

# Драйвер ModbusTCP MASTER для целевой системы ISaGRAF

**В статье описывается разработка драйвера протокола Modbus TCP системы ISaGRAF, который должен удовлетворять следующим требованиям: универсальность; простота использования, которая не требует высокой квалификации разработчика систем контроля и управления; увеличение функциональных возможностей целевой системы ISaGRAF для различных платформ (qnx4.25, qnx6.3, Linux, DOS32). Статья рассчитана на специалистов АСУ, использующих языки программирования IEC 61131-3.**

Ян БЕССОНОВ  
bessonov@fiord.com

## Протокол Modbus

Летом 2005 года на сайте [www.modbus-ida.org](http://www.modbus-ida.org) были опубликованы результаты исследования рынка автоматизации и программного обеспечения предприятий, проведенного компанией ARC Advisory Group в 2004 году. Исследование показывает, что среди группы протоколов industrial Ethernet протокол Modbus TCP наиболее распространен в промышленности.

Протокол Modbus предназначен для обеспечения связи промышленного оборудования на основе интерфейсов RS-485, Ethernet и др. В документации протокола зафиксировано, что он является де-факто стандартом в промышленности с 1979 года. Популярность его на рынке автоматизации обусловлена тем, что это открытый протокол, обеспечивающий возможность расширения. В связи с широкой популярностью Modbus перед разработчиками оборудования может стоять задача поддержки Modbus TCP. В данной статье описан пример решения задачи разработки драйвера протокола Modbus TCP MASTER для целевой системы ISaGRAF.

## Технология ISaGRAF

Технология ISaGRAF предназначена для программирования PLC-контроллеров. Она состоит из двух компонентов: среды разработки программ ISaGRAF Workbench и целевой системы исполнения программ. Первый из них предназначен для разработки и отладки программ на базе языков стандарта IEC 61131-3. Отладка программы в среде ISaGRAF Workbench производится на инструментальном компьютере в режиме симуляции, а также в режиме реального времени, когда программа исполняется на удаленном PLC-контроллере через интерфейсы Ethernet или RS-232/422/485. Программа может загружаться из среды разработки инструментального компьютера через интерфейсы Ethernet

или RS-232/485 на PLC-контроллер. Целевая система ISaGRAF предназначена для выполнения программ на PLC-контроллере, а также взаимодействия с его периферией. Взаимодействие целевой системы с периферией PLC происходит с помощью драйверов. Для того чтобы целевая система ISaGRAF поддерживала связь по протоколу Modbus TCP, были разработаны драйверы Modbus TCP MASTER и Slave.

Необходимо отметить преимущество совместного использования целевой системы ISaGRAF и протокола Modbus TCP, которое заключается в том, что используется единый интерфейс Ethernet для загрузки программ на PLC-контроллер и обеспечения связи контроллера с удаленными устройствами по Modbus TCP.

## Драйвер протокола Modbus в составе целевой системы ISaGRAF

Целевая система ISaGRAF предоставляет возможность разработчику реализовать драйвер в одном из двух вариантов: FBD-блок или

виртуальное устройство. Обе реализации имеют свои особенности, преимущества и недостатки. Особенность реализации драйвера в виде виртуального устройства заключается в том, что его функции read()/write() вызываются соответственно в начале и в конце цикла выполнения ресурса (исполнительной программы). Особенность FBD-блоков заключается в том, что они вызываются не обязательно циклично, а в соответствии с алгоритмом пользовательской программы. Таким образом, реализация драйвера в виде FBD-блока имеет преимущество для случая разового, редкого обмена данными с удаленным устройством Modbus TCP SLAVE, а также обладает возможностью выдачи данных в несколько выходных каналов различных типов данных за один вызов FBD-блока.

