



УТВЕРЖДЕН

ТИШЖ.468714.121 Д01-ЛУ

Линейный усилитель L/S-диапазона
с аттенюатором (RS-485/Ethernet)

Протокол информационно-логического взаимодействия

ТИШЖ.468714.121 Д01

Инв. № подл.		Подп. и дата	
Взам. инв. №		Инв. № дубл.	
Подп. и дата		Подп. и дата	

Перв. примен.	ТИШЖ.468714.121	СОДЕРЖАНИЕ	
		1. ОПИСАНИЕ ПРОТОКОЛА	3
Справ. №		2. СТРУКТУРА ПОСЫЛКИ	3
		3. ТИПЫ И СТРУКТУРА ЗАПРОСОВ (ПОЛЕ DATA)	4
		3.1. Команда на чтение регистра	4
		3.2. Ответ на команду чтения регистра	4
		3.3. Команда на запись регистра	5
		3.4. Ответ на команду записи	5
		4. СООБЩЕНИЯ ОБ ОШИБКАХ ОБМЕНА	6
		5. РЕГИСТРЫ ЛУ1К	7
		6. РАСЧЕТ КОНТРОЛЬНОЙ СУММЫ	11

Подп. и дата	Инв. № дубл.	Подп. и дата						
Взам. инв. №								
Подп. и дата								
Инв. №подл.	Разраб.	Орлов			ТИШЖ.468714.121 Д01 Линейный усилитель L/S-диапазона с аттенуатором (RS-485/Ethernet) Протокол информационно- логического взаимодействия	Лит.	Лист	Листов
	Пров.	Званцугов					2	12
	Т.контр.					ООО «Технологии Радиосвязи»		
	Н.контр.	Фадеев						
	Утв.	-						

Данный документ определяет протокол обмена данными по интерфейсу RS-485 между ЛИНЕЙНЫМ УСИЛИТЕЛЕМ 1-КАНАЛЬНЫМ (ЛУ1к) и устройством управления (УУ).

1. ОПИСАНИЕ ПРОТОКОЛА

Физический интерфейс: RS-485 двухпроводной.

Организация сети: ведущий - УУ, ведомый - ЛУ1К.

Инициировать передачу может только ведущий. Ведомый отвечает на запрос (если команда в запросе предполагает выдачу ответа)

Битовая структура данных: 8N2 (8 бит данных, без бита четности, два стоповых бита)

Скорость обмена: программируется. Возможные значения скорости передачи (бит/сек): 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 576000, 921600

Скорость обмена 115200 является скоростью по умолчанию (заводские установки)

Адресация:

Адреса ЛУ1К программируются. Допустимые значения адреса 0x01-0xFF.

Адрес 0xFF является циркулярным и может применяться только в пакете от УУ.

Пакеты с адресом 0xFF, воспринимаются всеми ЛУ1К.

Адрес 0 является запрещенным для ЛУ1К.

2. СТРУКТУРА ПОСЫЛКИ

Структура посылки передаваемой в прибор или принимаемой из прибора содержит следующие поля:

START	ADR_1	ADR_2	DATA	CRC	STOP
2 байта	1 байт	1 байт	N байт	2 байта	2 байта

Описание полей:

Поле START - флаг начала пакета. Содержит два байта 0xFE 0xFE.

Поле ADR_1 – адрес отправителя. Содержит 1 байт.

Поле ADR_2 – адрес получателя. Содержит 1 байт.

Поле DATA – данные пакета. Размер поля определяется типом запроса.

Поле CRC – контрольная сумма по полям START, ADR_1, ADR_2, DATA пакета.

Алгоритм вычисления контрольной суммы приведен в Приложении 1.

Изм.	Лист	№ докум.	Подпись	Дата

Подп. и дата

Инв.№дубл.

Взам. инв.№

Подп. и дата

Инв.№подл.

Поле STOP - флаг конца пакета. Содержит два байта 0xFC 0xFC.

Примечание 1: Если в полях ADR_1, ADR_2, DATA, CRC встречается байт 0xFE или 0xFC, то после него добавляется байт со значением равным 0x00. Соответственно, при приеме пакета этот байт из пакета изымается (байт-стаффинг).

Примечание 2: При передаче байт-стаффинг используется после расчета контрольной суммы. При приеме – сначала байт-стаффинг, потом расчет контрольной суммы.

3. ТИПЫ И СТРУКТУРА ЗАПРОСОВ (поле DATA)

3.1. Команда на чтение регистра

Команда «Чтение регистра»	Номер регистра
0x03	0xНННН
1 байт	2 байта

Где: 0x03 – код команды на чтение регистра

0xНННН – номер регистра (адресуемое пространство регистров 0x0000-0xFFFF)

3.2. Ответ на команду чтения регистра

Команда «Ответ на чтение регистра»	Номер регистра	Данные из регистра
0x04	0xНННН	Data_from_Registr
1 байт	2 байта	N байт

Где: 0x04 – код команды ответ на чтение регистра

0xНННН – номер регистра

Data_from_Registr - данные, считанные из регистра. Размер данных определяется номером регистра и может составлять до 255 байт.

