

УТВЕРЖДЕН
RU.СГМА.00060 94 01-ЛУ

СТАНЦИОННЫЕ УСТРОЙСТВА САУТ-ЦМ/НСП

Программное обеспечение

Еbilock. Протокол увязки с внешней системой САУТ

RU.СГМА.00060 94 01

Листов 14

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дудл.	Подп. и дата
03.14.049	 28.12.2016			

2017

Литера А

Обмен информацией между внешней системой САУТ и Ebilock 950 R4 осуществляется через пару шлюзов IPU_GATE_RF.

Обмен данными между шлюзом IPU_GATE_RF и внешней системой производится по интерфейсам RS-232/422/485 с использованием пакетов, формат которых описан в гл. 7 данного руководства. Более того перед передачей информации, каждый пакет должен быть дополнительно закодирован методом HDLC для однозначной идентификации пакета в потоке байт. При этом пара безопасных телеграмм А/В для одного ОК передается в одном пакете.

Обмен между IPU_GATE_RF и Ebilock 950 R4 осуществляется через Ethernet, путём включения IPU_GATE_RF в сеть объектных контроллеров с использованием стандартного протокола передачи данных.

Структурная схема коммуникации Ebilock 950 R4 и внешней системы, когда внешняя система не обеспечивает резервирование каналов связи, приведена ниже (рисунок 1):

В этом случае обмен внешней системы с Ebilock 950 R4 сводится к взаимодействию двух блоков коммуникации внешней системы с двумя IPU_GATE_RF, при этом каждый блок внешней системы соединён с каждым IPU_GATE_RF.

По одному RS-интерфейсу может передавать информация не более чем о 16 объектных контроллеров.

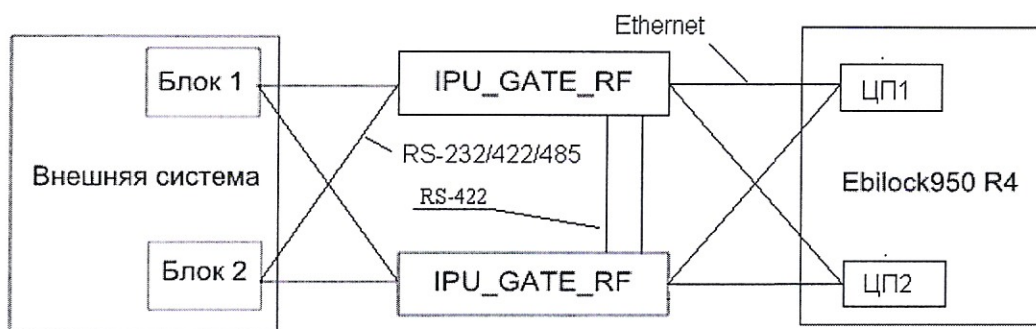


Рисунок 1

В случае когда, внешняя система обеспечивает внутреннее резервирование каналов связи, применима следующая схема (рисунок 2):

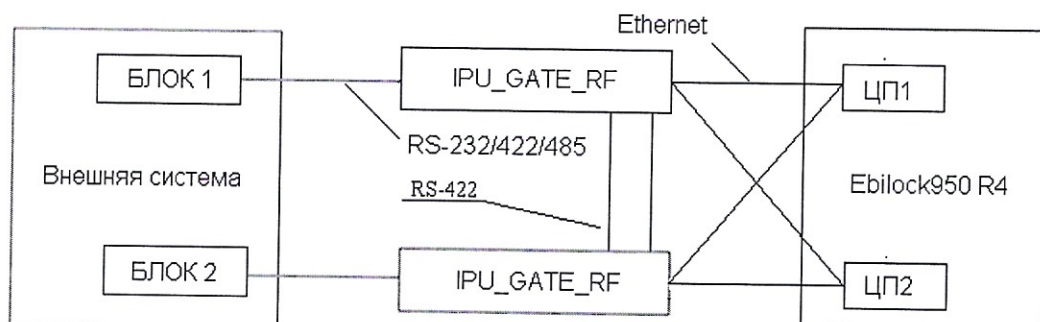


Рисунок 2

В данной ситуации один блок внешней системы соединяется с одним блоком IPU_GATE_RF.

