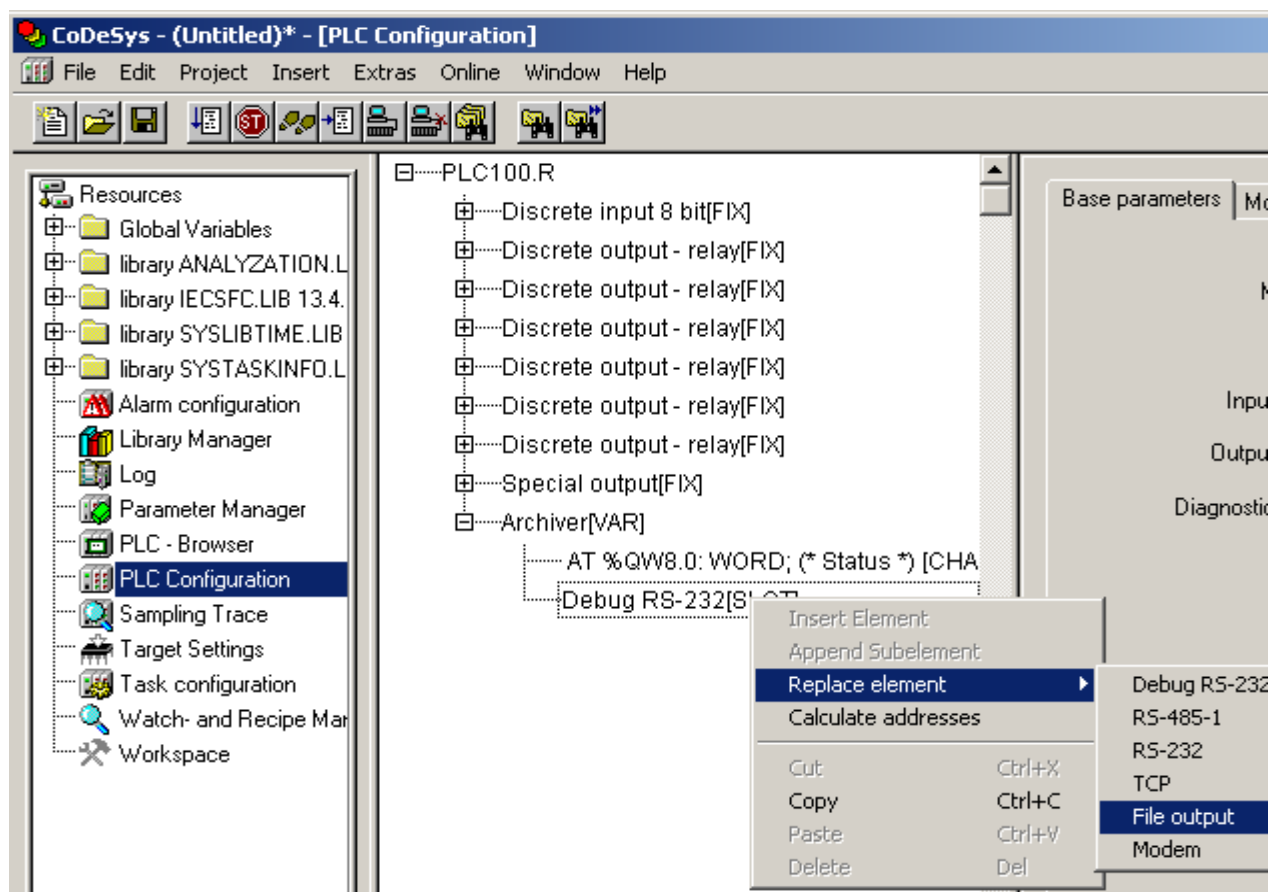


Рисунок 1.3 – Смена подмодуля интерфейса на «File Output»

Задать параметры подмодуля «File Output», включая имя «File name» файла и максимальный размер «Max file size»¹ (Рисунок 1.4).

Полное описание всех параметров подмодуля «File Output» и способы работы с ним представлено в документе «PLC_Configuration_OWEN.pdf», находящимся на Диске ПЛК в разделе «Документация».

¹ При большом количестве архивных файлов с большими размерами время цикла ПЛК может, увеличиться, поэтому рекомендуется увеличить минимальное значение цикла ПЛК.

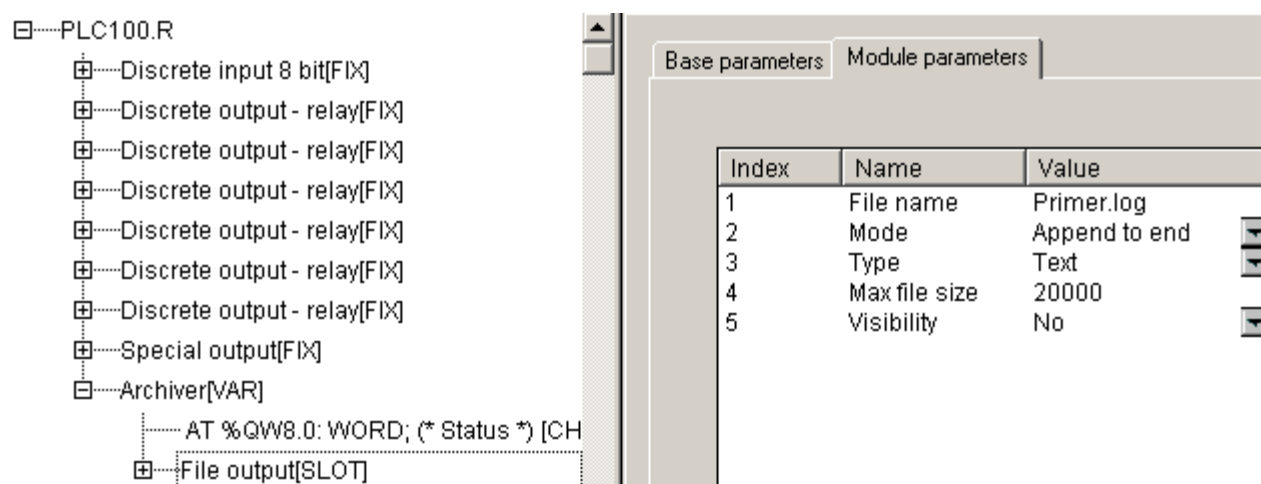


Рисунок 1.4 – Настройка параметров подмодуля «File Output»

Добавить переменные для архивации, в соответствии с поставленной задачей (Рисунок 1.5).

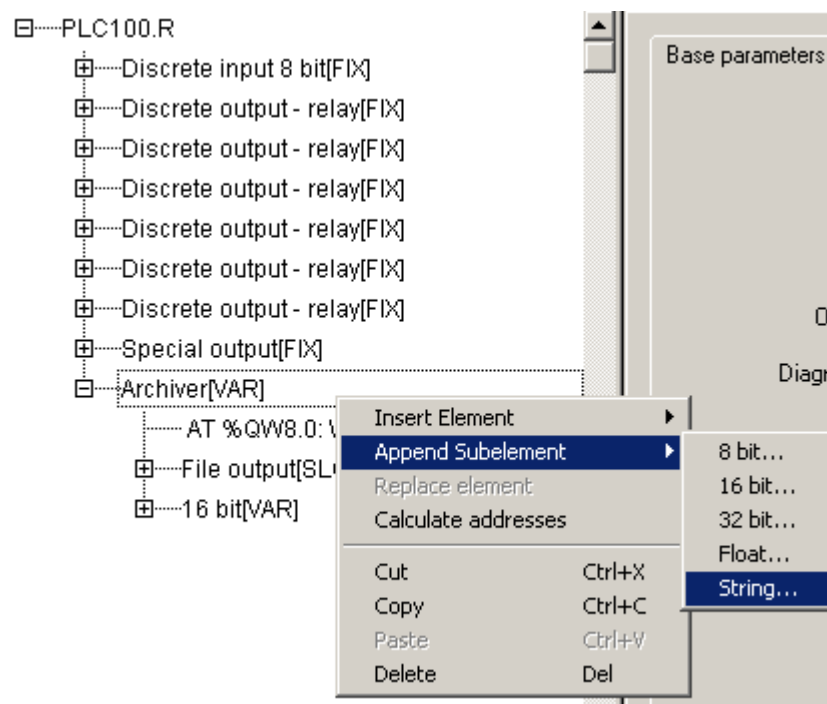


Рисунок 1.5 – Добавление переменных для архивации

Присвоить имена добавленным переменным (Рисунок 1.6).

