

# **СЕТЬ ПРИБОРОВ**

# **ПРОТОКОЛ MODBUS**

**Руководство пользователя**  
**версия 1.12 от 21.01.2015**





## Содержание

1	Введение.....	4
2	Интерфейс RS-485.....	4
3	Протокол MODBUS.....	7
4	Функции протокола MODBUS.....	8
5	Карты распределения памяти приборов.....	18
5.1	Веха-С.....	20
5.2	Веха-Т.....	22
5.3	Параграф-PL20.....	25
5.4	Отіх-Р94(D4/W100)-МХ-1.....	31
5.5	Спрут-Т(В)-06-RS485.....	32
5.6	Спрут-Т-02-RS485-TC1/TC2.....	34
5.7	Спрут-ТВ-03-RS485.....	34
5.8	Ординар.....	35
6	Обратная связь.....	36
7	Использованные источники информации.....	36

## 1 Введение

В этом руководстве описываются основные принципы построения RS485 сети приборов, работающей по протоколу Modbus. Описываются реализованные в приборах функции. Представлены карты адресных пространств памяти приборов.

Целью данного руководства не является дублирование текстов стандартов. Здесь лишь акцентируется внимание на ключевых моментах. Детально изучить тонкости протокола и интерфейса можно ознакомившись с литературой, указанной в разделе “Использованные источники информации”.

## 2 Интерфейс RS-485

Стандарт ANSI TIA/EIA-485, более известный как RS485, определяет сбалансированный способ надёжной передачи данных на длинные расстояния в условиях промышленных помех. Также стандарт определяет топологию сети и описывает способы согласования полного сопротивления линии интерфейса и предоставляет результаты лабораторных тестов.

Физически, интерфейс RS485 является дифференциальным, обеспечивает многоточечные соединения и позволяет передавать и принимать данные в обоих направлениях.

Упрощённо, сеть интерфейса RS485 представляет собой приемопередатчики, соединенные при помощи витой пары - двух скрученных проводов (см. рис. 2.1).

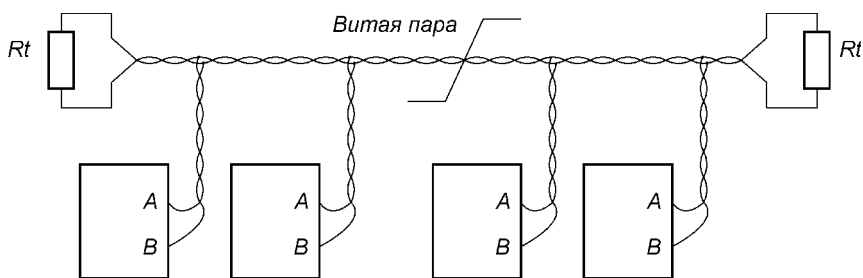


Рис. 2.1 Структура сети RS485

В основе интерфейса RS485 лежит принцип дифференциальной передачи сигнала. Суть его заключается в передаче одного сигнала по двум проводам. Причем по одному проводу (условно А) идет оригинальный сигнал, а по другому (условно В) - его инверсная копия.

Таблица 2.1 Сигналы линий (А и В) интерфейса RS485

Напряжение	Двоичное состояние	Напряжение на выходе передатчика
$V > A$	1	A: +1 Вольт B: +4 Вольта
$A < B$	0	A: +4 Вольта B: +1 Вольт

Типовая разница напряжений между линиями А и В передатчика равна  $3V$ , минимальная  $1.5V$ , максимальная  $5V$ .

Разница напряжений между линиями А и В на приёмнике должна быть не менее  $0.2V$  и абсолютная разница потенциалов относительно общего провода должно быть в диапазоне  $(-7...+12) V$ .

Таким образом, между двумя проводами витой пары всегда есть разность потенциалов. Именно этой разностью потенциалов и передается сигнал. Такой способ передачи обеспечивает высокую устойчивость к синфазной помехе. Максимальная скорость связи прибора по интерфейсу RS485 может достигать нескольких Мбод. Максимальное расстояние - 1200 метров. Если необходимо организовать связь на расстоянии больше чем 1200 метров или подключить больше устройств, чем допускает нагрузочная способность передатчика - применяют специальные повторители (репитеры). Типовое правило для расчёта максимальной длины линии связи таково: произведение скорости передачи в бодах на длину в метрах должно дать результат не более чем 108.

При значительных расстояниях между устройствами, связанными по витой паре или высоких скоростях передачи начинают проявляться так называемые эффекты длинных линий. Электромагнитный сигнал имеет свойство отражаться от открытых концов линии передачи и ее ответвлений. Фронт сигнала, отразившийся от конца линии, может исказить текущий или следующий сигнал. В таких случаях нужно подавлять эффект отражения.

Существуют стандартные решения этой проблемы (R, RC - терминаторы). У любой линии связи есть такой параметр, как волновое сопротивление  $Z_v$ . Оно зависит от характеристик используемого кабеля и не зависит от его длины. Для обычно применяемых в линиях связи витых пар волновое сопротивление  $Z_v$  составляет (90-120) Ом. Рассмотрим варианты:

1. Если на удаленном конце линии, между проводниками витой пары включить резистор с номинальным омическим

сопротивлением равным волновому сопротивлению линии, то электромагнитная волна, дошедшая до «тупика» поглощается на таком резисторе. Отсюда его названия - согласующий резистор или «терминатор».

Помимо достоинств этого метода (повышение скорости, увеличение длины и подавление отражений), есть и недостатки (дополнительная нагрузка на драйверы повышает энергопотребление, остальные ответвления линии продолжают вносить искажения, драйвер приёмника находится в неоднозначном состоянии: либо режим ожидания, либо режим приёма).

2. Если на удалённом конце вместо резистора установить RC цепочку  $R=(90-120)$  Ом,  $C=1000$  пФ, то можно устранить проблему повышенного энергопотребления и проблему неопределённости драйвера приёмника (для приёмников с функциями open-line и fail-safe). Но из-за постоянной времени RC цепи, максимальная скорость передачи и длина линии будут меньшими.

Эффект отражения и необходимость правильного согласования накладывают ограничения на конфигурацию линии связи (топология сети). Линия связи должна представлять собой один кабель витой пары. К этому кабелю присоединяются все приемники и передатчики (гирлянда). Расстояние от линии до микросхем интерфейса RS485 должно быть как можно короче, так как длинные ответвления вносят рассогласование и вызывают отражения. В оба наиболее удаленных конца кабеля включают терминаторы. Калибр витой пары достаточно не более AWG24.

