

Описание библиотеки OwenModbuSlave

(Codesys 2.3)

Руководство по применению

Версия: 2.3.9.4

Введение

Библиотека OwenModbusSlave применяется для настройки ОВЕН ПЛК в режиме Modbus RTU Slave. В отличие от настройки прибора через *Конфигурацию ПЛК* данный способ предоставляет более широкий функционал и является удобным при работе с большим количеством переменных.

1. Установка дополнительных библиотек для проекта

В системе CoDeSys все файлы библиотек функциональных блоков имеют расширения *.lib и находятся в папке Library – она расположена по месту размещения основной программы на диске компьютера (чаще всего в папке C:\Program Files\3S Software\CoDeSys V2.3\Library). Для подключения новых библиотек к проекту сначала соответствующие файлы переписываются пользователем в ту же папку, где находятся все остальные библиотеки.

Чтобы увидеть какие библиотеки уже были раньше подключены к проекту и установить дополнительные, используется Менеджер библиотек – его можно открыть из главного меню CoDeSys командами **Окно ► Менеджер библиотек** (или в менеджере проекта на закладке «Ресурсы» открывается папка **Менеджер библиотек**), рис. 1.1 (в окне вверху показан список установленных библиотек).

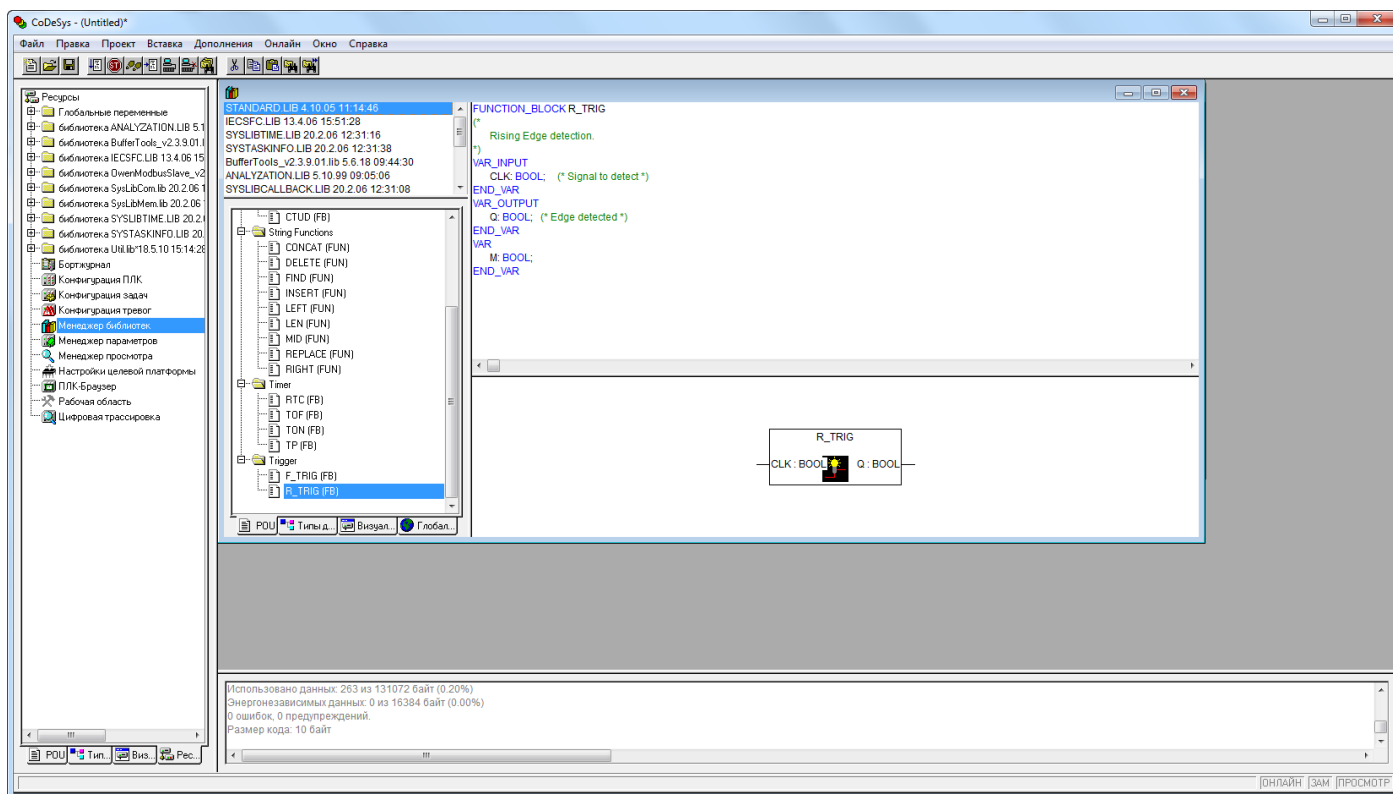


Рис. 1.1. Окно открытой папки «Менеджер библиотек»

Установка дополнительных библиотек выполняется из главного меню последовательным выбором команд: **Вставка ► Добавить библиотеку** (рис. 1.2), в открывшемся окне папки Library (рис. 1.3) выделяется файл с именем нужной библиотеки и выполняется команда **Открыть**.