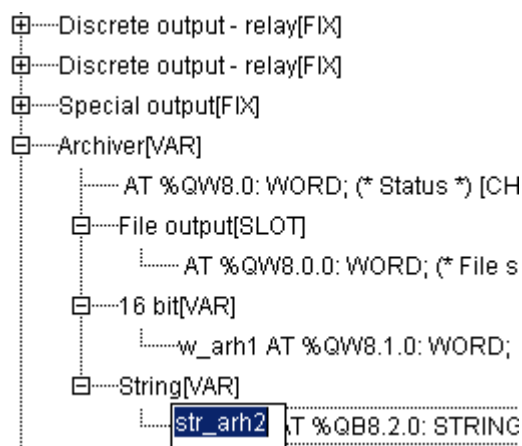


Рисунок 1.6 – Присвоение имен переменным

1.2 Организация передачи исторических данных с ПЛК по 20-ой функции Modbus

Записанные при помощи модуля «Archiver» данные необходимо передать через интерфейс по протоколу ModBus. Для передачи архивных файлов необходимо добавить модуль «Modbus(Slave)» (Рисунок 1.7).

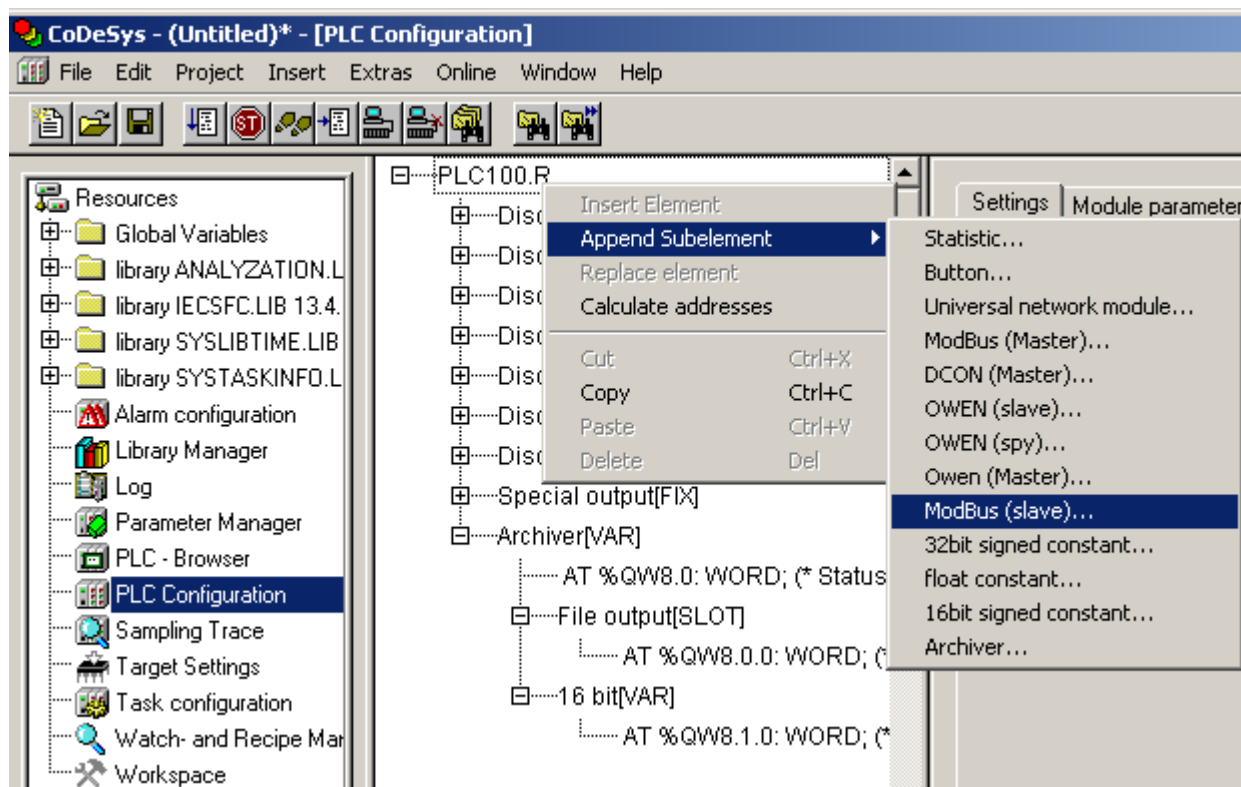


Рисунок 1.7 - Добавление модуля «Modbus(Slave)»

В параметрах модуля «Modbus slave» задать адрес устройства (ПЛК) в сети (по умолчанию 1) (Рисунок 1.8).

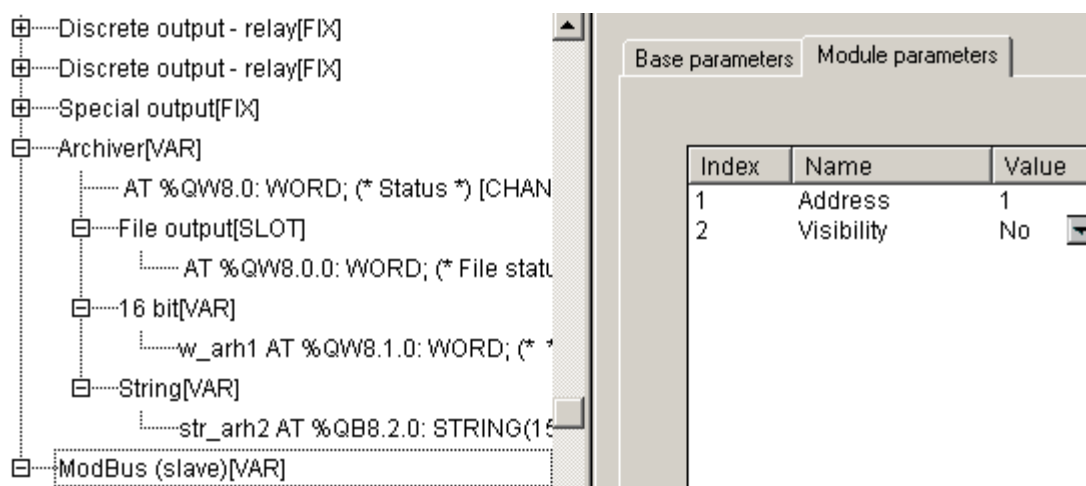


Рисунок 1.8 – Параметры модуля «Modbus (Slave)»

Для передачи архивного файла мастеру сети необходимо добавить подмодуль «File» (Рисунок 1.9). Для передачи файла по 20-й функции modbus каждому файлу должен соответствовать индивидуальный номер. Номер файла присваивается каждому подмодулю «File» автоматически, начиная с 0.

Например: При добавлении 1-го подмодуля «File» мастер должен обращаться к нему как к файлу с номером 0, при добавлении 2-го подмодуля «File» – как к файлу с номером 1 и т.д.

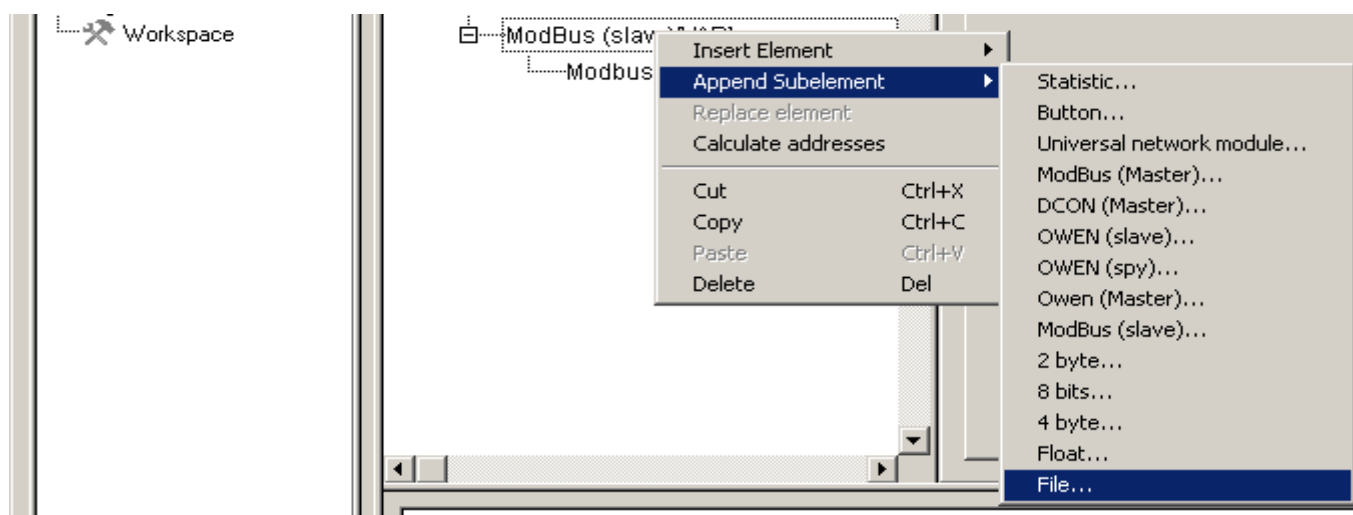


Рисунок 1.9 - Добавление подмодуля «File»

В параметрах подмодуля «File» необходимо указать: «File name» – имя архивного файла, который был создан модулем Archiver; «Amount byte» – размер одной физической записи истории в байтах.