В случае наличия клемм заземления у приборов, логичным является их заземление. Но из-за этого возможно возникновение паразитных токов во время кратковременных высоковольтных помех. Для устранения этого эффекта необходимо применять следующую схему заземления с ограничительными резисторами.

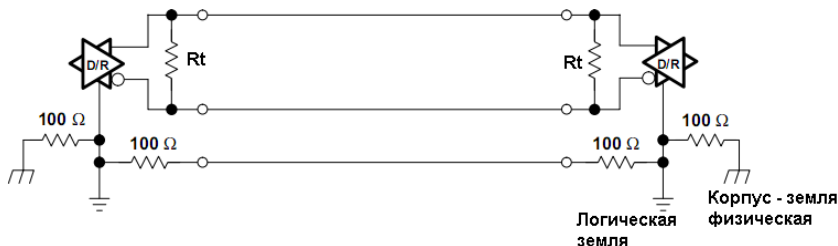


Рис. 2.2 Структура сети RS485 с заземлением

Следует также сказать, что к линии интерфейса все устройства подключаются через специализированные микросхемы

(драйверы интерфейса RS485). Они могут быть разных производителей и с различными техническими параметрами и как следствие различной стоимости. Эти драйверы в значительной степени могут определять эксплуатационные свойства приборов: дальность передачи, количество приборов в одном участке сети и надёжность передачи.

### 3 Протокол MODBUS

MODBUS – это протокол уровня приложений (уровень 7 модели OSI), что обеспечивает связь между устройствами, соединёнными различными каналами связи и сетями.

Де-факто, MODBUS является стандартом в сетях промышленного назначения с 1979 года. Он обеспечивает связь миллионам устройств во всём мире, в том числе и через Интернет. Есть различные реализации протокола:

- Для асинхронных беспроводных, оптических и проводных каналов связи (RS-232, RS-485, RS-422).
- Для TCP/IP (порт 502) через интернет
- MODBUS-PLUS - для высокоскоростных сетей с передачей меток (high speed token passing network).

Кроме того, разнородные участки сетей могут объединяться шлюзами (специальными конверторами).

Для асинхронных последовательных каналов связи существует две реализации MODBUS-SERIAL-LINE протокола MODBUS-RTU и MODBUS-ASCII (уровень 1 и 2 модели OSI). Разница между ними заключается в способе кодировки данных, способе синхронизации фреймов, и алгоритме обеспечения целостности данных. В нашем случае, в сети RS485 обмен данными реализован посредством протокола MODBUS-RTU. Далее по тексту будем рассматривать ситуацию только в этом аспекте.

MODBUS-SERIAL-LINE протокол – это протокол типа MASTER-SLAVE (протокол запросов-ответов). Ведущий в сети (MASTER) всегда один. Каждый подчинённый (SLAVE) должен иметь уникальный номер 1-247. Адрес 0 – это широковещательный запрос, адресованный сразу всем подчинённым. Таким образом, логически в одном участке сети может быть до 248 устройств (включая MASTER). Каждый запрос содержит код функции. Под MODBUS-функциями понимают определённые сервисы предоставляемые подчинёнными ведущему. Таким образом, роль клиента играет MASTER, а роль сервера, с определённым набором функций-сервисов, SLAVE.

## 4 Функции протокола MODBUS

Каждый SLAVE может содержать уникальный набор функций-сервисов, но есть и ряд стандартных функций, которые подробно описаны в документе “MODBUS Application Protocol Specification” ([www.modbus.org](http://www.modbus.org)). Также полезная информация может быть найдена в документе “MODBUS over serial line specification and implementation guide” ([www.modbus.org](http://www.modbus.org)).

Поддерживаемые нами функции (см. табл. 4.1 - 4.2).

Таблица 4.1 Стандартные функции

Функция	Название	Тип данных (big-endian)
1	read coils	sram bits
2	read discrete inputs	sram bits
3	read holding registers	eeprom words
4	read input registers	sram words
5	write single coil	sram bit
16	write holding registers	eeprom words

Таблица 4.2 Пользовательские функции

Функция	Название	Тип данных (little-endian)
100	read eeprom bytes	eeprom byte
101	write eeprom bytes	eeprom byte
102	read dataflash bytes	dataflash byte
104	read sram bytes	sram byte
108	service function	byte
109	archive function	byte

В более ранних версиях приборов (до 2010г) были реализованы лишь пользовательские функции, но со временем стало понятно, что для обеспечения совместного использования приборов с ПЛК (минуя ПК) необходимы и стандартные функции.

Будьте внимательны и обратите внимание на то, что стандартные функции оперируют только со словами (16-бит) и в формате big-endian, но при этом формат контрольной суммы CRC16 little-endian! Поэтому, для исключения разночтений в описании протокола MODBUS, в части порядка следования байт контрольной суммы CRC16, стоит пользоваться нехитрым правилом: правильно посчитанная контрольная сумма неповреждённого пакета (с участием 2-ух последних байт CRC16) всегда равна нулю.

Правильный запрос: CRC16 (1 104 0 0 8 0 103 195) = 0

Неверный запрос: CRC16 (1 104 0 0 8 0 195 103) <> 0



Далее в таблицах 4.3-4.7 будут представлены развёрнутые описания пользовательских функций.

Стандартные функции (см. таб. 4.1) подробно описаны в документе “MODBUS Application Protocol Specification” ([www.modbus.org](http://www.modbus.org)).

Таблица 4.3 Пользовательская функция 100

<b>Функция 100. Чтение энергонезависимой памяти (EEPROM)</b>				
<b>байт</b>	<b>описание</b>	<b>значение</b>	<b>порядок</b>	<b>размер</b>
<b>Фрейм ЗАПРОСА:</b>				
0	slave's adr	1-247		1 byte
1	function	100		1 byte
2	address	0-END	low	2 bytes
3			high	
4	bytes qty	1-248	low	2 bytes
5			high	
6	CRC16		low	2 bytes
7			high	
<b>Фрейм ОТВЕТА:</b>				
0	slave's adr	1-247		1 byte
1	function	100		1 byte
2	address	0-END	low	2 bytes
3			high	
4	bytes qty	1-248	low	2 bytes
5			high	
6	data (byte0)	(1-248) байт данных		1 byte
7	data (byte1)			1 byte
...	...			1 byte
252	data (byte246)			1 byte
253	data (byte247)			1 byte
254	CRC16			low
255		high		

<b>Фрейм ОТВЕТА С КОДОМ ОШИБКИ:</b>				
0	slave's adr	1-247		1 byte
1	function	100 + 0x80		1 byte
2	exception code	1	wrong function	1 byte
		2	wrong address	
		3	wrong bytes qty	
		4	read error	
3	CRC16		low	2 bytes
4			high	