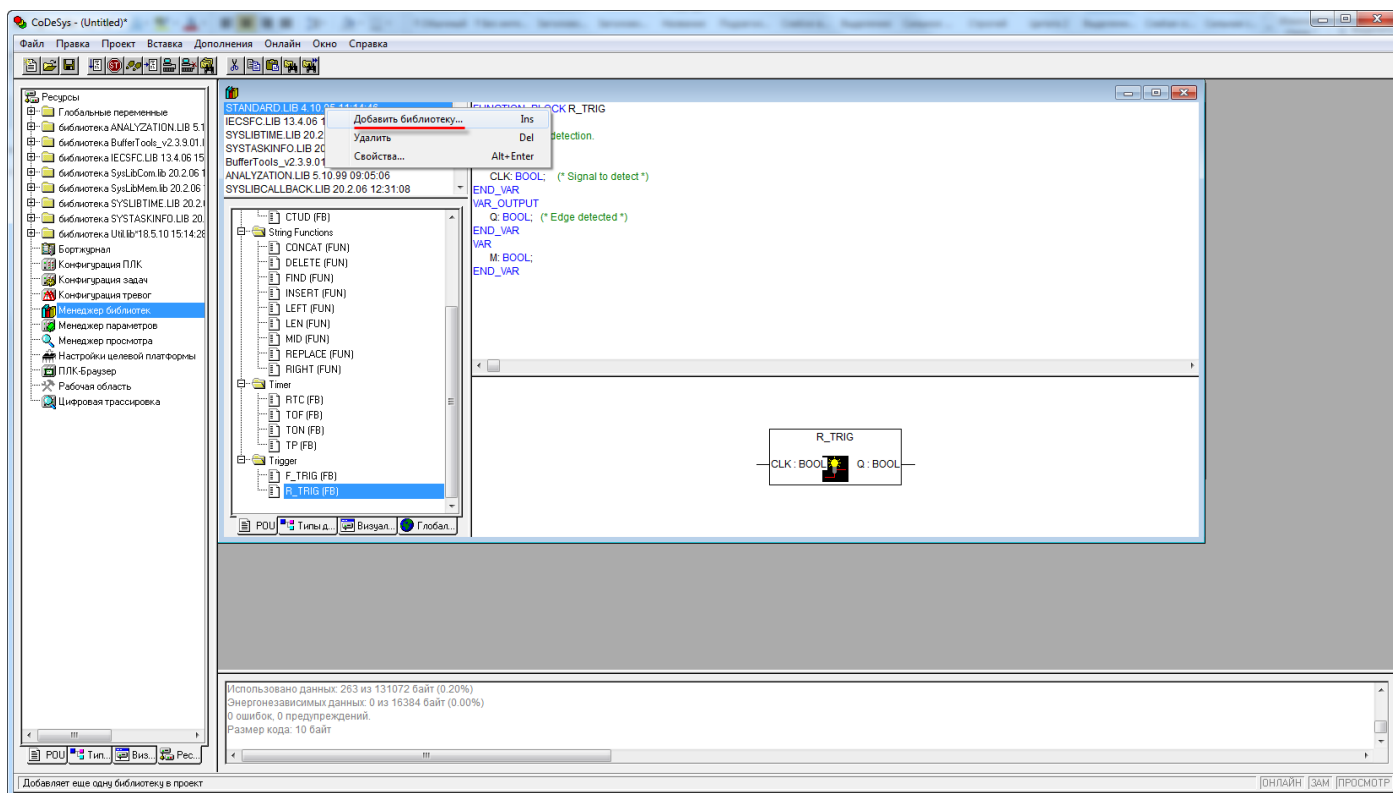


Рис. 1.2 – Команда «Добавить библиотеку»

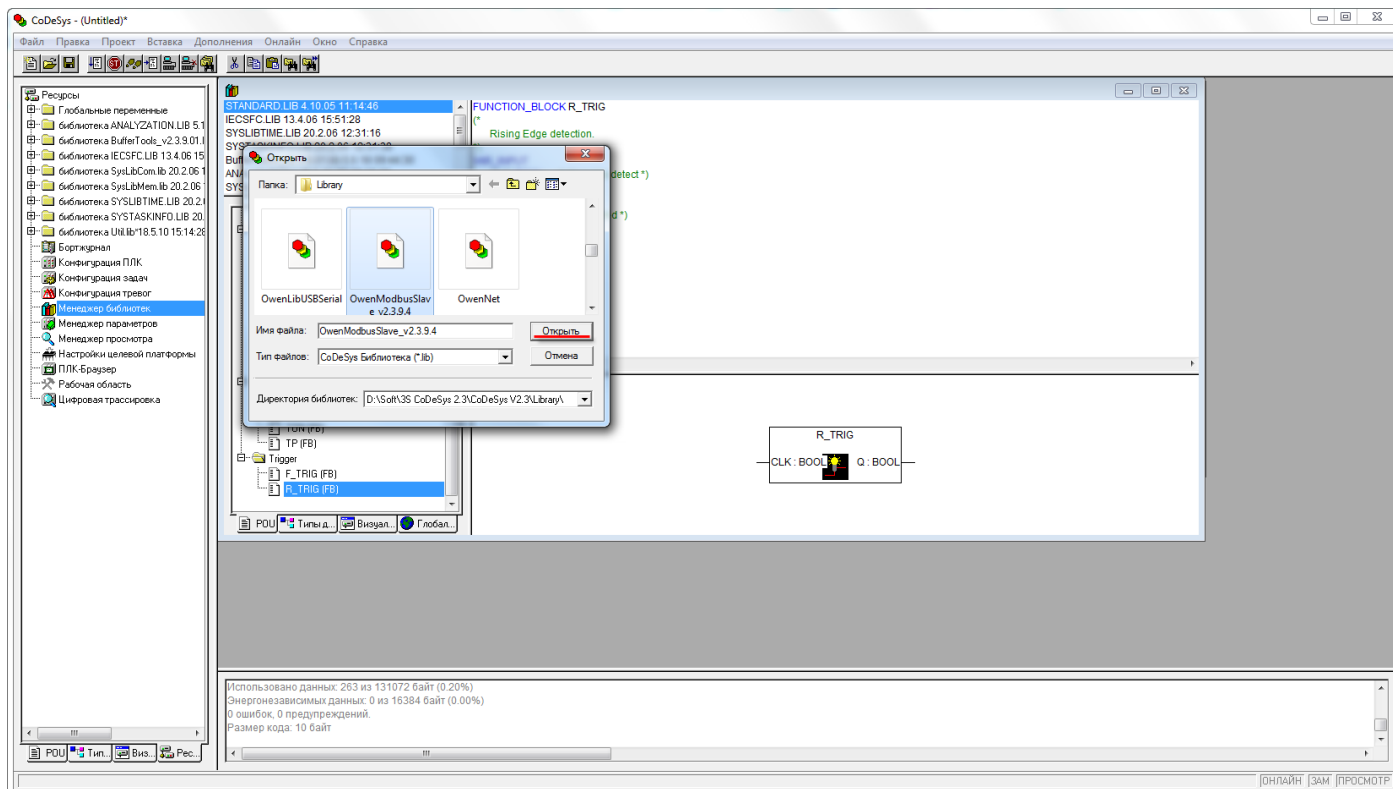


Рис. 1.3 – Окно для выбора подключаемой дополнительной библиотеки к проекту

Теперь в перечне доступных для проекта библиотек появится установленная библиотека – **OwenModbusSlave_v2.3.9.4.lib**. Для просмотра состава и свойств функциональных блоков, выбирается нужная библиотека – при этом появится папка с блоками (на рис. 1.1 справа дана краткая справочная информация по использованию компонентов).