Для данного примера «File name» = Primer.log, «Amount Byte» = 10 (Рисунок 1.10).

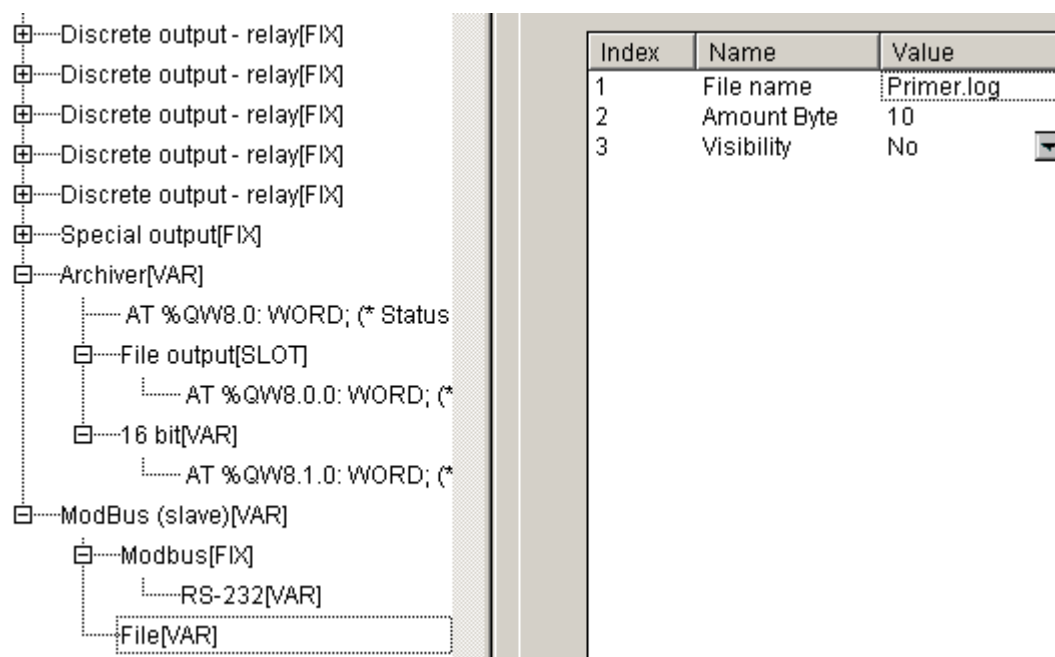


Рисунок 1.10 – Параметры подмодуля «File»

Данные из файла передаются по коммуникационному интерфейсу, настройка которого осуществляется аналогично, передаче текущих данных. В данном примере ПЛК будет осуществлять передачу данных по интерфейсу RS-232, добавление и настройка параметров которого соответственно представлены на рисунках 1.11, 1.12.

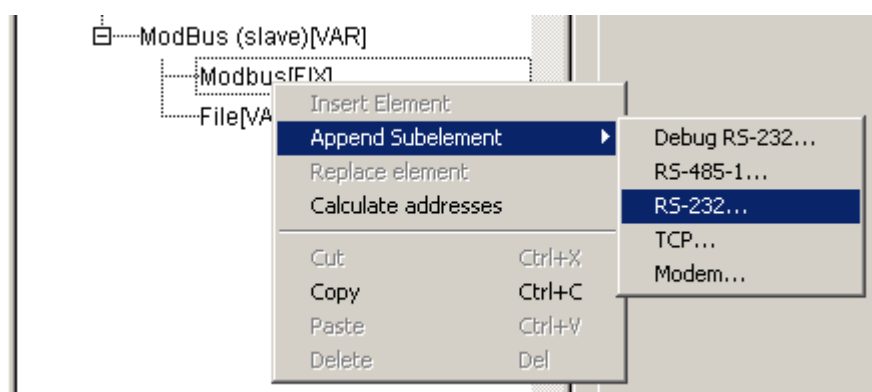


Рисунок 1.11 - Добавление коммуникационного интерфейса

Index	Name	Value	Default
1	Communication sp...	11520	11520
2	Parity	NO PARITY CHECK	NO PARITY ...
3	Data bits	8 bits	8 bits
4	Stop length	One stop bit	One stop bit
5	Interface Type	RS232	RS232
6	Frame oriented	ASCII	ASCII
7	Framing time ms	0	0
8	Visibility	No	No

Рисунок 1.12 – Настройка параметров коммуникационного интерфейса

Полное описание всех параметров коммуникационного интерфейса представлено в документе «PLC_Configuration_OWEN.pdf», находящимся на Диске ПЛК в разделе «Документация».

На этом создание проекта в CoDeSys закончено. Необходимо сохранить проект и загрузить его в ПЛК.

2 Настройка Lectus OPC

Lectus OPC поддерживает чтение исторических данных по 20-ой функции Modbus. В данном примере Lectus OPC выполняет следующие функции:

- 1) Чтение файла архива из памяти ПЛК по записям;
- 2) Преобразование данных из файла и предоставление их HDA-клиентам (версия OPC HDA 1.20);

2.1 Чтения файла архива

Чтение файла архива из ПЛК в Lectus OPC осуществляется через исторический Modbus узел по записям.

2.1.1 Добавление исторического Modbus узла

Добавьте новый узел **текущих данных** (Рисунок 2.1);

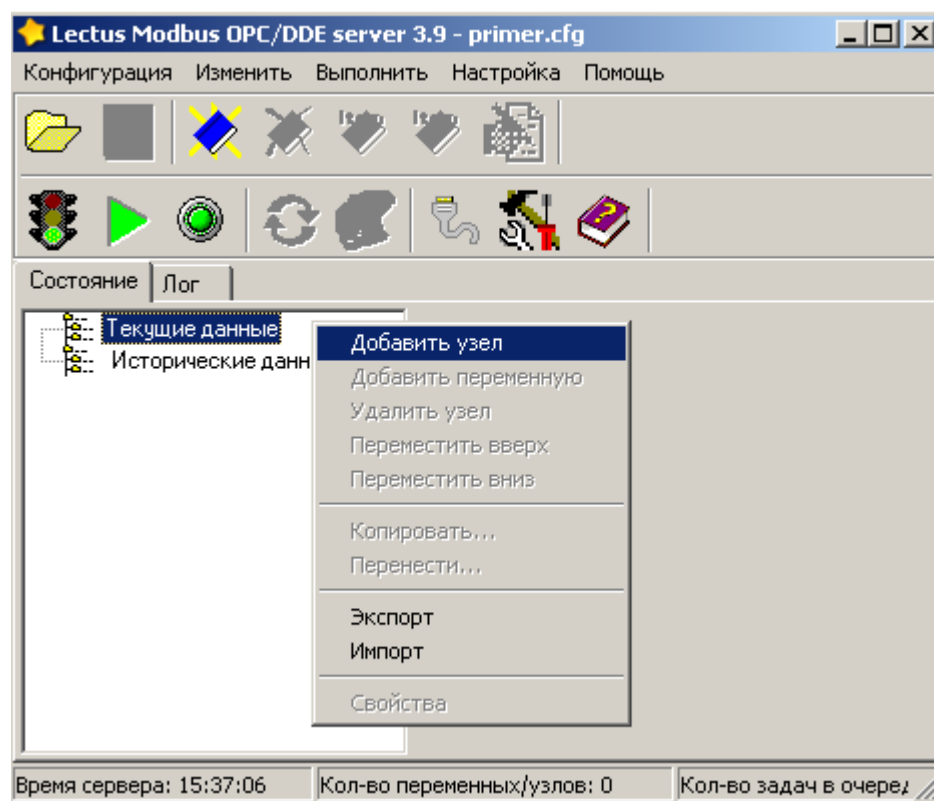


Рисунок 2.1 – Добавление нового узла текущих данных

В появившемся окне «Добавить узел» выбрать вкладку «Modbus», где указать:

- «имя узла» – primer;

- «Подключение» – Прямое подключение (Нажав на кнопку «Настройка..» выберите COM-порт);
- «тип данных» – **История**;
- Опрос – 5 мин.;
- настройки устройства (тип протокола идентичен выбранному в ПЛК, в качестве адреса укажите адрес ПЛК, при выборе типа данных «история» функция чтения по умолчанию = **20**);

Полное описание параметров Modbus узла представлено в справке по Lectus OPC.

Окно «Добавить узел» представлено на рисунке 2.2.