Преимущество реализации драйвера в виде виртуального устройства заключается в том, что разработчик освобождается от вызовов драйверов в своей программе, управление передает им сама исполнительная система в строгом соответствии со своей циклической схемой функционирования. Виртуальное устройство выполняет чтение значений каналов

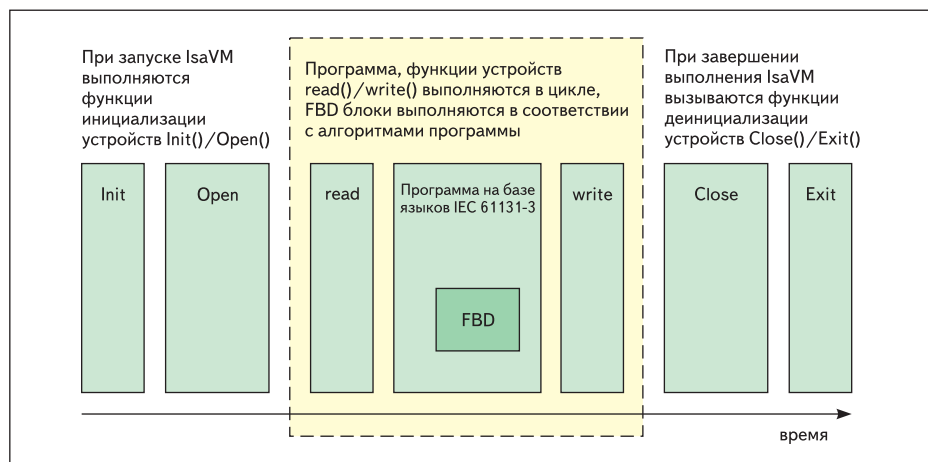


Рис. 1. Алгоритм драйвера Modbus

**Таблица**

Базовые функции протокола Modbus			Виртуальные устройства драйвера Modbus TCP Master	
Номер функции	Название функции	Тип данных	Название простого устройства	Тип данных простого устройства
1	read coils	Bool	di	bool
2	read discrete inputs	Bool	di	bool
3	read holding registers	unsigned integer	ai	dint, используются 16bit
			in_dint	dint, используются 32bit
			in_real	real, используются 32bit
4	read input registers	unsigned integer	ai	dint, используются 16bit
			in_dint	dint, используются 32bit
			in_real	real, используются 32bit
5	write single coil	bool	—	—
6	write singlr register	unsigned integer	—	—
15	write multiple coils	bool	do	bool
16	write multiple registers	unsigned integer	ao	dint, используются 16bit
			out_dint	dint, используются 32bit
			out_real	real, используются 32bit

в начале выполнения ресурса, а запись значений каналов — в конце выполнения ресурса.

При вызове каждого виртуального устройства ему передается указатель на соответствующий блок OEM-параметров и каналов устройства. Следующее преимущество виртуальных устройств заключается в том, что они имеют функции Init()/Exit и Open()/Close(), которые вызываются при первом старте и перед завершением целевой системы. Это свойство дает возможность разработчику определить в драйвере алгоритм инициализации соединения с удаленным устройством перед исполнением программы пользователя, а также алгоритм закрытия соединения перед завершением работы целевой системы. На рис. 1 представлен порядок запуска виртуальных устройств и FBD-блоков.

Драйвер Modbus TCP реализован в виде нескольких виртуальных устройств и встроен в целевую систему ISaGRAF. Это избавляет разработчика АСУ от необходимости помещать FBD-блоки внутри алгоритма для выполнения каждой транзакции. В состав драйвера входит одно комплексное устройство **ip\_client** (complex device) и 10 простых устройств (simple device). Комплексное устройство **ip\_client** выполняет функцию инициализации связи по виртуальному каналу с устройствами Modbus TCP SLAVE. В результате инициализации виртуального канала **ip\_client** сохраняет параметры виртуальной связи в динамической структуре данных. Эта структура данных с параметрами виртуального канала используется остальными виртуальными устройствами в процессе исполнения для обмена данными с удаленными устройствами Modbus TCP SLAVE. Каждое простое виртуальное устройство поддерживает одну из функций чтения-записи определенного типа данных. В таблице представлена информация о соответствии номеров функций протокола Modbus и названий виртуальных устройств драйвера Modbus TCP MASTER в целевой системе.