Таблица 4.4 Пользовательская функция 101

<b>Функция 101. Запись в энергонезависимую память (EEPROM)</b>				
байт	описание	значение	порядок	размер
<b>Фрейм ЗАПРОСА:</b>				
0	slave's adr	1-247		1 byte
1	function	101		1 byte
2	address	0-END	low	2 bytes
3			high	
4	bytes qty	1-248	low	2 bytes
5			high	
6	data (byte0)	(1-248) байт данных		1 byte
7	data (byte1)			1 byte
...	...			1 byte
252	data (byte246)			1 byte
253	data (byte247)			1 byte
254	CRC16			low
255			high	
<b>Фрейм ОТВЕТА:</b>				
0	slave's adr	1-247		1 byte
1	function	101		1 byte

2	address	0-2047	low	2 bytes
3			high	
4	bytes qty	1-248	low	2 bytes
5			high	
6	CRC16		low	2 bytes
7			high	
<b>Фрейм ОТВЕТА С КОДОМ ОШИБКИ:</b>				
0	slave's adr	1-247		1 byte
1	function	101 + 0x80		1 byte
2	exception code	1	wrong function	1 byte
		2	wrong address	
		3	wrong bytes qty	
		4	write error	
3	CRC16		low	2 bytes
4			high	

Таблица 4.5 Пользовательская функция 102

<b>Функция 102. Чтение энергонезависимой памяти (DATA_FLASH)</b>				
байт	описание	значение	порядок	раз-мер
<b>Фрейм ЗАПРОСА:</b>				
0	slave's adr	1-247		1 byte
1	function	102		1 byte
2	page	0-8191	low	2 bytes
3			high	
4	address	0-527	low	2 bytes
5			high	
6	bytes qty	1-246	low	2 bytes
7			high	
8	CRC16		low	2 bytes

9			high	
<b>Фрейм ОТВЕТА:</b>				
0	slave's adr	1-247		1 byte
1	function	102		1 byte
2	page	0-8191	low	2 bytes
3			high	
4	address	0-527	low	2 bytes
5			high	
6	bytes qty	1-246	low	2 bytes
7			high	
8	data (byte0)	(1-246) байт данных		1 byte
9	data (byte1)			1 byte
...	...			1 byte
252	data (byte244)			1 byte
253	data (byte245)			1 byte
254	CRC16			low
255		high		
<b>Фрейм ОТВЕТА С КОДОМ ОШИБКИ:</b>				
0	slave's adr	1-247		1 byte
1	function	102 + 0x80		1 byte
2	exception code	1	wrong function	1 byte
		2	wrong address	
		3	wrong bytes qty	
		4	read error	
3	CRC16		low	2 bytes
4		high		

Таблица 4.6 Пользовательская функция 104

<b>Функция 104. Чтение оперативной памяти (SRAM)</b>				
<b>байт</b>	<b>описание</b>	<b>значение</b>	<b>порядок</b>	<b>размер</b>
<b>Фрейм ЗАПРОСА:</b>				
0	slave's adr	1-247		1 byte
1	function	104		1 byte
2	address	0-END	low	2 bytes
3			high	
4	bytes qty	1-248	low	2 bytes
5			high	
6	CRC16		low	2 bytes
7			high	
<b>Фрейм ОТВЕТА:</b>				
0	slave's adr	1-247		1 byte
1	function	104		1 byte
2	address	0-END	low	2 bytes
3			high	
4	bytes qty	1-248	low	2 bytes
5			high	
6	data (byte0)	(1-248) байт данных		1 byte
7	data (byte1)			1 byte
...	...			1 byte
252	data (byte246)			1 byte
253	data (byte247)			1 byte
254	CRC16			low
255		high		
<b>Фрейм ОТВЕТА С КОДОМ ОШИБКИ:</b>				
0	slave's adr	1-247		1 byte
1	function	104 + 0x80		1 byte

2	exception code	1	wrong function	1 byte
		2	wrong address	
		3	wrong bytes qty	
		4	read error	
3	CRC16		low	2 bytes
4			high	

Таблица 4.7 Пользовательская функция 108

<b>Функция 108. Служебные команды</b>				
байт	описание	значение	порядок	размер
<b>Фрейм ЗАПРОСА:</b>				
0	slave's adr	1-247		1 byte
1	function	108		1 byte
2	SubFunction	1 - DEVICE_START 2 - DEVICE_STOP 3 - DEVICE_RESET 4 - RS485_RESET 5 - xxx 6 - xxx 7 - GET_DEVICE_NAME 8 - GET_DEVICE_ID 9 - UPDATE_FIRMARE 10 - GET_FULL_ID 11 - GET_BOOT_SIZE		1 byte
3	CRC16		low	2 bytes
4			high	
<b>Фрейм ОТВЕТА:</b>				
0	slave's adr	1-247		1 byte
1	function	108		1 byte
2	SubFunction	1-11		1 byte
3	data (byte0)	(0-251) байт данных, если таковые имеются		1 byte
...	...			1 byte

252	data (249)			1 byte
253	data (250)			1 byte
254	CRC16		low	2 bytes
255			high	
<b>Фрейм ОТВЕТА С КОДОМ ОШИБКИ:</b>				
0	slave's adr	1-247		1 byte
1	function	108 + 0x80		1 byte
2	exception code	1	wrong function	1 byte
		2	wrong address	
		3	wrong bytes qty	
		4	read error	
3	CRC16		low	2 bytes
4			high	

Функция 108 «Служебные команды» имеет следующие коды подфункций (см. таб. 4.8).

Подфункции, возвращающие какие-либо данные, имеют префикс GET. Подфункции, не возвращающие данных, не содержат поля данных и, при удачном выполнении, возвращаются эхом.