Рисунок 2.2 – Окно «Добавить узел»

Не закрывая окно нажать на кнопку «Параметры» поля «Формат данных». Появится новое окно программы «Формат данных истории», в котором необходимо задать следующие параметры архивного файла:

- «Размер физической записи истории» – указывается равный значению параметру «Amount Byte» подмодуля «File» проекта CoDeSys;

- «Номер файла» – номер файла (0, 1, 2, 3...), который будет читаться по 20-ой функции, в процессе работы может быть изменен с помощью системной исторической переменной HISTORY_FILE_NUMBER;
- «Номер первой записи» – номер записи, с которой начнётся читаться файл;
- «Номер последней записи» – номер записи, до которой будет читаться файл (если количество записей в файле меньше указанного, то чтение корректно прекратится на последней записи файла);
- «Формат данных» – выбрать «История ОВЕН», для **наличия в списке данного типа необходимо добавить файл opcserv.key, поставляемый вместе с лицензионной программой, в директорию программы Lectus OPC**;
- «Формат истории» – указывается тип, как в параметре «Archive Type» модуля «Archiver» проекта CoDeSys (если был задан «ASCII only», то установить «Текстовый», если Mixed – «Бинарный»);

Окно «Формат данных истории» представлено на рисунке 2.3.

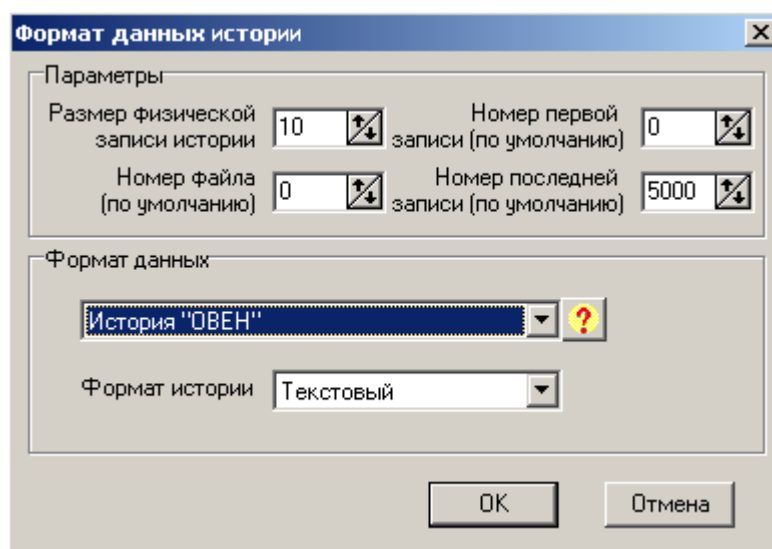


Рисунок 2.3 – Окно «Формат данных истории»

Подтвердите изменения, нажав кнопку ОК.

Вернувшись в окно «Добавить узел» нажмите на кнопку «Параметры» поля «Сохранение». Появится новое окно программы «Параметры сохранения истории», в котором необходимо выбрать из списка «База данных» (Рисунок 2.4).

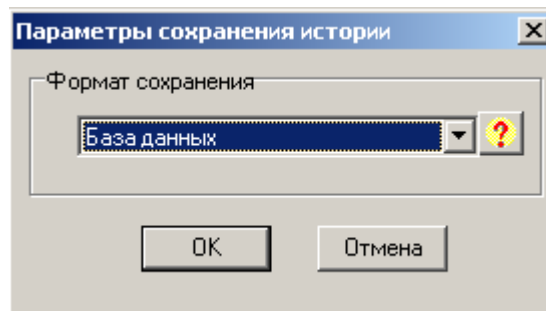


Рисунок 2.4 – Окно «Параметры сохранения истории»

Вернувшись в окно «Добавить узел» нажмите на кнопку «Добавить».

2.1.2 Добавление переменных в исторический Modbus узел

Теперь в созданный узел необходимо добавить переменные (Рисунок 2.5).

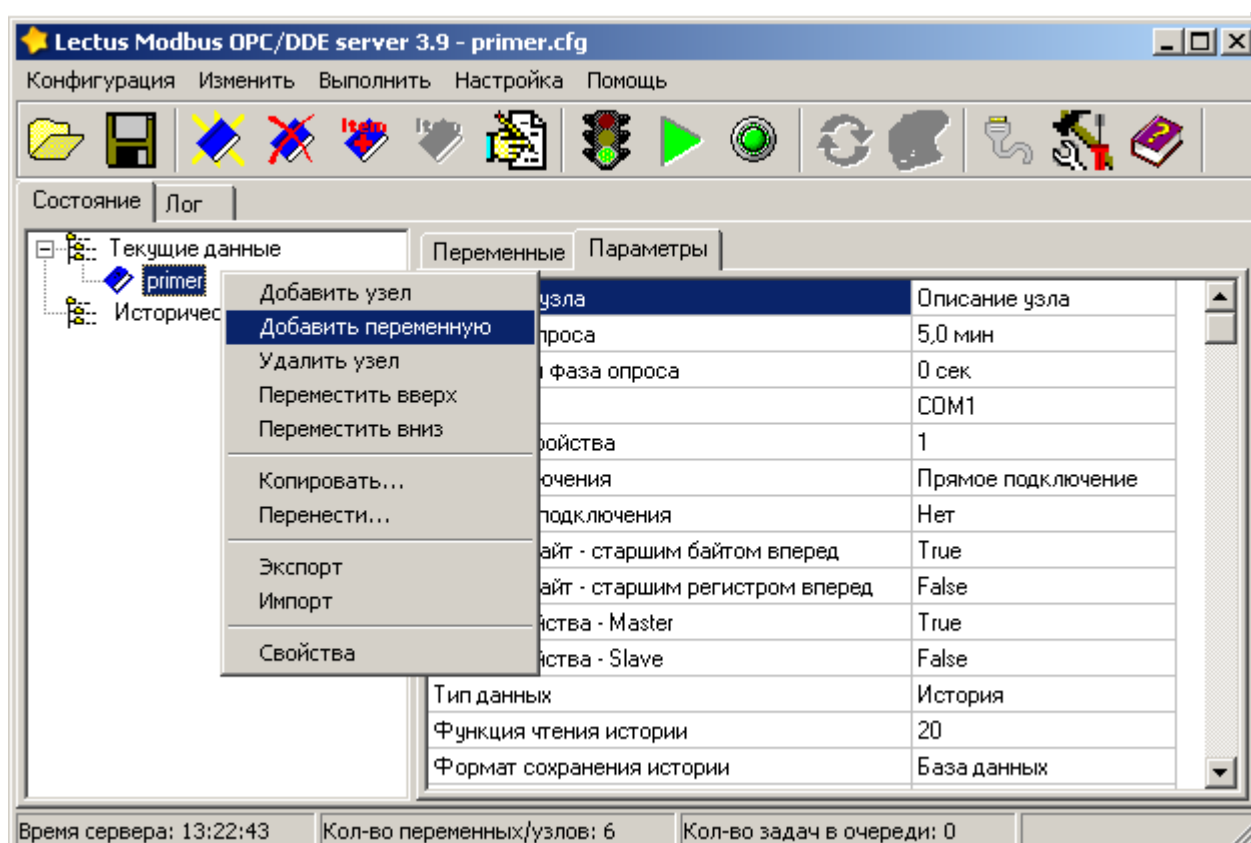


Рисунок 2.5 – Добавление переменных

В появившемся окне «Добавить переменную»: выбрать вкладку «Modbus», где указать:

- «имя переменной» ...;

- «тип данных» – должен совпадать с типом архивируемой переменной добавленной в PLC Configuration проекта CoDeSys;
- «номер переменной» – номер переменной (0, 1, 2, 3...), по порядку объявления в проекте CoDeSys;

Окно «Добавить переменную» представлено на рисунке 2.6.

Рисунок 2.6 – Окно «Добавить переменную»

Не закрывая окно «Добавить переменную»: нажать на кнопку «Настройка...» поля «Дополнительные параметры». В появившемся окне установить галку в поле «Сохранять состояние во встроенной базе данных» (Рисунок 2.7).