На основании данных колонки «Номер функции» из таблицы следует, что драйвер предоставляет пользователю возможность использовать стандартные базовые функции протокола Modbus TCP. Кроме того, драйвер

позволяет с помощью функций 3, 4 и 16 интерпретировать данные регистровых каналов протокола Modbus в качестве различных типов данных.

Например, устройства **in\_dint**, **out\_dint**, **in\_real**, **out\_real** интерпретируют два канала функций 3, 4 и 16 как один канал типа **dint** или **real**.

Для большей гибкости и совместимости с большим количеством типов контроллеров (Modbus TCP SLAVE) в структуру OEM-параметров виртуального устройства добавлены

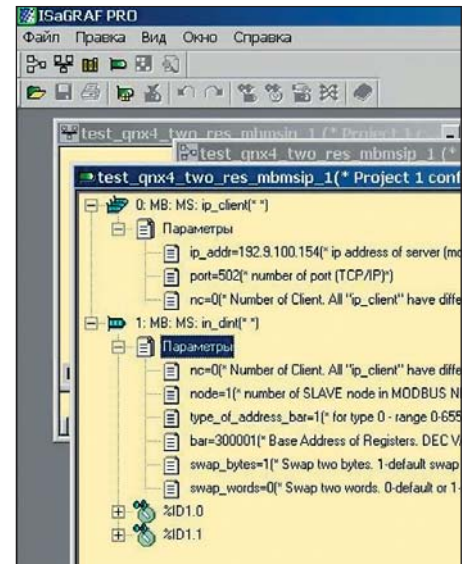


Рис. 2. Пример проекта

параметры **swap\_bytes** и **swap\_words**. В зависимости от значений параметров **swap\_bytes** и **swap\_words** происходит перестановка байт и слов в значении канала.

Устройство **ip\_client** содержит три OEM-параметра: **ip\_addr**, **port**, **nc**. В параметр