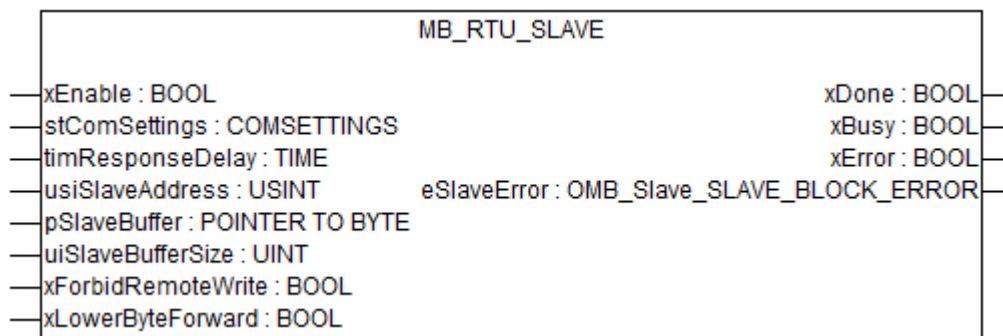
Подобным образом можно добавлять любую из поддерживаемых библиотек.

ПРИМЕЧАНИЕ:

1. Для функционирования библиотеки **OwenModbusSlave_v2.3.9.4.lib** также необходима дополнительная библиотека **BufferTools_v2.3.9.01.lib**.
2. По умолчанию доступен только стандартный набор библиотек. Для каждого нового проекта проводится присоединение новых библиотек индивидуально, по мере необходимости.
3. У функциональных блоков, разработанных для контроллеров фирмы ОВЕН, режим симуляции (**Simulation Mode**) не предусмотрен. Отладка программы возможна только при подключенном контроллере.

2. Описание функциональных блоков библиотеки

Библиотека содержит только один пользовательский функциональный блок – **MB_RTU_SLAVE**.



Блок используется для настройки ПЛК в режиме Slave по протоколу Modbus RTU. Пока вход **xEnable** имеет значение **TRUE**, slave активен и отвечает на запросы. По заднему фронту входа **xEnable** блок прекращает работу.

ФБ содержит интегрированный блок COM-портом, поэтому в качестве аргументов принимает только сетевые настройки (открывать порт другими блоками для получения дескриптора не требуется).

Поддерживаются следующие функции Modbus: 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0xF, 0x10.

Все области памяти наложены друг на друга (Data Model with only 1 block).

Поддерживается до 125 регистров в запросе.

Название	Тип данных	Описание
Входы		
xEnable	BOOL	Пока вход имеет значение TRUE , блок находится в работе. По заднему фронту происходит прекращение работы slave-устройства.
stComSettings	COMSETTINGS (см. библиотеку SysLibCom)	Настройки COM-порта, на котором функционирует slave.
timResponseDelay	TIME	Задержка перед отправкой ответа master-устройству (значение по умолчанию – 20 мс.
usiSlaveAddress	USINT	Адрес данного устройства (1...247).
pSlaveBuffer	POINTER TO BYTE	Указатель на буфер данных slave-устройства.
uiSlaveBufferSize	UINT	Размер буфера в байтах.
xForbidRemoteWrite	BOOL	Запрет на изменение данных slave-устройства со стороны master-устройства. Если вход имеет значение TRUE , то данные slave-устройства будут доступны только для чтения.
xLowerByteForward	BOOL	Определяет порядок следования байт в буфере Slave-устройства. Значение FALSE (по умолчанию) - little endian, TRUE - big endian.
Выходы		
xDone	BOOL	Флаг «работа блока была завершена».
xBusy	BOOL	Флаг "блок находится в работе".
xError	BOOL	Флаг "произошла ошибка".