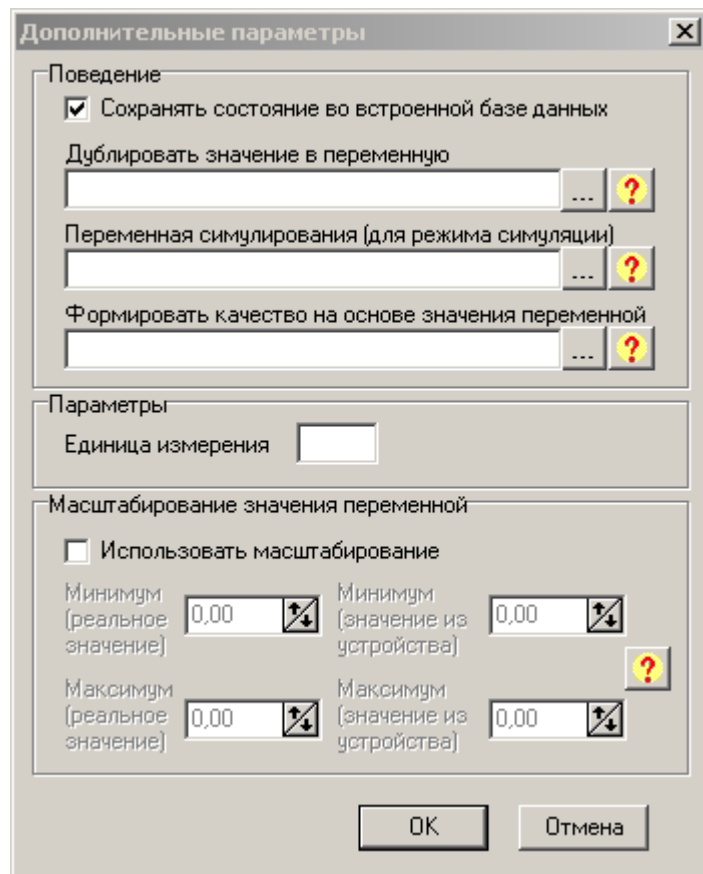


Рисунок 2.7 – Окно «Дополнительные параметры»

Подтвердить изменения, нажав кнопку «ОК». Затем нажать кнопку «Добавить» окна «Добавить переменную».

2.2 Предоставление данных HDA-клиентам

После добавления переменных в исторический узел Modbus, нужно добавить узел «Исторические данные». Его задача предоставлять, прочтенные из файла данные HDA-клиентам, например Scada-системам.

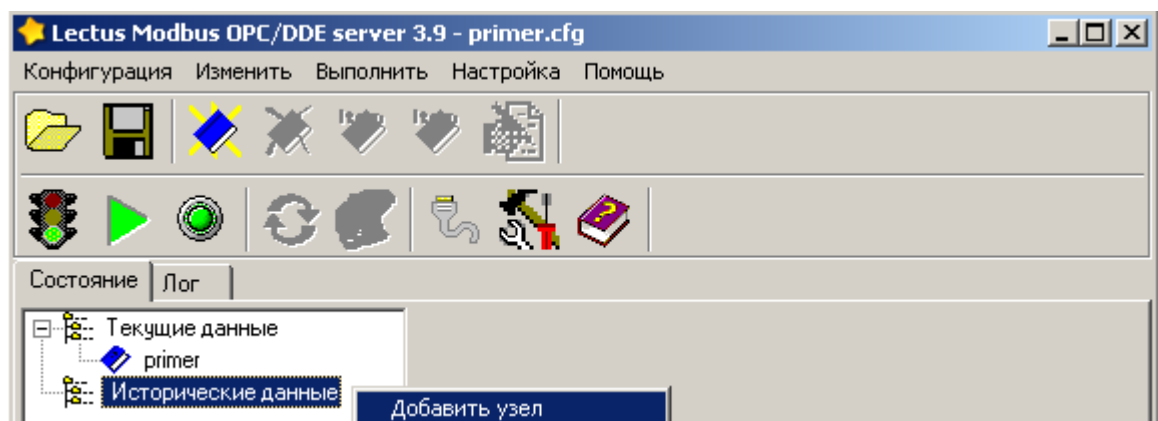


Рисунок 2.8 – Добавление узла к историческим данным

На корневом узле «Исторические данные» нажать правой кнопкой мыши и в сплывающем меню выбрать «Добавить узел».

В открывшемся окне указать имя узла.

Добавить переменные в созданный узел (Рисунок 2.9).

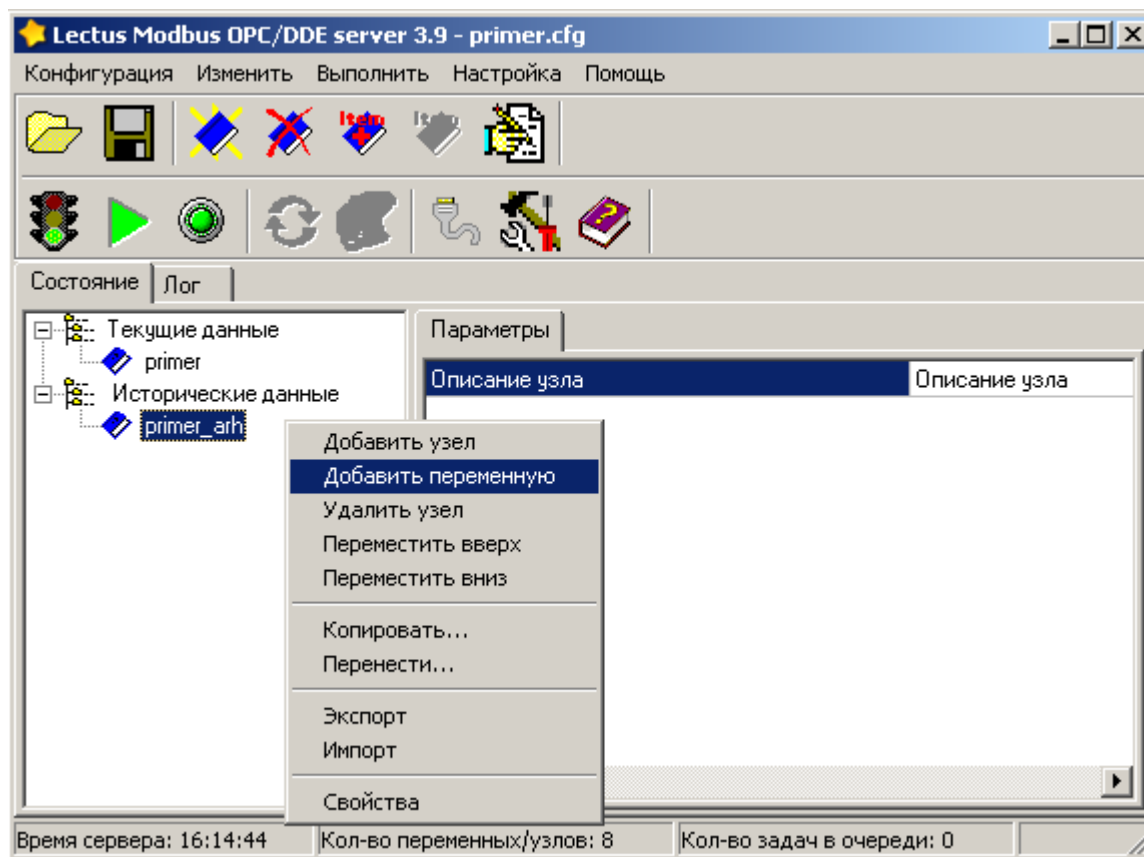


Рисунок 2.9 – Добавление переменных к историческим данным

Создаваемая переменная должна принимать значение переменной хранящейся в архиве для последующей его передачи в HDA-клиент. Это значение переменная должна получить из внутренней базы данных, где она привязывается к исходной (переменной архива).

Для реализации этого в появившемся окне «Добавить переменную» перейти на вкладку «SQL», где указать:

- «имя переменной» - ...;
- «период опроса» - период записи значения в переменной из внутренней базы;
- «тип переменной» - тип SQL-переменной должен совпадать с типом соответствующей переменной, значение которой необходимо передать из Базы данных Lectus, т. е с типом исходной архивируемой переменной;

- в качестве SQL скрипта указать «Переменная», затем нажать на кнопку «...», появиться окно «Выбор переменной». Выбрать переменную, значение которой необходимо передавать в HDA-клиент (Рисунок 2.10).

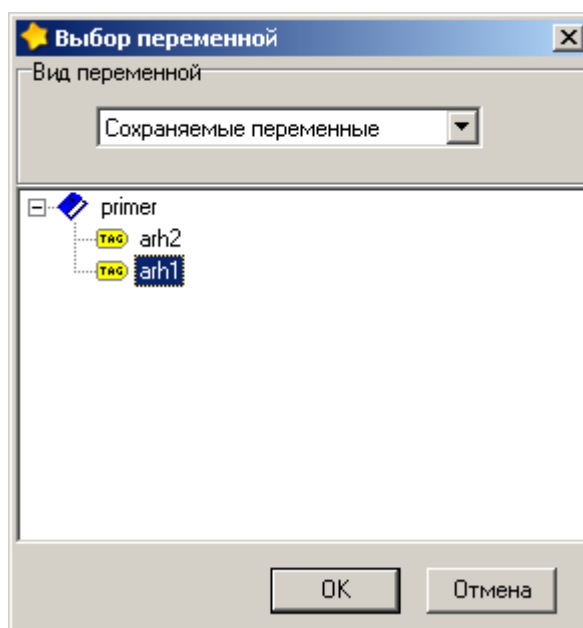


Рисунок 2.10 – Окно «Выбор переменной»

Окно «Добавить SQL переменную» представлено на рисунке 2.11.