*Таблица 4.8 Коды подфункций функции 108*

<b>№</b>	<b>Имя</b>	<b>Описание</b>	<b>Размер блока данных, байт</b>
1	DEVICE_START	Запустить прибор	0
2	DEVICE_STOP	Остановить прибор	0
3	DEVICE_RESET	Перезагрузить прибор	0
4	RS485_RESET	Проинициализировать интерфейс	0
7	GET_DEVICE_NAME	Получить имя прибора (строка)	≤ 251

8	GET_DEVICE_ID	Получить идентификационный номер прибора	1
9	UPDATE_FIRMWARE	Перейти в режим обновления микропрограммы	0
10	GET_FULL_ID	Получить полный идентификационный номер прибора	4
11	GET_BOOT_SIZE	Получить размер секции загрузчика	2

Таблица 4.9 Код приборов подфункции GET\_DEVICE\_ID

Название прибора	Код
DEVICE_ID_PARAGRAPH_PL20	5
DEVICE_ID_VEHA_C	8
DEVICE_ID_VEHA_T	9
DEVICE_ID_OMIX_MX_1	10
DEVICE_ID_SPRUT_T_06_RS485	11
DEVICE_ID_SPRUT_T_02_RS485_TS1	12
DEVICE_ID_LED_TABLO	13
DEVICE_ID_SPRUT_TV_03_RS485	14
DEVICE_ID_ORDINAR	15
DEVICE_ID_VEHA_RTC	16
DEVICE_ID_SPRUT_T_02_RS485_TS2	17

Таблица 4.10 Пользовательская функция 109

Функция 109. Работа с файловой системой DFFS				
байт	описание	значение	порядок	размер
<b>Фрейм ЗАПРОСА:</b>				
0	slave's adr	1-247		1 byte
1	function	109		1 byte
2	SubFunc	1.GET_FIRST_VOLUME 2.GET_LAST_VOLUME 3.GET_VOL_BEGIN_PAGE 4.GET_VOL_END_PAGE		1 byte



3	CRC16		low	2 bytes
4			high	
<b>Фрейм ОТВЕТА:</b>				
0	slave's adr	1-247		1 byte
1	function	109		1 byte
2	SubFunc	1-4		1 byte
3	data(byte0)	(4 или 6) байт данных		1 byte
4	data(byte1)			1 byte
...				1 byte
6	data(byte3)			1 byte
7	CRC16		low	2 bytes
8			high	
<b>Фрейм ОТВЕТА С КОДОМ ОШИБКИ:</b>				
0	slave's adr	1-247		1 byte
1	function	109 + 0x80		1 byte
2	exception code	1	wrong function	1 byte
		2	wrong address	
		3	wrong bytes qty	
		4	read error	
3	CRC16		low	2 bytes
4			high	

Подфункции 1 и 2, возвращающие номер тома всегда возвращают 4-х байтное значение типа DWORD.

Подфункции 3 и 4, возвращают номера страниц как 2-х байтные (WORD) значения, при этом до номера страницы всегда идёт номер тома этой страницы (DWORD). Таким образом получается 6 байт данных.

## 5 Карты распределения памяти приборов

В следующих таблицах представлены карты распределения памяти приборов. Следует отметить тот факт, что в стандартных MODBUS функциях размеры типов данных могут отличаться от типов данных пользовательских функций (в большую сторону) в случае, если размер типа данных не кратен типу WORD (2 байта).

Порядок следования байт указан в столбце Order. Обозначение BE соответствует порядку big endian, а LE – little endian.

Операции, доступные для данной переменной, указываются в последнем столбце gw (read-write). R – разрешается только чтение, W – разрешается только запись, RW – разрешается, как чтение, так и запись.

Массивы обозначены словом array, а количество элементов массива указано в квадратных скобках [n].

Таблица 5.1 Используемые типы данных

Тип данных	Описание
BIT	бит
BYTE	байт размером в 8 бит
WORD	слово без знака 2 байта
DWORD	двойное слово без знака 4 байта
SIGNED WORD	слово со знаком 2 байта
float	число с плавающей точкой в формате IEEE754 размером 4 байта
typedef struct { WORD f; BYTE n_z; BYTE x; } sfloat_t;	Структура с тремя полями:  Значение (4 десятичных разряда). Число знаков после запятой. Множитель: 1, 10 или 100. Размер этой структуры 4 байта или 3 слова в стандартных функциях
typedef struct { BYTE sec; BYTE min; BYTE hour; BYTE hsec; BYTE day; BYTE month; BYTE year; } rtc_t;	Структура с 7 полями:  [0..59] секунды [0..59] минуты [0..23] часы [0..99] sec/100 = сотые секунд [1..31] день месяца [1..12] месяц [0..99], 0 = 2000 год Размер этой структуры 7 байт или

	7 слов в стандартных функциях
<pre>typedef struct {     BYTE relay;     BYTE reg_ena;     BYTE ctrl_reg_start;      BYTE cond_type;     float sp;     float dsp; } reg_set_t;</pre>	<p>Структура с 6 полями:</p> <p>Привязка измеряемой величины к реле  Регистировать или нет  Определяет ли изм. величина условие запуска регистрации  Тип условия запуска регистрации  Уставка  Дельта (гистерезис)  Размер этой структуры 8 слов в стандартных функциях</p>
<pre>typedef struct {     float sp;     float dsp; } rel_set_t;</pre>	<p>Структура с 2 полями:</p> <p>Уставка для реле  Дельта (гистерезис)  Размер этой структуры 4 слова в стандартных функциях</p>
<pre>typedef struct {     float min;     float max;     BYTE convert_param; } dac_t;</pre>	<p>Структура с 3 полями:</p> <p>Нижняя граница масштабирования  Верхняя граница масштабирования  Преобразуемая изм. величина  Размер этой структуры 5 слов в стандартных функциях</p>
<pre>typedef struct {     BYTE year;     BYTE month;     BYTE day;     BYTE hour;     BYTE min;     BYTE sec; } omix_rtc_t;</pre>	<p>Структура с 6 полями:</p> <p>[0..99], 0 = 2000 год  [1..12] месяц  [1..31] день месяца  [0..23] часы  [0..59] минуты  [0..59] секунды  Размер этой структуры 6 слов в стандартных функциях</p>
WORD_FP1	число в формате с фиксированной десятичной точкой. Число дробных разрядов — 1.
WORD_FP2	число в формате с фиксированной десятичной точкой. Число дробных разрядов — 2.

## 5.1 Веха-С

Таблица 5.2 Карта распределения памяти Веха-С

Addr	Type	Name	Order	Size	rw
<b>Coils - std modbus functions 1 &amp; 5 (bit addressing mode)</b>					
0	array of BIT	relay[2]		2	rw
8	BIT	DEVICE_START_STOP		1	rw
9	BIT	DEVICE_RESET		1	w
13	BIT	UPDATE_FIRMWARE		1	w
14	BIT	COUNTER_RESET		1	w
<b>Discrete inputs - std modbus function 2 (bit addressing mode)</b>					
0	BIT	ext_key_stop		1	r
1	BIT	ext_key_reset		1	r
<b>Holding registers - std modbus 3 &amp; 16 (word addressing mode)</b>					
0	array of sfloat_t	ee_sp[2]	BE	6	rw
6	array of sfloat_t	ee_relay_n_on[2]	BE	6	rw
12	array of sfloat_t	ee_relay_t_on[2]	BE	6	rw
18	BYTE	ee_cnt_logic	BE	1	rw
19	BYTE	ee_sensor_type	BE	1	rw
20	WORD	ee_divisor	BE	1	rw
21	sfloat_t	ee_multiplier	BE	3	rw
24	sfloat_t	ee_init_value	BE	3	rw
27	BYTE	ee_overflow_logic	BE	1	rw
28	BYTE	ee_power_off_logic	BE	1	rw
29	BYTE	ee_ext_stop_logic	BE	1	rw
30	array of BYTE	ee_relay_logic[2]	BE	2	rw
32	BYTE	ee_pass_level	BE	1	rw

