**Конфигурация Complex Systems Modbus OPC Server**

 Сервер может быть запущен либо в режиме службы, либо в режиме обычного COMсервера. Для запуска сервера в режиме службы наберите в командной строке **uc\_modbus\_opc –service**

Для запуска а режиме обычного COM-сервера наберите **uc\_modbus\_opc -regserver**

 Настройки сервера хранятся в файле **uc\_modbus\_opc.xml**.

 При старте OPC-сервера этот файл используется для инициализации и если при этом произошли какие-либо ошибки, то сообщения об этом будут записаны в файл **uc\_modbus\_opc.log**. Конфигурационный файл имеет следующую структуру:

<?xml version="1.0" encoding="utf-8"?>

<CONFIG>

 <UC\_MODBUS\_OPC>

 <PORT ... >

 <DEVICE ... >

 <ITEM ... />

 <ITEM ... />

 ...

 <REQUEST ... />

 <REQUEST ... />

 ...

 </DEVICE>

 <DEVICE ... />

 ...

 </PORT>

 <PORT ... />

 ...

 </UC\_MODBUS\_OPC>

</CONFIG>

 Имя корневого тэга (здесь, **<CONFIG>**) не имеет значения, но в нем должен быть дочерний тэг **<UC\_MODBUS\_OPC>**, в котором и храниться конфигурация OPC-сервера. В качестве атрибутов в этом тэге можно указать:

TIMEBASE - гранулярность обновлений OPC-тэгов из устройств в миллисекундах,

 по умолчанию 125;

MIN\_UPDATERATE - минимальный период опроса OPC-групп, разрешенный OPC-клиентам

 в миллисекундах, по умолчанию 125;

CONTROLTAG - имя OPC-тэга, используемого для управления OPC-сервером; если атрибут отсутствует, этот тэг не создается.

 В OPC-тэг, имя которого задается атрибутом CONTROLTAG, OPC-клиент может записывать следующие команды:

|  |  |
| --- | --- |
| Команда  | Значение  |
| LOCKSERVER  | Выделяет эксклюзивный доступ на чтение/запись устройств, опрашиваемых OPCсервером, OPC-клиенту, который записал команду. При этом другой OPC-клиент сможет добавлять тэги в OPC-группы, и значения этих тэгов все же будут опрашиваться асинхронно. Запретить асинхронный опрос можно командой PAUSEASYNC. Данная блокировка будет отменена автоматически, если вызвавший ее клиент отключится от OPC-сервера. ВНИМАНИЕ. Если клиент, вызвавший команду LOCKSERVER, отключится некорректно (процесс клиента будет просто выгружен и интерфейсы OPC-сервера не будут освобождены), то в этом случае отмены команды LOCKSERVER не произойдет.  |
| LOCKSERVERPUSH  | Аналогично LOCKSERVER, но при этом запоминается состояние опроса портов (включен/отключен). При отмене команды автоматически будет восстановлено состояние опроса портов, которое было актуально на момент получения команды LOCKSERVERPUSH.  |
| WAITLOCKSERVER  | Аналогично LOCKSERVER, но при этом, если доступ выделен другому клиенту, то происходит ожидание момента, когда этот клиент освободит доступ к OPCсерверу.  |
| WAITLOCKSERVERPUSH  | Аналогично LOCKSERVERPUSH, но при этом, если доступ выделен другому клиенту, то происходит ожидание момента, когда этот клиент освободит доступ к OPC-серверу.  |
| UNLOCKSERVER  | Отменяет действие команды LOCKSERVER.  |
| PAUSEASYNC  | Приостанавливает асинхронный опрос всех портов. ВНИМАНИЕ! Пользователь может возобновить опрос из апплета Services приостановив и возобновив сервис uc\_modbus\_opc (если Modbus OPC Server зарегистрирован как сервис Windows).  |
| RESUMEASYNC  | Отменяет действие команды PAUSEASYNC.  |

 Здесь и далее, если атрибут отсутствует, то используется значение по умолчанию (если это возможно). Внутри **<UC\_MODBUS\_OPC>** может содержаться произвольное количество тэгов **<PORT>**.