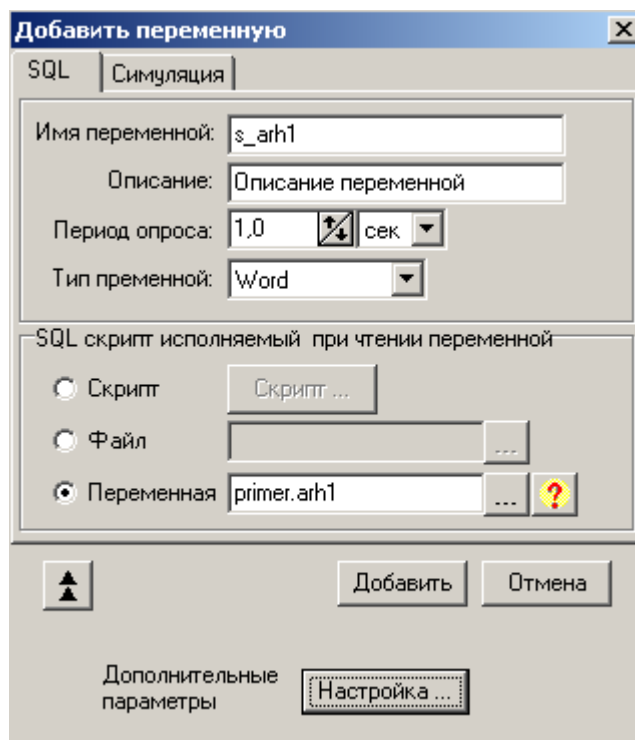


Рисунок 2.11 – Окно «Добавить SQL переменную»

Нажать кнопку «Добавить». Переменная создана. Аналогичную операцию проделать с остальными переменными.

После добавления SQL переменных к историческим данным настройка Lectus OPC закончена, именно к этим переменным будут обращаться HDA - клиенты.

3 Чтение исторических данных HDA-клиентом

В качестве HDA-клиента в данном проекте выступает MatrikonOPC HDA Explorer. Данная программа бесплатна и доступна на сайте производителя: <http://www.matrikon.com/>.

Для настройки MatrikonOPC HDA Explorer на чтение переменных нужно:

- 1) В главном окне программы выбрать HDA OPC-сервер – «Lectus.OPC.1» (Рисунок 3.1);

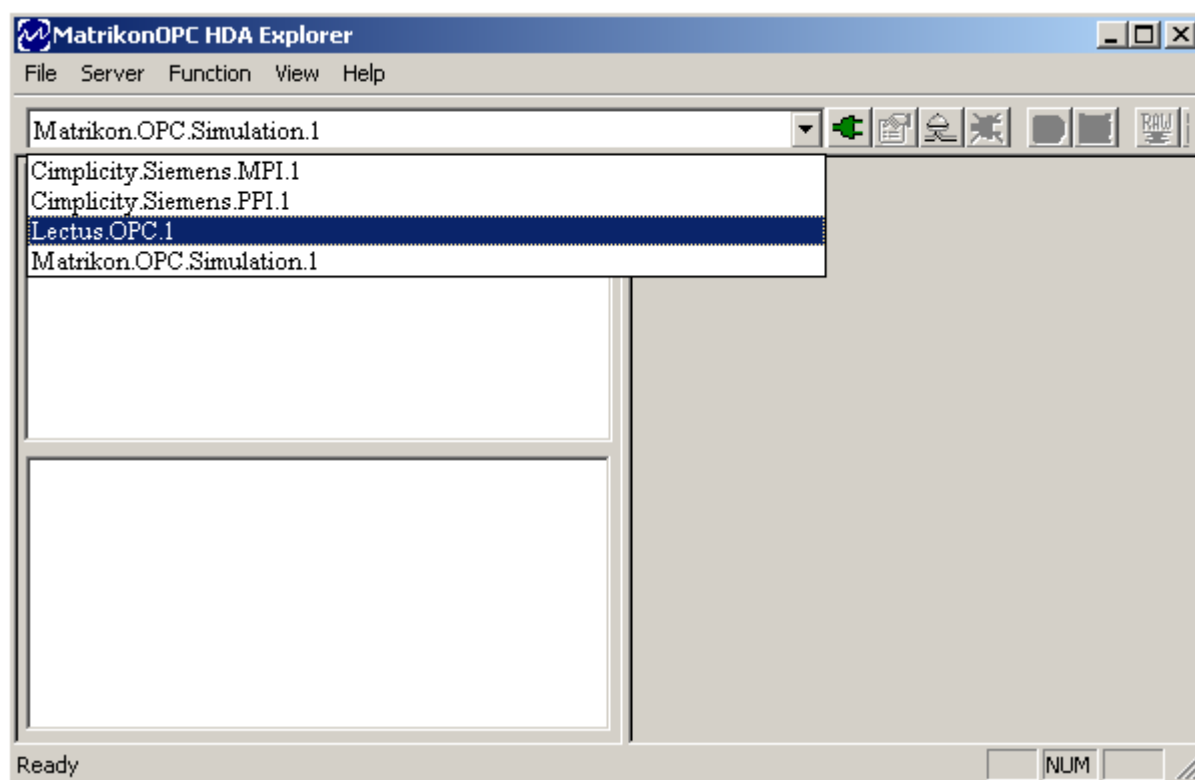


Рисунок 3.1 – Добавление OPC HAD-сервера в MatrikonOPC HDA Explorer

- 2) Соединиться с выбранным сервером нажав «Connect» вкладки «Server» или соответствующую кнопку панели инструментов (Рисунок 3.2);

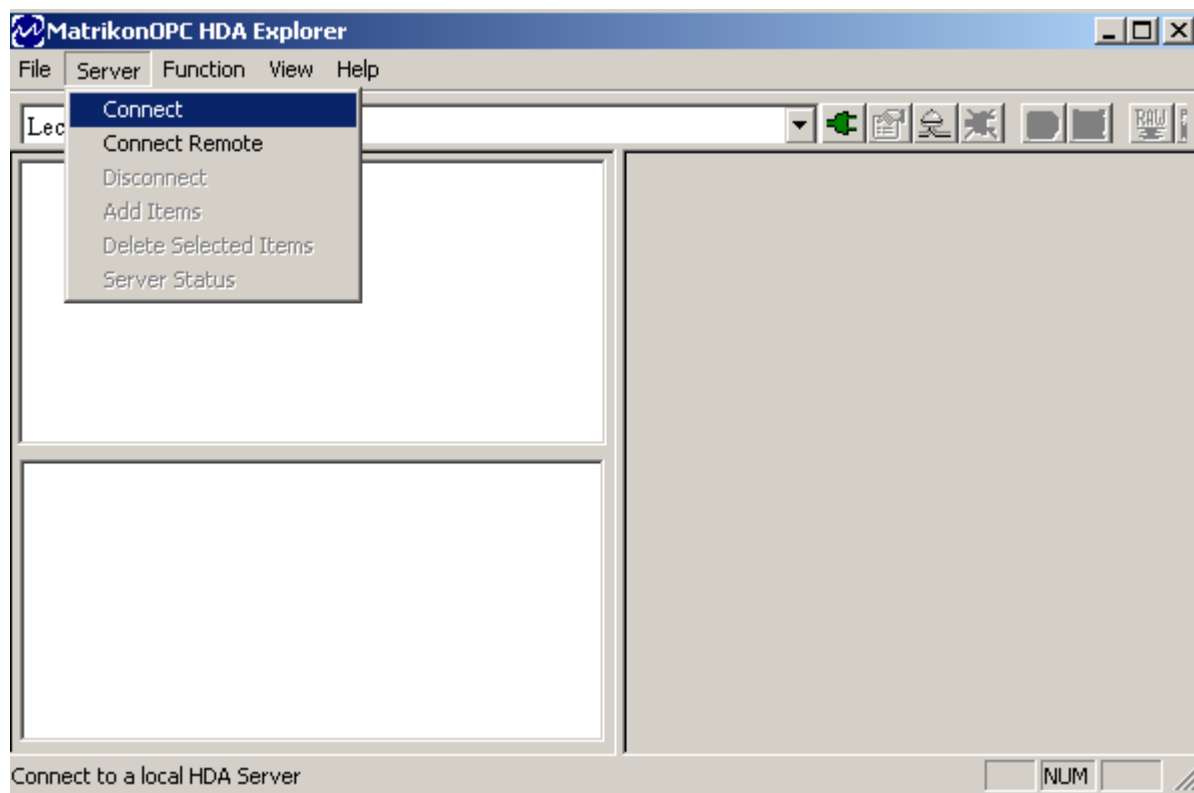


Рисунок 3.2 – Соединение OPC HDA-сервером

- 3) Показать переменные выбранного сервера, нажав «Add Items» контекстного меню (Рисунок 3.3);

