



Технологии Радиосвязи

УТВЕРЖДЕН

ТИШЖ.468157.163 Д01-ЛУ

Тест-транслятор S-диапазона

Протокол информационно-логического взаимодействия

ТИШЖ.468157.163 Д01

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

Перв. примен.	ТИШЖ.468157.163
Справ. №	

Содержание

1	Описание протокола	3
2	Структура посылки.....	3
3	Типы и структура запросов	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 ТТ	7
6	Расчет контрольной суммы.....	11

Подп. и дата	
Инв. № дубл.	
Взам. инв.№	
Подп. и дата	
Инв. № подл.	
Разраб.	Орлов
Пров.	Касатов
Т.контр.	Званцугов
Н.контр.	Фадеев
Утв.	-

ТИШЖ.468157.163 Д01					
Изм	Лист	№ докум.	Подп.	Дата	
Блок тест-транслятора S-диапазона				Лит.	Лист
Протокол информационно-логического взаимодействия					2
					12

Данный документ определяет протокол обмена данными по интерфейсу RS-485 между блоком тест-транслятора (далее по тексту ТТ) и устройством управления (УУ).

1. Описание протокола

Физический интерфейс: RS-485 двухпроводной.

Организация сети: ведущий - УУ, ведомый - ТТ.

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 ТТ программируются. Допустимые значения адреса 0x01-0xFF.

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

Пакеты с адресом 0xFF, воспринимаются всеми ТТ.

Адрес 0 является запрещенным для ТТ.

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.

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

Изм.	Лист	№ докум.	Подпись	Дата	ТИШЖ.468157.163 Д01					Лист
										3

Поле CRC – контрольная сумма по полям START, ADR_1, ADR_2, DATA пакета. Алгоритм вычисления контрольной суммы приведен в разделе 6.

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.

Изм.	Лист	№ докум.	Подпись	Дата

Изм.	Лист	№ докум.	Подпись	Дата	ТИШЖ.468157.163 Д01	Лист
						4

3.3. Команда на запись регистра

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

Где: 0x05 – код команды на запись регистра,

0хНННН – 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	0хНННН	Data_from_Registr
1 байт	2 байта	N байт

Где: 0x06 – код команды ответ на запись регистра,

0хНННН – номер регистра,

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

Примечание: Порядок следования байтов – младший бат передается первым.

Инд. № подл.					Подп. и дата	
						Инд. № дубл.
						Подп. и дата
					Лист	
ТИШЖ.468157.163 Д01					5	
Изм.	Лист	№ докум.	Подпись	Дата		

4. Сообщения об ошибках обмена

При ошибках обмена ТТ высылает пакет со следующей структурой поля DATA

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

Где: 0x0A – признак ошибки

0хНННН – 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

Инд. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата	Изм.	Лист	№ докум.	Подпись	Дата	ТИШЖ.468157.163 Д01	Лист
											6

5. Регистры ТТ

Номер, дес	При знак	Описание регистра	Длина, байт
СТАТУСНЫЕ ПАРАМЕТРЫ			
0	R	<p><u>Регистр состояния ТТ</u></p> <p>Байт 0 – статус ТТ (тип unsigned char) Бит 0 – Общая Авария 0 – нет аварии 1 – авария Бит 1 – Авария нет захвата 10 МГц 0 – нет аварии - захват (LOCK) 1 – авария - нет захвата (UNLOCK) Бит 2 – MUTE 0 – UNMUTE 1 – MUTE Бит 3 – Опора 10 MHz 0 – внутренняя 1 – внешняя Бит 4 – Выход ТТ 0 – на антенну шлейфового контроля 1 – на направленный ответвитель Бит 5 – Авария Flash-памяти Бит 6 – Невалидный пользовательский ключ Бит 7 – зарезервировано</p> <p>Байт 1 – аттенюатор ТТ (тип unsigned char)</p>	2
1	R	<p><u>Регистр индикатора ТТ</u></p> <p>Содержит 48 байтов индикатора ТТ</p>	48
2	R	<p><u>Регистр состояния ТТ+Регистр индикатора ТТ</u></p> <p>Содержит 6 байт регистра состояния R0 и 48 байтов индикатора ТТ</p>	48+2
3	R/W	<p><u>Регистр кнопок ТТ</u> (тип unsigned char)</p> <p>0 – кнопка ButtonNULL 1 – кнопка ButtonLeft 2 – кнопка ButtonUP 3 – кнопка ButtonRight</p>	1

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

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

ТИШЖ.468157.163 Д01

Лист

7

			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 ТТ

	4	R/W	Байт 0 Вкл/выкл MUTE Управление переключением MUTE 0 – UNMUTE 1 – MUTE Чтение – считывается текущее состояние Запись – MUTE переводится в заданное состояние (тип unsigned char)	1
	5	R/W	Байт 0 Управление переключением опоры 10 МГц 0 – внутренняя 1 – внешняя Чтение – считывается текущее состояние Запись – переводится в заданное состояние (тип unsigned char)	1
	6	R/W	Байт 0 Управление выходом ТТ 0 – на антенну шлейфового контроля 1 – БУП в состоянии 1 Чтение – считывается текущее состояние Запись – на направленный ответвитель (тип unsigned char)	1
	7	R/W	Байт 0 аттенюатор ТТ Задаёт коэффициент ослабления аттенюатора в ТТ Значения от 0 до 63 дБ (тип unsigned char)	1

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

Изм.	Лист	№ докум.	Подпись	Дата	ТИШЖ.468157.163 Д01	Лист
						8

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

	8	R/W	зарезервировано	-
	9	R/W	Байты 0-3 Текущие аварии ТТ При чтении содержит битовую структуру текущих аварий ТТ Бит 0- Авария нет захвата 10 МГц Бит 1- Авария Flash-памяти Бит 2- Невалидный пользовательский ключ При записи в этот регистр любого значения сбрасывает текущие аварии ТТ (Журнал аварий при этом НЕ сбрасывается!) Тип unsigned long (4 байта)	8
	10-42	R/W	зарезервировано	-
	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 ТТ Допустимые значения адреса 0x01-0xFF. Адрес 0xFF является циркулярным. Адрес 0 является запрещенным для ТТ Тип unsigned char (0-255)	1
	64-78	R/W	Зарезервировано	-
	79	R/W	Байты 0-3 Журнал аварий ТТ При чтении содержит битовую структуру журнала аварий ТТ Бит 0- Авария нет захвата 10 МГц Бит 1- Авария Flash-памяти Бит 2- Невалидный пользовательский ключ	4

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

ТИШЖ.468157.163 Д01

Лист

9

			При записи в этот регистр любого значения сбрасывает журнал текущих аварии ТТ Тип unsigned long (4 байта)	
	80 ... 65530	...	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
	65534	R/W	Пользовательский ключ 0XXXXXXXXX Тип unsigned long	4
	65535	R/W	Регистр перезагрузки ТТ (запись в этот регистр вызывает перезагрузку ТТ) Тип unsigned char (0-255)	1

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

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

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

ТИШЖ.468157.163 Д01

Лист

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;
    }//for j
  }//while(length--)
  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;

```

Изм.	Лист	№ докум.	Подпись	Дата	ТИШЖ.468157.163 Д01	Лист
						11