 Тэг **<PORT>** описывает конкретный последовательный порт и имеет следующие атрибуты:

|  |  |
| --- | --- |
| PORT  | - имя последовательного порта;  |
| PORTRATE  | - скорость передачи данных (бод в секунду);  |
| DATABITS  | - количество бит данных;  |
| STOPBITS  | - количество стоп-бит (1, 1.5 или 2);  |
| PARITY  | - четность (EVEN, ODD, MARK, SPACE, NO);  |
| CONTROLTAG  | - имя OPC-тэга, используемого для управления портом; если атрибут отсутствует, этот тэг не создается.  |

 В OPC-тэг, имя которого задается атрибутом CONTROLTAG, OPC-клиент может записывать следующие команды:

|  |  |
| --- | --- |
| Команда  | Значение  |
| PAUSEASYNC  | Приостанавливает асинхронный опрос порта. ВНИМАНИЕ! Пользователь может возобновить опрос из апплета Services приостановив и возобновив сервис uc\_modbus\_opc (если Modbus OPC Server зарегистрирован как сервис Windows). Также эта команда выполняет ожидание завершения выполняющейся операции с портом (при наличии таковой).  |
| RESUMEASYNC  | Отменяет действие команды PAUSEASYNC.  |
| WAITPORT  | Выполняет ожидание завершения выполняющейся операции с портом (при наличии таковой).  |

 Внутри **<PORT>** может содержаться произвольное количество тэгов **<DEVICE>**, описывающих устройства в сети MODBUS, подключенной к данному последовательному порту. Тэг **<DEVICE>** описывает конкретное физическое устройство MODBUS и имеет следующие атрибуты:

DEVICEADDR - MODBUS-адрес устройства, нет значения по умолчанию;

RETRY - число повторов при запросах к устройству, по умолчанию 1; TIMEOUT - таймаут запроса к устройству (мс), по умолчанию 1000.

Внутри **<DEVICE>** может содержаться произвольное количество тэгов **<ITEM>** и **<REQUEST>**.

Тэг **<ITEM>** описывает элемент в иерархии OPC-сервера. Его атрибуты:

|  |  |
| --- | --- |
| OPCID  | - имя OPC-тега, нет значения по умолчанию;  |
| MODBUSADDR  | - адрес регистра MODBUS, нет значения по умолчанию;  |
| MODBUSTYPE  | - тип данных элемента (возможные значения в таблице ниже), по умолчанию VT\_I2;  |
| ACCESSRIGHTS  | - права доступа к тэгу (R - чтение, W - запись, RW - чтение/запись), по умолчанию R;  |
| UPDATERATE  | - период опроса регистра (мс), по умолчанию 1000.  |
| FUNCTION  | - команда MODBUS, используемая для выполнения запроса (3 или 4),  |
|   |  нет значения по умолчанию, можно не указывать, если элемент входит в запрос;  |