33	BYTE	ee_rs485_number	BE	1	rw
34	BYTE	ee_rs485_speed	BE	1	rw
35	BYTE	ee_rs485_byte_len	BE	1	rw
36	BYTE	ee_rs485_parity	BE	1	rw
37	BYTE	ee_rs485_stop_	BE	1	rw
38	float	ee_backup_counter	BE	2	rw
40	BYTE	ee_led_br_level	BE	1	rw
41	sfloat_t	ee_dac_min	BE	3	rw
44	sfloat_t	ee_dac_max	BE	3	rw
47	BYTE	ee_dac_type	BE	1	rw
<b>Input registers - std modbus function 4 (word addressing mode)</b>					
0	float	cnt.result.floating	BE	2	r
2	DWORD	cnt.result.integer	BE	2	r
4	DWORD	cnt.result.wo_multiplier	BE	2	r
<b>SRAM - user's function 104 (byte addressing mode)</b>					
0	float	cnt.result.floating	LE	4	r
<b>EEPROM - user's function 100 &amp; 101 (byte addressing mode)</b>					
1	array of sfloat_t	ee_sp[2]	LE	8	rw
9	array of sfloat_t	ee_relay_n_on[2]	LE	8	rw
17	array of sfloat_t	ee_relay_t_on[2]	LE	8	rw
25	BYTE	ee_cnt_logic	LE	1	rw
26	BYTE	ee_sensor_type	LE	1	rw
27	WORD	ee_divisor	LE	2	rw
29	sfloat_t	ee_multiplier	LE	4	rw
33	sfloat_t	ee_init_value	LE	4	rw
37	BYTE	ee_overflow_logic	LE	1	rw

38	BYTE	ee_power_off_logic	LE	1	rw
39	BYTE	ee_ext_stop_logic	LE	1	rw
40	array of BYTE	ee_relay_logic[2]	LE	2	rw
42	BYTE	ee_pass_level	LE	1	rw
43	BYTE	ee_rs485_number	LE	1	rw
44	BYTE	ee_rs485_speed	LE	1	rw
45	BYTE	ee_rs485_byte_len	LE	1	rw
46	BYTE	ee_rs485_parity	LE	1	rw
47	BYTE	ee_rs485_stop_bits	LE	1	rw
48	float	ee_backup_counter	LE	4	rw
52	BYTE	ee_led_br_level	LE	1	rw
53	BYTE	ee_dac_type	LE	1	rw
54	sfloat_t	ee_dac_min	LE	4	rw
58	sfloat_t	ee_dac_max	LE	4	rw

## 5.2 Bexa-T

Таблица 5.3 Карта распределения памяти Bexa-T

Addr	Type	Name	Order	Size	rw
<b>Coils - std modbus functions 1 &amp; 5 (bit addressing mode)</b>					
0	array of BIT	relay[2]		2	rw
8	BIT	DEVICE_START_STOP		1	rw
9	BIT	DEVICE_RESET		1	w
13	BIT	UPDATE_FIRMWARE		1	w
14	BIT	COUNTER_RESET		1	w
15	BIT	UPTIME_RESET		1	w
<b>Discrete inputs - std modbus function 2 (bit addressing mode)</b>					
0	BIT	ext_key_start		1	r
1	BIT	ext_key_div1_div2		1	r
<b>Holding registers - std modbus 3 &amp; 16 (word addressing mode)</b>					

0	array of sfloat_t	ee_sp[2]	BE	6	rw
6	array of sfloat_t	ee_dsp[2]	BE	6	rw
12	BYTE	ee_indicator_format	BE	1	rw
13	WORD	ee_f_input_max	BE	1	rw
14	WORD	ee_waiting_time	BE	1	rw
15	WORD	ee_divisor1	BE	1	rw
16	WORD	ee_divisor2	BE	1	rw
17	sfloat_t	ee_multiplier	BE	3	rw
20	WORD	ee_start_delay	BE	1	rw
21	BYTE	ee_start_logic	BE	1	rw
22	array of BYTE	ee_relay_logic[2]	BE	2	rw
24	BYTE	ee_pass_level	BE	1	rw
25	sfloat_t	ee_dac_speed_min	BE	3	rw
28	sfloat_t	ee_dac_speed_max	BE	3	rw
31	BYTE	ee_dac_type	BE	1	rw
32	BYTE	ee_rs485_number	BE	1	rw
33	BYTE	ee_rs485_speed	BE	1	rw
34	BYTE	ee_rs485_byte_len	BE	1	rw
35	BYTE	ee_rs485_parity	BE	1	rw
36	BYTE	ee_rs485_stop_bits	BE	1	rw
37	BYTE	ee_show_mode	BE	1	rw
38	DWORD	ee_uptime	BE	2	rw
40	BYTE	ee_led_br_level	BE	1	rw
41	array of sfloat_t	ee_lo_alarm[2]	BE	6	rw
47	array of sfloat_t	ee_hi_alarm[2]	BE	6	rw
53	DWORD	turn_counter_wo_div	BE	2	rw

<b>Inputregisters-stdmodbusfunction4(wordaddressingmode)</b>					
0	float	tacho.result.floating_abs	BE	2	r
2	float	tacho.result.floating	BE	2	r
4	DWORD	tacho.result.integer_abs	BE	2	r
6	SIGNED DWORD	tacho.result.integer	BE	2	r
8	BYTE	tacho.result.dir	BE	1	r
9	DWORD	tacho.result.uptime_sec	BE	2	r
11	DWORD	tacho.result.uptime_hour	BE	2	r
13	DWORD	tacho.result. turn_counter_wo_div	BE	2	r
15	DWORD	tacho.result.turn_counter	BE	2	r
<b>SRAM-user'sfunction104(byteaddressingmode)</b>					
0	float	tacho.result.floating_abs	LE	4	r
<b>EEPROM-user'sfunction100&amp;101(byteaddressingmode)</b>					
1	array of sfloat_t	ee_sp[2]	LE	8	rw
9	array of sfloat_t	ee_dsp[2]	LE	8	rw
17	BYTE	ee_indicator_format	LE	1	rw
18	WORD	ee_f_input_max	LE	2	rw
20	WORD	ee_waiting_time	LE	2	rw
22	WORD	ee_divisor1	LE	2	rw
24	sfloat_t	ee_multiplier	LE	4	rw
28	WORD	ee_start_delay	LE	2	rw
30	BYTE	ee_start_logic	LE	1	rw
31	array of BYTE	ee_relay_logic[2]	LE	2	rw
33	BYTE	ee_pass_level	LE	1	rw