eSlaveError	OBM_Slave_SLAVE_BLOCK_ERROR	Код ошибки.
-------------	-----------------------------	-------------

3. Создание проекта с использованием библиотеки

- Добавьте в проект следующие библиотеки:
 - OwenModbusSlave_v2.3.9.4.lib;
 - BufferTools_v2.3.9.01.lib.
- Объявите в программе структуру SLAVE_DATA, которая будет представлять собой карту регистров slave-устройства. При наполнении структуры данными требуется соблюдать выравнивание (см. п. 4).

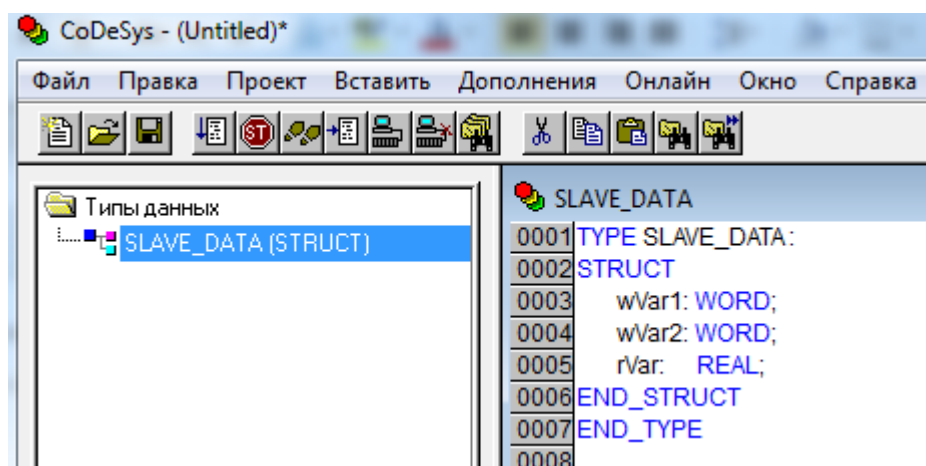


Рис. 3.1 – Создание структуры с картой регистров slave-устройства

Данная структура соответствует следующей карте регистров:

Переменная	Номер регистра	Комментарий
wVar1	WORD	Поддерживаются функции как для Input, так и для Holding регистров.
wVar2	WORD	
rVar	REAL	

- Объявите в программе экземпляры структур SLAVE_DATA и COMSETTINGS, а также экземпляр ФБ MB_RTU_SLAVE:

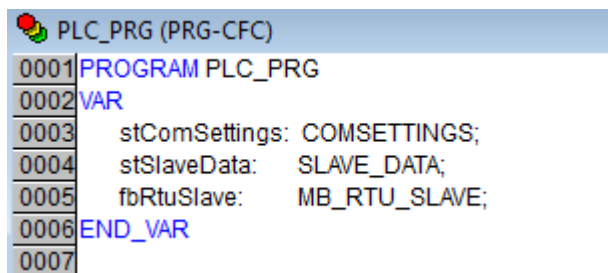


Рис. 3.2 – Объявление переменных программы

4. Код программы будет выглядеть следующим образом:

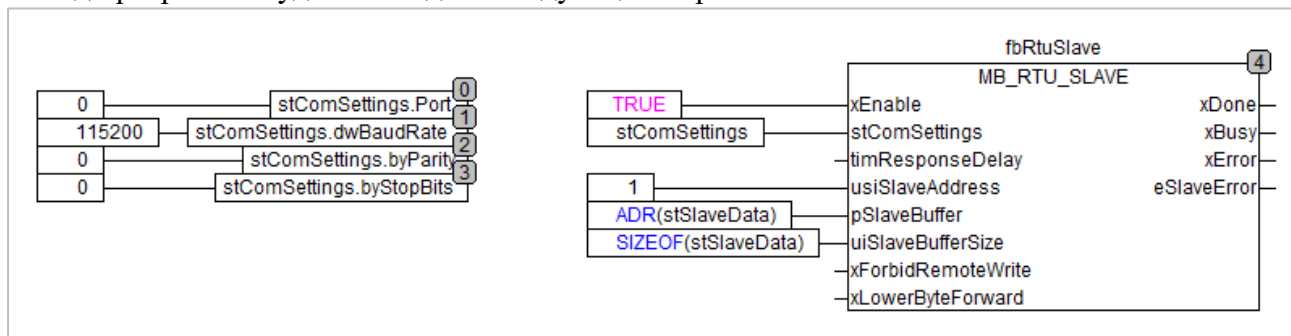


Рис. 3.3 – Код программы

В рамках примера используется ПЛК110 [M02]. Идентификатор его порта RS485-1 имеет значение «0». Информация об идентификаторах портов других ПЛК приведена в документации на ПЛК. Используемые сетевые настройки: 115200-8-N-1. Расшифровка значений для переменных byParity и byStopBits структуры COMSETTINGS приведены в документации на библиотеку SysLibCom.

5. Подключитесь к ПЛК и загрузите проект. Опросите его нужным вам master-устройством.