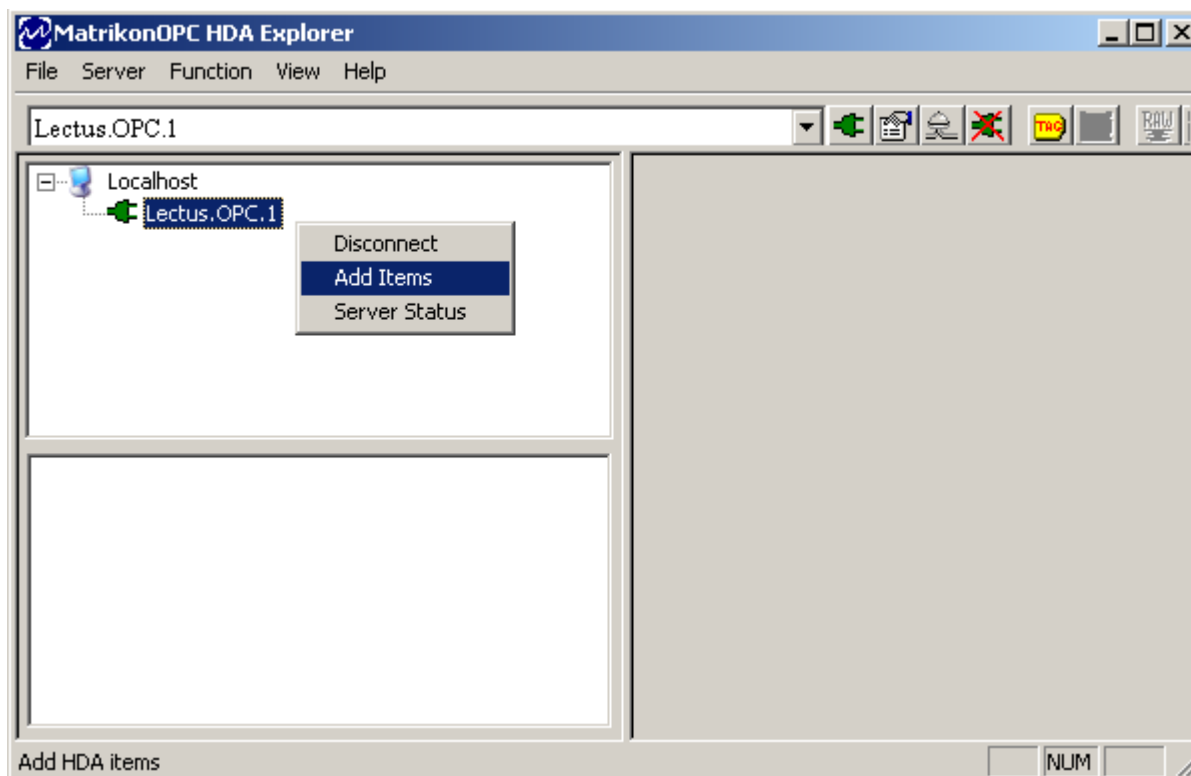


Рисунок 3.3 – Просмотр переменных выбранного OPC HDA - сервера

- 4) Добавить все переменные в проект. Для этого выбрать узел, после чего нажать «Add All Items To List» контекстного меню (Рисунок 3.4) затем нажать кнопку «ОК»;

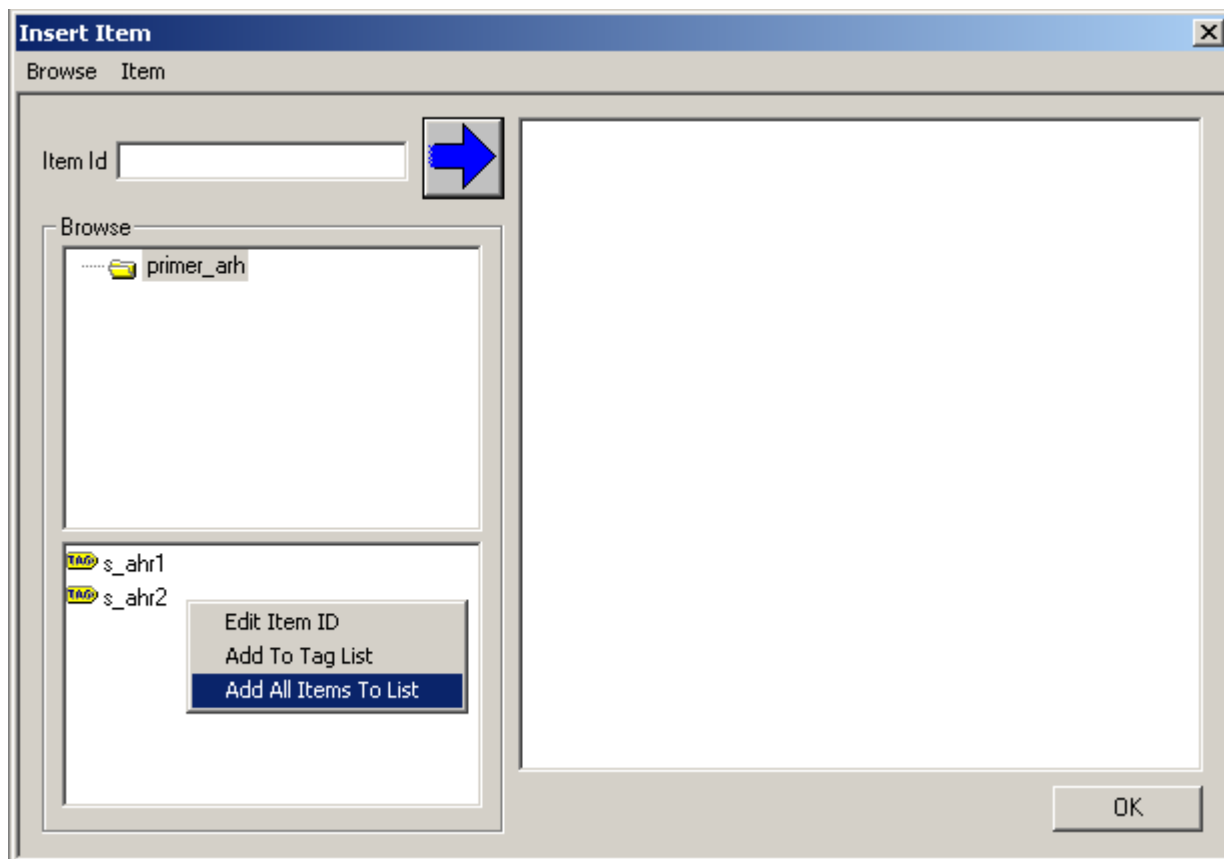


Рисунок 3.4 – Добавление переменных в проект MatrikonOPC HDA Explorer

- 5) Для чтения значений архивируемой переменной выберите её, а затем воспользуйтесь одной из функций чтения в MatrikonOPC HDA Explorer. Например «Read Raw» (Рисунок 3.5);