34	sfloat_t	ee_dac_speed_min	LE	4	rw
38	sfloat_t	ee_dac_speed_max	LE	4	rw
42	BYTE	ee_rs485_number	LE	1	rw
43	BYTE	ee_rs485_speed	LE	1	rw
44	BYTE	ee_rs485_byte_len	LE	1	rw
45	BYTE	ee_rs485_parity	LE	1	rw
46	BYTE	ee_rs485_stop_bits	LE	1	rw
47	BYTE	ee_show_mode	LE	1	rw
48	DWORD	ee_uptime	LE	4	rw
52	BYTE	ee_led_br_level	LE	1	rw
53	array of sfloat_t	ee_lo_alarm[2]	LE	8	rw
61	array of sfloat_t	ee_hi_alarm[2]	LE	8	rw
69	WORD	ee_divisor2	LE	2	rw
71	BYTE	ee_dac_type	LE	1	rw
72	DWORD	ee_turn_counter_wo_div	LE	4	rw

### 5.3 Параграф-PL20

Таблица 5.4 Карта распределения памяти Параграф

Addr	Type	Name	Order	Size	rw
<b>Coils - std modbus functions 1 &amp; 5 (bit addressing mode)</b>					
0	array of BIT	relay[4]		4	rw
8	BIT	DEVICE_START_STOP		1	rw
9	BIT	DEVICE_RESET		1	w
13	BIT	UPDATE_FIRMWARE		1	w
<b>Discrete inputs - std modbus function 2 (bit addressing mode)</b>					
0	BIT	ext_key_start		1	r
1	BIT	ext_key_stop		1	r
<b>Holding registers - std modbus 3 &amp; 16 (word addressing mode)</b>					

0	array of float	ee_sp[4]	BE	8	rw
8	array of float	ee_dsp[4]	BE	8	rw
16	array of float	ee_amplifier[3]	BE	6	rw
22	array of float	ee_offset[3]	BE	6	rw
28	array of float	ee_analog_in_min[2]	BE	4	rw
32	array of float	ee_analog_in_max[2]	BE	4	rw
36	array of float	ee_dac_min[2]	BE	4	rw
40	array of float	ee_dac_max[2]	BE	4	rw
44	array of BYTE	ee_relay_owner[4]	BE	4	rw
48	array of BYTE	ee_sensor_type[2]	BE	2	rw
50	array of BYTE	ee_math_processing_type[2]	BE	2	rw
52	array of BYTE	ee_compensation_state[2]	BE	2	rw
54	array of BYTE	ee_relay_logic[4]	BE	4	rw
58	array of BYTE	ee_dac_owner[2]	BE	2	rw
60	array of BYTE	ee_dac_type[2]	BE	2	rw
62	BYTE	ee_rs485_number	BE	1	rw
63	BYTE	ee_rs485_speed	BE	1	rw
64	BYTE	ee_rs485_byte_len	BE	1	rw
65	BYTE	ee_rs485_parity	BE	1	rw

66	BYTE	ee_rs485_stop_bits	BE	1	rw
67	BYTE	ee_plot_x_scale	BE	1	rw
68	BYTE	ee_dffs_rec_interval	BE	1	rw
69	BYTE	ee_access	BE	1	rw
70	BYTE	ee_dffs_rec_mode	BE	1	rw
71	BYTE	ee_work_ctl	BE	1	rw
72	BYTE	ee_work_rec	BE	1	rw
73	BYTE	ee_ext_keys_logic	BE	1	rw
74	array of BIT	ee_plot_visible_series[2]	BE	1	rw
75	array of BYTE	ee_median_filter_rank[3]	BE	3	rw
78	array of float	ee_integral[2]	BE	4	rw
82	rtc_t	ee_rtc_edit	BE	7	rw
89	array of BYTE	ee_led_n_f[2]	BE	2	rw
91	BYTE	ee_language	BE	1	rw
92	BYTE	ee_plot_y_autoscale	BE	1	rw
93	float	ee_plot_y_min	BE	2	rw
95	float	ee_plot_y_max	BE	2	rw
97	BYTE	ee_plot_y_min_n_f	BE	1	rw
98	BYTE	ee_plot_y_max_n_f	BE	1	rw
99	float	ee_pid_t	BE	2	rw
101	array of float	ee_pid_xp[4]	BE	8	rw
109	array of float	ee_pid_ti[4]	BE	8	rw
117	array of float	ee_pid_td[4]	BE	8	rw
125	array of float	ee_pid_min_power[4]	BE	8	rw

133	array of float	ee_pid_max_power[4]	BE	8	rw
141	array of float	ee_pid_alarm_power[4]	BE	8	rw
149	array of BYTE	ee_sensor_validate[2]	BE	2	rw
151	array of BYTE	ee_relay_on_alarm_state[4]	BE	4	rw
155	BYTE	ee_rtc_summer_time_mode_edit	BE	1	rw
156	WORD	ee_rtc_daily_time_correction	BE	1	rw
<b>Input registers - std modbus function 4 (word addressing mode)</b>					
0	array of float	analog_inputs[3]	BE	6	r
6	array of WORD	dac_pwm[2]	BE	2	r
8	array of BYTE	relays_state[4]	BE	4	r
12	rtc_t	rtc	BE	7	r
<b>SRAM - user's function 104 (byte addressing mode)</b>					
0	array of float	analog_inputs[3]	LE	12	r
12	array of WORD	dac_pwm[2]	LE	4	r
16	array of BYTE	relays_state[4]	LE	4	r
20	rtc_t	rtc	LE	7	r
<b>EEPROM - user's function 100 &amp; 101 (byte addressing mode)</b>					
1	array of float	ee_sp[4]	LE	16	rw
17	array of float	ee_dsp[4]	LE	16	rw
33	array of float	ee_amplifier[3]	LE	12	rw