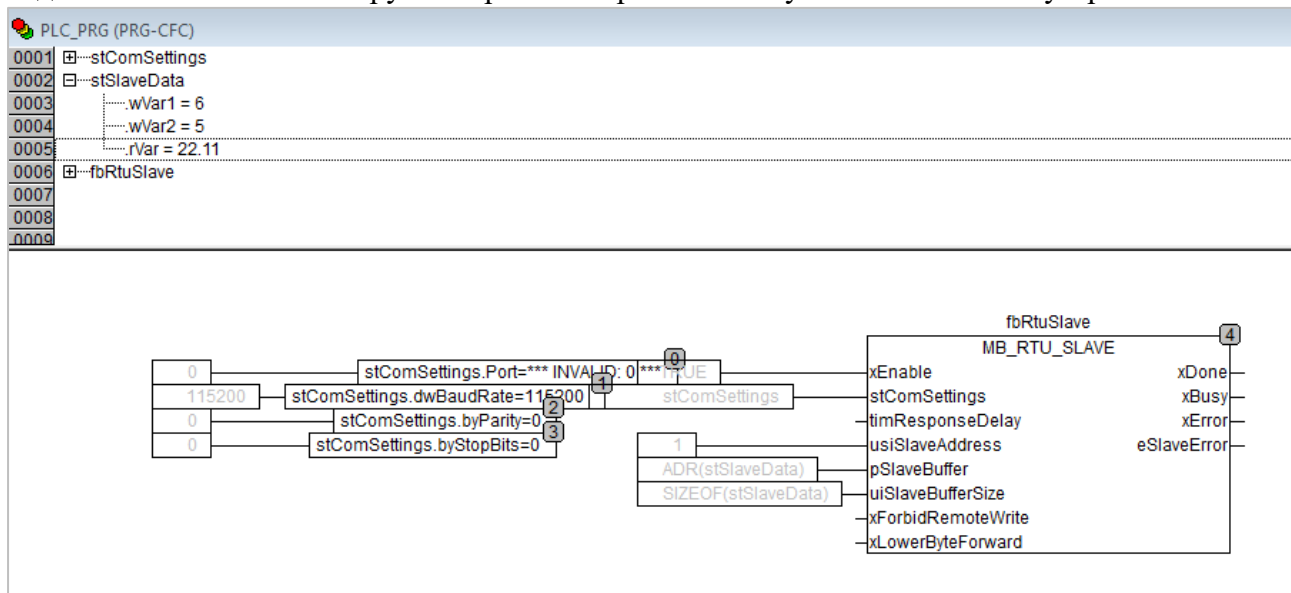


Рис. 3.4 – Запуск проекта

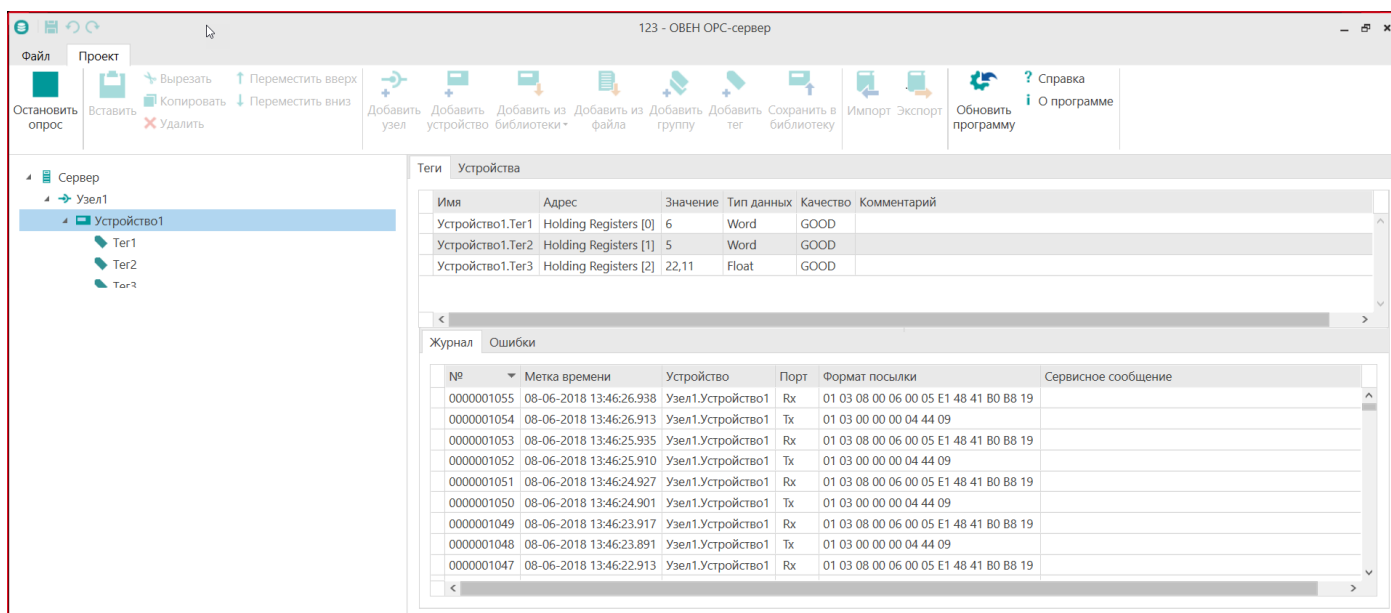


Рис. 3.5 – Опрос ПЛК с помощью OPC-сервера

4. Выравнивание данных в структурах

При работе с библиотекой удобно формировать карту регистров ПЛК с помощью создания структуры. Для корректного размещения элементов структуры требуется соблюдать выравнивание. Для этого необходимо следовать описанным ниже правилам:

- Не использовать переменные типа BOOL, BYTE, SINT, USINT (при необходимости можно работать с логическими значениями с помощью переменных типа WORD/DWORD по принципу битовой маски);
- Переменные DWORD/DINT/UDINT/REAL должны объявляться либо после переменной одного из таких же типов, либо после двух последовательно объявленных переменных типа WORD/UINT/INT.
- Переменные типа STRING должны иметь количество символов, кратное 4.

```

SLAVE_DATA
0001 TYPE SLAVE_DATA:
0002 STRUCT
0003   wVar1:      WORD;      (*регистр 0*)
0004   wVar2:      WORD;      (*регистр 1*)
0005   dwVar1:     DWORD;     (*регистры 2-3*)
0006   dwVar2:     DWORD;     (*регистры 4-5*)
0007   rVar:       REAL;      (*регистры 6-7*)
0008   dwVar3:     DWORD;     (*регистры 8-9*)
0009   sVar:       STRING(12); (*регистры 10-14*)
0010 END_STRUCT
0011 END_TYPE
  
```

Рис. 4.1 – Пример правильного объявления структуры


	 SLAVE_DATA
0001	TYPE SLAVE_DATA:
0002	STRUCT
0003	wVar1: WORD;
0004	dwVar1: DWORD;
0005	wVar2: WORD;
0006	dwVar2: DWORD;
0007	END_STRUCT
0008	END_TYPE

Рис. 4.2 – Пример **неправильного** объявления структуры – перед переменными типа DWORD объявлена только одна переменная типа WORD (их должно быть как минимум две)