Инва.№подгл.	Подгл. и дата
Взам. инв.№	Инва.№дубл.
Подгл. и дата	

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

3.3. Команда на запись регистра

Команда «Запись регистра»	Номер регистра	Данные в регистр
0x05	0xНННН	Data_In_Registr
1 байт	2 байта	N байт

Где: 0x05 – код команды на запись регистра

0xНННН – номер регистра

Data_In_Registr – данные на запись в регистр (до 255 байт)

3.4. Ответ на команду записи

Команда «Ответ на запись регистра»	Номер регистра	Данные из регистра
0x06	0xНННН	Data_from_Registr
1 байт	2 байта	N байт

Где: 0x06 – код команды ответ на запись регистра

0xНННН – номер регистра

Data_from_Registr - данные считанные из регистра после его записи (до 255 байт).

Примечание: Порядок следования байтов – младший бат передается первым.

Инв.№подл.	Подп. и дата	Инв.№дубл.	Подп. и дата	Взам. инв.№	Инв.№дубл.	Подп. и дата
------------	--------------	------------	--------------	-------------	------------	--------------

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

4. СООБЩЕНИЯ ОБ ОШИБКАХ ОБМЕНА

При ошибках обмена ЛУ1К высылает пакет со следующей структурой поля DATA

Команда «Признак ошибки»	Код ошибки
0x0A	0xНННН
1 байт	2 байта

Где: 0x0A – признак ошибки

0xНННН – код ошибки

Перечень кодов ошибок

Код ошибки	Что означает
0x02	Чтение регистра невозможно, либо регистр не найден
0x03	Запись в регистр невозможна, либо регистр не найден
0x04	Неудачная попытка чтения регистра
0x05	Неудачная попытка записи регистра
0x06	Неверное кол-во байтов в запросе в поле DATA при записи регистра

Инв.№подл.	Подп. и дата	Взам. инв.№	Инв.№дубл.	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата

ТИШЖ.468714.121 Д01

Лист

6

5. РЕГИСТРЫ ЛУ1К

Номер, дес	Признак	Описание регистра	Длина, байт
СТАТУСНЫЕ ПАРАМЕТРЫ			
0	R	<p>Регистр состояния ЛУ1К</p> <p>Байт 0 – общий статус ЛУ1К (тип unsigned char)</p> <p>Бит 0 – Флаг суммарной аварии 0 – нет аварии 1 – авария</p> <p>Бит 1 – Авария ЛУ1К «Ток потребления выше нормы» 0 – нет 1 – установлена</p> <p>Бит 2 – Авария ЛУ1К «Ток потребления ниже нормы» 0 – нет 1 – установлена</p> <p>Бит 3 – Состояние ЛУ1К канал 1 0 – выключен 1 – включен</p> <p>Бит 4-5 – зарезервировано Бит 6 – Авария Flash-памяти Бит 7 – Невалидный пользовательский ключ</p> <p>Байт 1 Значение аттенюатора ЛУ1К, дБ (тип unsigned char)</p> <p>Байт 2-3 Ток потребления ЛУ1К, мА Тип unsigned short (0-65535)</p>	4
1	R	<p>Регистр индикатора ЛУ1К</p> <p>Содержит 48 байтов индикатора ЛУ1К</p>	48
2	R	<p>Регистр состояния ЛУ1К+Регистр индикатора ЛУ1К</p> <p>Содержит байты регистра состояния R0 и 48 байтов индикатора ЛУ1К</p>	48+R0
3	R/W	<p>Регистр кнопок ЛУ1К (тип unsigned char)</p> <p>0 – кнопка ButtonNULL</p>	1

Инв.№подгл.	Подгл. и дата
Взам. инв.№	Инв.№дубл.
Подгл. и дата	Подгл. и дата

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ТИШЖ.468714.121 Д01

Лист

7

			1 – кнопка ButtonLeft 2 – кнопка ButtonUP 3 – кнопка ButtonRight 4 – кнопка ButtonDown 5 – кнопка ButtonOK 6 – кнопка ButtonRedit 7 – кнопка ButtonALARM 8 – кнопка ButtonKrest 9 – кнопка ButtonESCAPE 10 – кнопка ButtonAR 11-255 - зарезервировано	
--	--	--	---	--