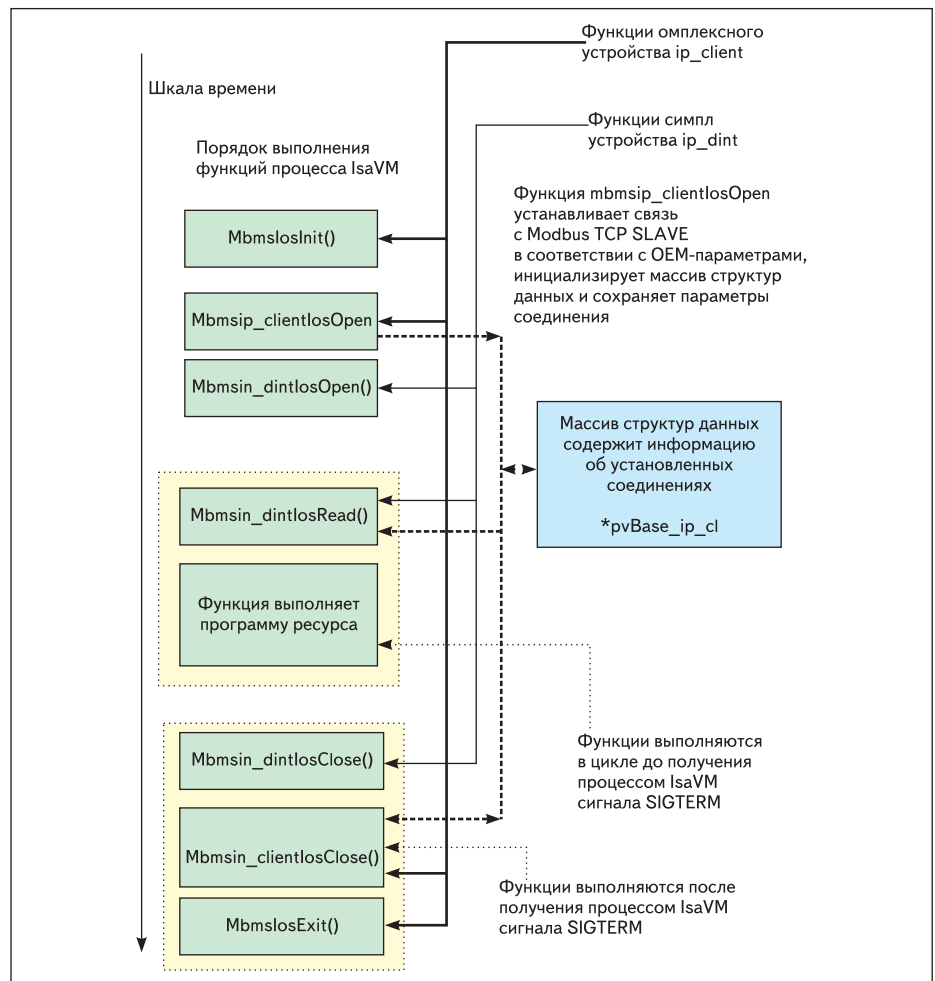


Рис. 3

**ip\_addr** пользователь вводит IP-адрес удаленного контроллера.

В параметр **port** пользователь вводит номер IP-порта устройства Modbus TCP SLAVE. В спецификации [1] обозначено со ссылкой на стандарт IP [3], что в стеке TCP/IP порт 502 закреплен за протоколом Modbus, но обычно ПО поддерживает возможность изменения номера порта. Пользователь вводит в параметр **pc** номер элемента в массиве структур данных, указывая на структуру, в которой сохраняется информация об инициализации связи с выбранным контроллером.

Простые виртуальные устройства содержат параметры:

- **pc** — номер элемента в массиве структур данных, содержащих информацию о соединении с удаленными контроллерами (ModbusTCP SLAVE);
- **node** — номер удаленного контроллера;
- **type\_of\_address\_bar** — форма записи адреса: 0-[0..65535], 1-[300001-365536] и [400001-465536];
- **bar** — адрес памяти;

- **swap\_bytes** — перестановка байтов: 0 — нет перестановки, 1 — есть перестановка;
- **swap\_words** — перестановка слов: 0 — нет перестановки, 1 — есть перестановка.

На рис. 2 представлен пример проекта с использованием драйвера Modbus TCP MASTER. В составе проекта находится комплексное устройство **ip\_client** и простое устройство **in\_dint**.

На рис. 3 представлен порядок выполнения функций комплексного устройства **ip\_client** и простого устройства **in\_dint**. Порядок выполнения базовых функций **Init()/Exit()**, **Open()/Close()** и **Write()/Read()** устройств определен в документации на ISaGRAF.

### Заключение

Разработанный драйвер обладает следующими свойствами:

- универсальность;
- простота использования;
- поддержка платформы qnx4.25, qnx6.3, Linux, DOS32;

- обмен данных между драйвером ISaGRAF Modbus TCP MASTER и Modbus TCP SLAVE происходит автоматически в каждом цикле целевой системы ISaGRAF;

- OEM-параметры простых устройств содержат поля данных для обеспечения совместимости с оборудованием различных производителей.

Драйвер достаточно удобен для пользователя, так как минимальным образом отвлекает на конфигурирование соединения и обеспечения связи с удаленными контроллерами внутри разрабатываемого алгоритма. Таким образом, основное время работы пользователя используется для разработки и отладки алгоритмов АСУ. ■

### Литература

1. [www.modbus.org](http://www.modbus.org) (Modbus\_Application\_Protocol\_V1\_1a.pdf)
2. [www.Modbus-IDA.org](http://www.Modbus-IDA.org)
3. RFC 791. Internet Protocol. Sep., 1981. DARPA.