|  |  |
| --- | --- |
| MODBUSTYPE  | Значение  |
| VT\_I2  | Целое число со знаком, 2 байта  |
| VT\_UI2  | Целое число без знака, 2 байта  |
| VT\_I2\_O  | Целое число со знаком, 2 байта, кодируется со сдвигом на 8000h (-32768 кодируется нулем, 0 кодируется 8000h, 32767 кодируется FFFFh)  |
| VT\_R4  | Число с плавающей запятой, 4 байта (2 регистра MODBUS)  |
| VT\_I4  | Целое число со знаком, 4 байта (2 регистра MODBUS)  |
| VT\_UI4  | Целое число без знака, 4 байта (2 регистра MODBUS)  |
| VT\_I4\_INV  | Целое число со знаком, 4 байта (2 регистра MODBUS, регистр со старшим словом числа по младшему адресу)  |
| VT\_UI4\_INV  | Целое число без знака, 4 байта (2 регистра MODBUS, регистр со старшим словом числа по младшему адресу)  |
| VT\_I6  | Целое число со знаком, 6 байт (3 регистра MODBUS)  |
| VT\_UI6  | Целое число без знака, 6 байт (3 регистра MODBUS)  |
| VT\_I6\_INV  | Целое число со знаком, 6 байт (3 регистра MODBUS, регистр со старшим словом числа по младшему адресу)  |
| VT\_UI6\_INV  | Целое число без знака, 6 байт (3 регистра MODBUS, регистр со старшим словом числа по младшему адресу)  |
| VT\_I8  | Целое число со знаком, 8 байт (4 регистра MODBUS)  |
| VT\_UI8  | Целое число без знака, 8 байт (4 регистра MODBUS)  |
| VT\_I8\_INV  | Целое число со знаком, 8 байт (4 регистра MODBUS, регистр со старшим словом числа по младшему адресу)  |
| VT\_UI8\_INV  | Целое число без знака, 8 байт (4 регистра MODBUS, регистр со старшим словом числа по младшему адресу)  |
| VT\_I6\_BCD4  | Целое со знаком, 6 байт (3 регистра MODBUS). В каждом регистре храниться 4 десятичных разряда. Значение равно REG1 + REG2 \* 104 + REG3 \* 108, где REG1 – регистр с младшим адресом  |
| VT\_UI6\_BCD4  | Целое без знаком, 6 байт (3 регистра MODBUS). В каждом регистре храниться 4 десятичных разряда. Значение равно REG1 + REG2 \* 104 + REG3 \* 108  |
| VT\_I8\_BCD4  | Целое со знаком, 8 байт (4 регистра MODBUS). В каждом регистре храниться 4 десятичных разряда. Значение равно REG1 + REG2 \* 104 + REG3 \* 108 + REG4 \* 1012  |
| VT\_UI8\_BCD4  | Целое без знаком, 8 байт (4 регистра MODBUS). В каждом регистре храниться 4 десятичных разряда. Значение равно REG1 + REG2 \* 104 + REG3 \* 108 + REG4 \* 1012  |
| VT\_DATE\_IEC87054  | Время в формате МЭК 870-5-4 (4 регистра MODBUS). REG1LO – год (0-99), REG2HI – месяц (1-12), REG2LO – день (1-31), REG3HI – часы (0-23), REG3LO – минуты (0-59), REG4 – миллисекунды (0-59999).  |

 Если тип данных тэга – это величина, хранимая более чем в одном регистре MODBUS, то для чтения такого значения необходим **<REQUEST>**, описывающий чтение необходимых регистров.

В именах элементов в качестве разделителя используется точка.

Тэг **<REQUEST>** описывает запросы. Запрос – это чтение нескольких регистров MODBUS с последовательными адресами за одну MODBUS-команду. Запрос имеет следующие атрибуты:

|  |  |
| --- | --- |
| FUNCTION  | - команда MODBUS, используемая для выполнения запроса (3 или 4),  |
|   |  нет значения по умолчанию;  |
| REG0  | - адрес первого MODBUS-регистра в запросе, нет значения по умолчанию;  |
| REGCOUNT  | - количество регистров в запросе, нет значения по умолчанию;  |
| UPDATERATE  | - периодичность выполнения запроса (мс), по умолчанию 1000;  |
| ACTIVE  | - активность запроса (YES или NO), по умолчанию YES.  |

 После выполнения запроса, значения считанных регистров помещаются в кэш значений OPC-сервера. При этом обновляются значения каждого из элементов OPC, соответствующих считанным регистрам и адресуемому устройству.

 Активные запросы (**ACTIVE=”YES”**) выполняются периодически с указанным периодом, независимо от того затребованы эти данные OPC-клиентами или нет. Неактивные запросы выполняюися только, когда OPC-клиенты запросили соответвтующие элементы (эти элементы активны и входят в активную группу или было затребовано единовременное чтение из устройства). Такой подход позволяет оптимально настроить обмен с физическими устройствами. Если элемент, описанный в конфигурации, не входит ни в один из приведенных запросов, то при его считывании из устройства будет создан временный запрос на чтение одного регистра. Причем в качестве **UPDATERATE** и **FUNCTION** будут использованы значения, приведенные в описании этого элемента.