ПАРАМЕТРЫ УПРАВЛЕНИЯ ЛУ1К

	4-8	R/W	Зарезервировано	1
	9	R/W	Байты 0-3 Текущие аварии ЛУ1К При чтении содержит битовую структуру текущих аварий ЛУ1К Бит 0- Ток ЛУ1К выше порога Бит 1- Ток ЛУ1К ниже порога Бит 2- Ошибка FLASH-памяти Бит 3-Невалидный пользовательский ключ При записи в этот регистр любого значения сбрасывает текущие аварии ЛУ1К (Журнал аварий при этом НЕ сбрасывается!) Тип unsigned long (4 байта)	8
	10	R/W	Байт 0 Включение питания ЛУ1К 0-выключено 1-включено (тип unsigned char)	1
	11-13	R/W	зарезервировано	-
	14	R/W	Байт 0 Значение аттенюатора ЛУ1К, дБ 0-30 Тип unsigned char	1
	15-24	R/W	зарезервировано	-
	25	R/W	Байты 0-1 Максимальный порог по току ЛУ1К мА Тип unsigned short (0-65535)	2
	26-29	R/W	зарезервировано	-
	30	R/W	Байты 0-1 Минимальный порог по току ЛУ1К мА Тип unsigned short (0-65535)	2
	31-42	R/W	зарезервировано	-

Инв.№подгл.	Подгл. и дата
	Инв.№дубл.
Взам. инв.№	Подгл. и дата
	Инв.№дубл.
Инв.№подгл.	Подгл. и дата
	Инв.№дубл.

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ТИШЖ.468714.121 Д01

Лист

8

	43	R/W	Байт 0 Скорость по UART в канале управления M&C 1 - 9600 2 - 19200 3 - 38400 4 - 57600 5 - 115200 6 - 230400 7 - 460800 8 - 500000 9 - 576000 10 – 921600 Тип unsigned char (0-255)	1
	44-62	R/W	Зарезервировано	-
	63	R/W	Адрес ЛУ1К Допустимые значения адреса 0x01-0xFF. Адрес 0xFF является циркулярным. Адрес 0 является запрещенным для ЛУ1К Тип unsigned char (0-255)	1
	64-78	R/W	Зарезервировано	-
	79	R/W	Байты 0-3 Журнал аварий ЛУ1К При чтении содержит битовую структуру регистра R9 аварий ЛУ1К При записи в этот регистр любого значения сбрасывает журнал текущих аварии ЛУ1К Тип unsigned long (4 байта)	4
	80 ... 65529	...	Зарезервировано	
	65530	W	Выставить параметры по умолчанию (запись 1 приводит к активации заводских настроек) Тип unsigned char (0-255)	1
	65531	R	Версия ПО Тип string[48]	48
	65532	R	ID-номер контроллера Тип unsigned long	4
	65533	R	Признак валидности пользовательского ключа 0-валиден 1-невалиден Тип unsigned char	1

Инв.№подл.	Подп. и дата
Взам. инв.№	Инв.№дубл.
Подп. и дата	Подп. и дата

Изм.	Лист	№ докум.	Подпись	Дата
------	------	----------	---------	------

ТИШЖ.468714.121 Д01

Лист

9

	65534	R/W	Пользовательский ключ 0XXXXXXXXX Тип unsigned long	4
	65535	R/W	Регистр перезагрузки РУ-L (запись в этот регистр вызывает перезагрузку РУ-L) Тип unsigned char (0-255)	1

Признак: **R** – только чтение, **W/R** – чтение и запись

Инв.№подл.	Подп. и дата				Инв.№дубл.	Подп. и дата				Лист																								
	Взам. инв.№					Инв.№дубл.																												
	Подп. и дата					Подп. и дата																												
Изм.					Лист					№ докум.					Подпись					Дата					ТИШЖ.468714.121 Д01					10				

6. РАСЧЕТ КОНТРОЛЬНОЙ СУММЫ

Примеры процедур расчета контрольной суммы по пакету на языке ANSI C приведены ниже.

```

unsigned int crc_chk(unsigned char* data, unsigned char length)
{//расчет контрольной суммы
int j;
unsigned int reg_crc=0xFFFF;
while(length--)
{
reg_crc ^= *data++;
for(j=0;j<8;j++)
{
if(reg_crc & 0x01) reg_crc=(reg_crc>>1) ^ 0xA001;
else reg_crc=reg_crc>>1;
}
}
return reg_crc;
}

```

Где: data – принятые данные, length – размер (длина) данных

Примеры процедур расчета контрольной суммы на языке Pascal по пакету приведены ниже.

```

function C485Modbus(unCRC_temp,unData:integer):integer;
//вспомогательная функция
Var LSB:integer;
i:integer;
begin
unCRC_temp:=((unCRC_temp xor unData) or $FF00) and (unCRC_temp or $FF);
for i:=1 to 8 do begin
LSB:=unCRC_temp and $1;
unCRC_temp:=unCRC_temp shr 1;
if (LSB<>0) then unCRC_temp:=unCRC_temp xor $A001;
end;//for i
C485Modbus:=unCRC_temp;
end;
//=====
function CRC_Modbus(LenDat:integer;DATAsend: array[1..100] of integer):integer;
//расчет контрольной суммы
Var CRC:word;
i:integer;
begin
CRC:=$FFFF;
for i:=1 to LenDat do CRC:=C485Modbus(CRC,DATAsend[i]);
CRC_Modbus:=CRC;
end;

```

Инв.№подл.	Подл. и дата	Взам. инв.№	Инв.№дубл.	Подл. и дата	ТИШЖ.468714.121 Д01	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