45	array of float	ee_offset[3]	LE	12	rw
57	array of float	ee_analog_in_min[2]	LE	8	rw
65	array of float	ee_analog_in_max[2]	LE	8	rw
73	array of float	ee_dac_min[2]	LE	8	rw
81	array of float	ee_dac_max[2]	LE	8	rw
101	array of BYTE	ee_relay_owner[4]	LE	4	rw
105	array of BYTE	ee_sensor_type[2]	LE	2	rw
107	array of BYTE	ee_math_processing_type[2]	LE	2	rw
109	array of BYTE	ee_compensation_state[2]	LE	2	rw
111	array of BYTE	ee_relay_logic[4]	LE	4	rw
115	array of BYTE	ee_dac_owner[2]	LE	2	rw
117	array of BYTE	ee_dac_type[2]	LE	2	rw
119	BYTE	ee_rs485_number	LE	1	rw
120	BYTE	ee_rs485_speed	LE	1	rw
121	BYTE	ee_rs485_byte_len	LE	1	rw
122	BYTE	ee_rs485_parity	LE	1	rw
123	BYTE	ee_rs485_stop_bits	LE	1	rw
124	BYTE	ee_plot_x_scale	LE	1	rw
125	BYTE	ee_dffs_rec_interval	LE	1	rw
126	BYTE	ee_access	LE	1	rw

127	BYTE	ee_dffs_rec_mode	LE	1	rw
128	BYTE	ee_work_ctl	LE	1	rw
129	BYTE	ee_work_rec	LE	1	rw
130	BYTE	ee_ext_keys_logic	LE	1	rw
131	array of BIT	ee_plot_visible_series[2]	LE	1	rw
132	array of BYTE	ee_median_filter_rank[3]	LE	3	rw
135	array of float	ee_integral[2]	LE	8	rw
144	rtc_t	ee_rtc_edit	LE	7	rw
152	array of BYTE	ee_led_n_f[2]	LE	2	rw
170	BYTE	ee_language	LE	1	rw
171	BYTE	ee_plot_y_autoscale	LE	1	rw
172	float	ee_plot_y_min	LE	4	rw
176	float	ee_plot_y_max	LE	4	rw
180	BYTE	ee_plot_y_min_n_f	LE	1	rw
181	BYTE	ee_plot_y_max_n_f	LE	1	rw
182	float	ee_pid_t	LE	4	rw
186	array of float	ee_pid_xp[4]	LE	16	rw
202	array of float	ee_pid_ti[4]	LE	16	rw
218	array of float	ee_pid_td[4]	LE	16	rw
234	array of float	ee_pid_min_power[4]	LE	16	rw
250	array of float	ee_pid_max_power[4]	LE	16	rw
266	array of float	ee_pid_alarm_power[4]	LE	16	rw

282	array of BYTE	ee_sensor_validate[2]	LE	2	rw
284	array of BYTE	ee_relay_on_alarm_state[4]	LE	4	rw
296	BYTE	ee_rtc_summer_time_mode_edit	LE	1	rw
297	WORD	ee_rtc_daily_time_correction	LE	2	rw

#### 5.4 Omix-P94(D4/W100)-MX-1

Таблица 5.5 Карта распределения памяти OMIX-P94-MX-1

Addr	Type	Name	Order	Size	rw
<b>Discrete inputs - std modbus function 2 (bit addressing mode)</b>					
0	array of BIT	global_relay_state[2]		2	r
<b>Holding registers - std modbus 3 &amp; 16 (word addressing mode)</b>					
0	BYTE	ee_rs485_number	BE	1	rw
1	BYTE	ee_rs485_speed	BE	1	rw
2	BYTE	ee_rs485_parity	BE	1	rw
3	BYTE	ee_rs485_stop_bits	BE	1	rw
4	BYTE	ee_rs485_byte_len	BE	1	rw
5	float	ee_voltage_ratio	BE	2	rw
7	float	ee_current_ratio	BE	2	rw
9	array of BYTE	ee_relay_logic[2]	BE	2	rw
11	array of reg set t	ee_reg_sets[7]	BE	56	rw
67	array of dac_t	ee_dacs[2]	BE	10	rw
77	BYTE	ee_access_type	BE	1	rw
78	BYTE	ee_show_variable	BE	1	rw
79	omix_rtc_t	ee_rtc_edit	BE	6	rw
85	BYTE	ee_rtc_summer_time_mode_edit	BE	1	rw

86	WORD	ee_rtc_daily_time_correction	BE	1	rw
87	BYTE	ee_dffs_rec_interval	BE	1	rw
88	BYTE	ee_dffs_rec_mode	BE	1	rw
89	array of rel set t	ee_rel_sets[7]	BE	28	rw
120	array of BYTE	ee_dac_type[2]	BE	2	rw
<b>Input registers - std modbus function 4 (word addressing mode)</b>					
0	array of float	channels_data[7]	BE	14	r
14	array of BYTE	global_relay_state[2]	BE	2	r
18	omix_rtc_t	rtc	BE	6	r
26	WORD_FP1	cur_voltage_wo_Ktr_int	BE	1	r
27	WORD_FP2	cur_current_wo_Ktr_int	BE	1	r
28	WORD_FP2	cur_frequency_int	BE	1	r
29	WORD_FP1	cur_app_power_wo_Ktr_int	BE	1	r
30	WORD_FP1	cur_act_power_wo_Ktr_int	BE	1	r
31	WORD_FP1	cur_rea_power_wo_Ktr_int	BE	1	r
32	WORD_FP2	cur_power_factor_int	BE	1	r

Однофазный прибор OMIX измеряет 7 параметров качества электроэнергии, в массивах памяти (array) они расположены в следующем порядке: [0]-напряжение, [1]-ток, [2] – частота, [3] – полная мощность, [4]- активная мощность, [5] – реактивная мощность, [6] – cos(Ф).

### 5.5 Спрут-Т(В)-06-RS485

Таблица 5.6 Карта распределения памяти Спрут-Т(В)-06-RS485

Addr	Type	Name	Order	Size	rw
<b>Coils - std modbus functions 1 &amp; 5 (bit addressing mode)</b>					
0	BIT	device_reset		1	rw
<b>Discrete inputs - std modbus function 2 (bit addressing mode)</b>					
0	array of BIT	relays_state[2]		2	r