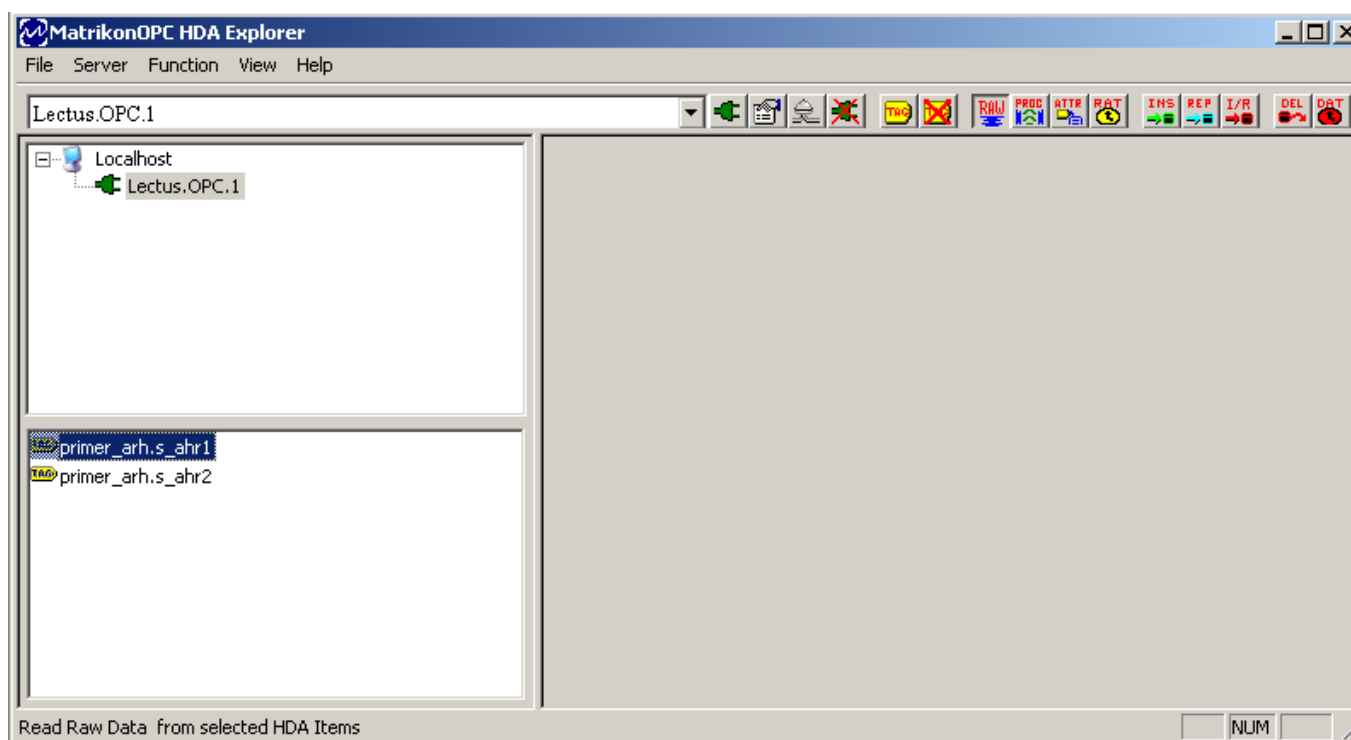


Рисунок 3.5 – Чтение архивируемых переменных в MatrikonOPC HDA Explorer

б) В появившемся окне необходимо задать временной интервал чтения переменной и задать ограничение по выводимому значению (Рисунок 3.6); Для чтения значений архивируемой переменной выберите её, а затем воспользуйтесь одной из функций чтения в MatrikonOPC HDA Explorer. Например «Read Raw» (Рисунок 3.6).

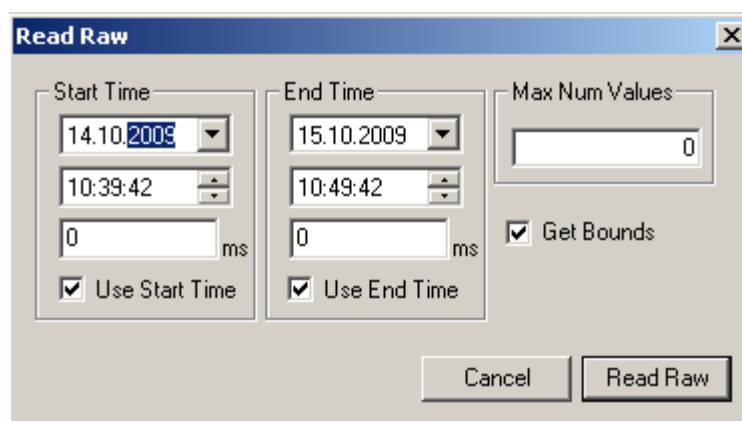


Рисунок 3.6 – Окно «Read Raw»

После нажатия кнопки «Read Raw» появиться в окне программы появиться вкладка со значениями выбранной переменной.

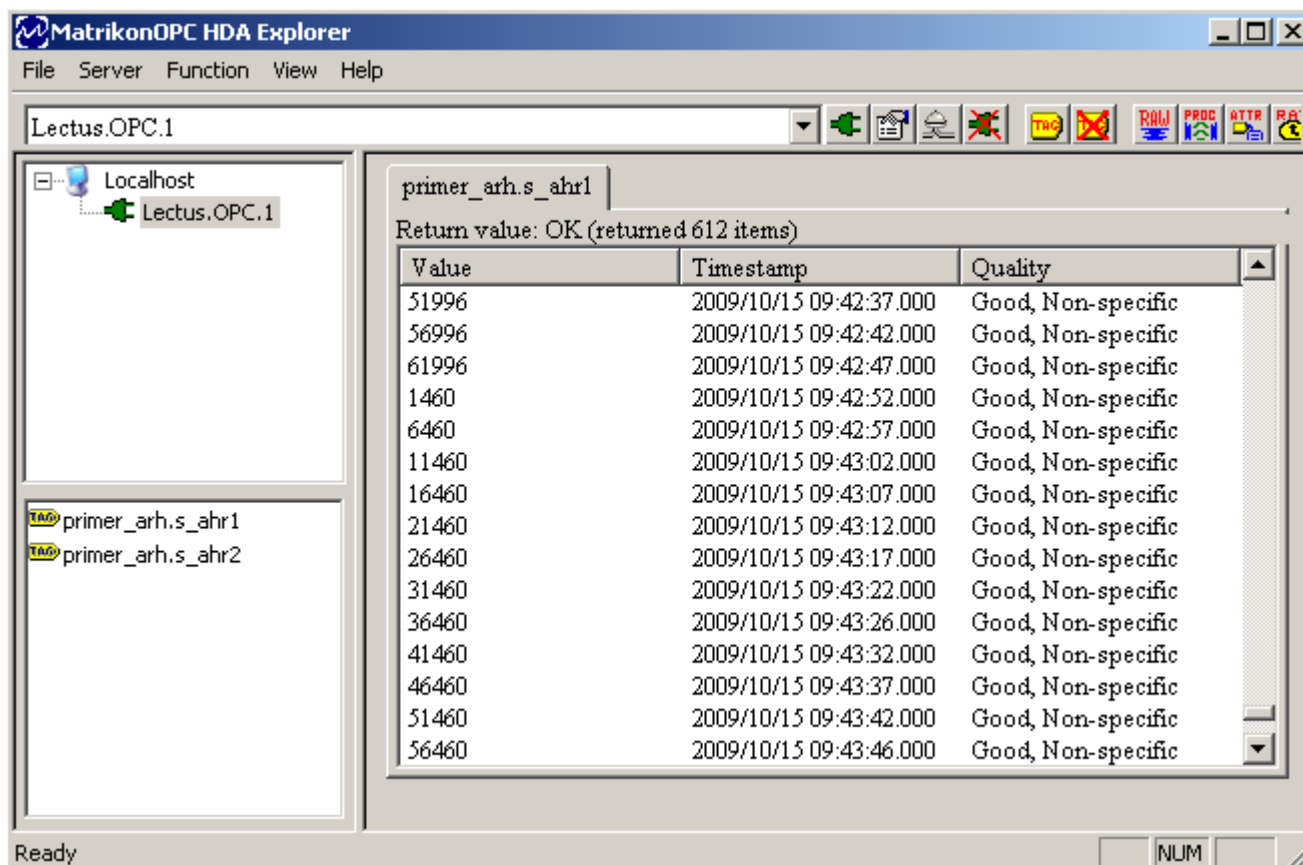


Рисунок 3.7 – Представление значений исторической переменной в MatrikonOPC HDA Explorer

На этом работа с MatrikonOPC HDA Explorer закончена.

Приложение

3.1 Форматы запросов и ответов при обмене по 20-ой функции Modbus.

Данная информация носит справочный характер.

Формат запроса:

Function Code	0x14	Byte	Код функции
Byte count	0x07	Byte	Кол-во байт, следующих ниже
Referens Type	0x06	Byte	Подфункция (в нашем случае константа =6)
Hi File number	-	Byte	Старший Байт номера требуемого файла
Lo File number	-	Byte	Соответственно младший
Hi Rec addr	-	Byte	Старший Байт адреса записи в файле
Lo Rec addr	-	Byte	Соответственно младший
Hi Rec num	-	Byte	Старший Байт количества запрашиваемых записей
Lo Rec num	-	Byte	Соответственно младший

Ответ в случае успеха:

Function Code	0x14	Byte	Код функции
Byte count	0x07	Byte	Кол-во байт, следующих ниже
Byte count	0x07	Byte	Кол-во байт, следующих ниже (необходимо по стандарту)
Referens Type	0x06	Byte	Подфункция (в нашем случае константа =6)
Data	-	Byte*Rec_num*Amount_b	Собственно данные длиной Rec num из (запроса), умноженные на длину одной записи из конфигурации.

File number – может принимать значение от 0x0000 до 0xffff

Rec number – может принимать значение от 0x0000 до 0xffff

Rec addr – может принимать значение 0x0000-0xffff

Важно! При запросе с адресом Recaddr=0xffff происходит удаление файла.

При ошибке возвращается стандартный код ошибки ModBus, то есть 0x02.