2	BIT	sensor_disconnect		1	r
3	BIT	min_out_of_range		1	r
4	BIT	max_out_of_range		1	r
5	BIT	board_disconnect		1	r
6	BIT	err_adc_connect		1	r
<b>Holding registers – std modbus 3 &amp; 16 (word addressing mode)</b>					
0	array of rel_set_t	ee_rel_sets[2]	BE	8	rw
8	BYTE	ee_NSX_type	BE	1	rw
9	BYTE	ee_show_type	BE	1	rw
10	BYTE	ee_regulator_var1	BE	1	rw
10	BYTE	ee_regulator_var2	BE	1	rw
12	array of BYTE	ee_relay_logic[2]	BE	2	rw
13	float	ee_correction_t_k	BE	2	rw
16	float	ee_correction_t_b	BE	2	rw
18	BYTE	ee_rs485_number	BE	1	rw
19	BYTE	ee_rs485_speed	BE	1	rw
20	BYTE	ee_rs485_data_bit	BE	1	rw
21	BYTE	ee_rs485_parity	BE	1	rw
22	BYTE	ee_rs485_stop_bit	BE	1	rw
23	BYTE	ee_access_type	BE	1	rw
24	float	ee_correction_rh_k	BE	2	rw
26	float	ee_correction_rh_b	BE	2	rw
<b>Input registers – std modbus function 4 (word addressing mode)</b>					
0	float	temperature_value	BE	2	r
2	float	humidity_value	BE	2	r
4	array of BYTE	relays_state[2]	BE	2	r
6	BYTE	status_word	BE	1	r
7	WORD_FP1	temperature_value_int	BE	1	r
8	WORD_FP1	humidity_value_int	BE	1	r

## 5.6 Спрут-T-02-RS485-TC1/TC2

Таблица 5.7 Карта распределения памяти Спрут-T-02-RS485-TC1

Addr	Type	Name	Order	Size	rw
<b>Coils - std modbus functions 1 &amp; 5 (bit addressing mode)</b>					
9	BIT	device_reset		1	rw
16	BIT	default_network_settings		1	rw
<b>Discrete inputs - std modbus function 2 (bit addressing mode)</b>					
0	BIT	sensor_disconnect		1	r
1	BIT	min_out_of_range		1	r
2	BIT	max_out_of_range		1	r
3	BIT	err_adc_connect		1	r
<b>Holding registers - std modbus 3 &amp; 16 (word addressing)</b>					
0	BYTE	ee_NSX_type	BE	1	rw
1	float	ee_correction_k	BE	2	rw
3	float	ee_correction_b	BE	2	rw
5	BYTE	ee_rs485_number	BE	1	rw
6	BYTE	ee_rs485_speed	BE	1	rw
7	BYTE	ee_rs485_data_bit	BE	1	rw
8	BYTE	ee_rs485_parity	BE	1	rw
9	BYTE	ee_rs485_stop_bit	BE	1	rw
<b>Input registers - std modbus function 4 (word addressing mode)</b>					
0	float	temperature_value	BE	2	r
6	BYTE	status_word	BE	1	r
7	WORD_FP1	temperature_value_int	BE	1	r

## 5.7 Спрут-TB-03-RS485

Таблица 5.8 Карта распределения памяти Спрут-TB-03-RS485

Addr	Type	Name	Order	Size	rw
<b>Coils - std modbus functions 1 &amp; 5 (bit addressing mode)</b>					
9	BIT	device_reset		1	rw
16	BIT	default_network_settings		1	rw

<b>Holding registers - std modbus 3 &amp; 16 (word addressing)</b>					
0	BYTE	ee_rs485_number	BE	1	rw
1	BYTE	ee_rs485_speed	BE	1	rw
2	BYTE	ee_rs485_data_bit	BE	1	rw
3	BYTE	ee_rs485_parity	BE	1	rw
4	BYTE	ee_rs485_stop_bit	BE	1	rw
5	float	ee_correction_t_k	BE	2	rw
7	float	ee_correction_t_b	BE	2	rw
9	float	ee_correction_rh_k	BE	2	rw
11	float	ee_correction_rh_b	BE	2	rw
<b>Input registers - std modbus function 4 (word addressing mode)</b>					
0	float	temperature_value	BE	2	r
2	float	humidity_value	BE	2	r
4	BYTE	status_word	BE	1	r
5	WORD_FP1	temperature_value_int	BE	1	r
6	WORD_FP1	humidity_value_int	BE	1	r

## 5.8 Ординар

Таблица 5.9 Карта распределения памяти прибора Ординар

Addr	Type	Name	Order	Size	rw
<b>Coils - std modbus functions 1 &amp; 5 (bit addressing mode)</b>					
0	array of BIT	relay[3]		3	rw
9	BIT	device_reset		1	rw
13	BIT	firmware_update		1	rw
<b>Discrete inputs - std modbus function 2 (bit addressing mode)</b>					
0	array of BIT	level_conditions[4]		4	r
<b>Holding registers - std modbus 3 &amp; 16 (word addressing)</b>					
0	BYTE	ee_rs485_number	BE	1	rw
1	BYTE	ee_rs485_speed	BE	1	rw
2	BYTE	ee_rs485_parity	BE	1	rw

3	BYTE	ee_rs485_stop_bit	BE	1	rw
4	BYTE	ee_rs485_data_bit	BE	1	rw
5	WORD	algorithm_number	BE	1	rw
6	WORD	logic_mode	BE	1	rw
7	WORD	sensivity	BE	1	rw
8	WORD	bounce_check_delay	BE	1	rw
9	WORD	param_E	BE	1	rw
10	WORD	param_F	BE	1	rw
11	WORD	param_G	BE	1	rw
12	WORD	auto_manual_switch	BE	1	rw
<b>Input registers - std modbus function 4 (word addressing mode)</b>					
0	array of BYTE	relay[3]	BE	3	r

## 6 Обратная связь

Со всеми вопросами и предложениями обращайтесь по адресу электронной почты [support@automatix.ru](mailto:support@automatix.ru) или по телефонам:

(812) 327-32-74, (812) 928-32-74

Почтовый адрес: 195265, С-Петербург, а/я 71.

Офис, склад, выставка:

Санкт-Петербург, м. «Девяткино» (пос. Мурино),  
ул. Ясная, д. 11

Программное обеспечение и дополнительная информация могут быть найдены на нашем интернет сайте [www.automatix.ru](http://www.automatix.ru) или на сайте интернет-магазина [www.kipspb.ru](http://www.kipspb.ru).

## 7 Использованные источники информации

1. Electrical Characteristics of Balanced Voltage Digital Interface Circuits, ANSI/TIA/EIA-422-B-1994, Telecommunications Industry Association, 1994

2. Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems, ANSI/TIA/EIA-485-A-1998, Telecommunications Industry Association, 1998

3. Application Guidelines for TIA/EIA-485-A, TIA/EIA Telecommunications Systems Bulletin, Telecommunications Industry Association, 1998

4. A Comparison of Differential Termination Techniques, Joe Vo, National Semiconductor, Application Note AN-903
5. Data Transmission Design Seminar Reference Manual, 1998, Texas Instruments, literature number SLLE01
6. Data Transmission Line Circuits Data Book, 1998, Texas Instruments, literature number SLLD001
7. MODBUS Application Protocol Specification
8. MODBUS over serial line specification and implementation guide