Характеристики линий связи

Для взаимодействия IPU_GATE_RF и внешней системы используются линии связи RS-232/422/485. При передаче данных используются следующие параметры соединения: 8 бит данных, отсутствуют управляющие сигналы и контроль передачи, скорость передачи данных устанавливается в 115200 бод.

Процедура коммуникации

Обмен между IPU_GATE_RF и Ebilock 950 R4

После того как соединение между шлюзами IPU_GATE_RF и Ebilock 950 R4 установилось, Ebilock 950 R4 начинает передавать накачку (приказ) для внешней системы на активный шлюз IPU_GATE_RF. Активный шлюз IPU_GATE_RF, передаёт приказ внешней системе и пассивному шлюзу. Отправка приказа от Ebilock 950 R4 производится раз в 0.6 с. Пассивный шлюз передаёт приказы внешней

системе в случае, если он получил накачку от активного шлюза, соответственно, приказ от пассивного шлюза передаётся с незначительной (~10мс) задержкой.

На следующем такте активный шлюз может стать пассивным, а пассивный активным. Выбор активного и пассивного шлюза ложится на ЦП, в нормальной ситуации используется для передачи данных поочерёдно то один, то другой шлюз. Это обусловлено необходимостью уменьшения задержек/сбоев при выходе из строя одного из IPU_GATE_RF или линий связи.

Как только все внешние системы предадут статусы, активный IPU_GATE_RF пересылает эту информацию Ebilock 950 R4. Если все статусы по каким-то причинам не пришли (нет связи с одной из систем или с одним из контроллеров), то через 150мс IPU_GATE_RF передаст Ebilock 950 R4 только те статусы, которые успели придти.

Обмен между внешней системой и IPU_GATE_RF

При включении внешняя система устанавливает счётчик циклов для телеграмм А в 0x00, а для телеграмм В в 0xff, все контроллеры переводятся в безопасное состояние.

При получении корректной телеграммы приказа (верные идентификаторы, контрольная сумма и др.) статусная информация в виде пары безопасных телеграмм А/В передается по той линии связи, по которой пришел приказ.

Также при получении пустого приказа по одной из линий связи (не содержащего телеграмм А/В и счетчиков А/В), внешняя система должна передать по данной линии связи текущие статусы своих контроллеров, с текущими счётчиками А/В.

По каждому каналу связующему IPU_GATE_RF и блок внешней системы в пакете приказа/статуса для объектного контроллера передается пара безопасных телеграмм А/В. Нет каналов выделенных только для телеграмм А или только для телеграмм В.

Если внешняя система не получает корректного приказа в течении 1.5 с, то все контроллеры переводятся в безопасное состояние.

Если внешняя система получает корректные пакеты от IPU_GATE_RF, но в этих пакетах для одного из объектных контроллеров нет приказа более чем 1.5 с, то этот контроллер переводится в безопасное состояние, а остальные работают в соответствии с полученным приказом.

Формат сетевого пакета

Сетевой пакет, используемый при обмене информацией между IPU_GATE_RF и внешней системой является бинарным и включает в себя три блока, как показано на ниже

Заголовок	Тело пакета	Контрольная сумма
-----------	-------------	-------------------

Более того, если используется обмен данными между IPU_GATE_RF и внешней системой через интерфейсы семейства RS-XXX, перед передачей пакет дополнительно кодируется по HDLC.

При отправке пакета его байты передаются последовательно: первый, второй и т.д.

Заголовок сетевого пакета

Заголовок сетевого пакета включает в себя следующую информацию – идентификатор отправителя, идентификатор получателя, тип пакета, длина пакета, нулевой байт. При упаковке многобайтовых полей сначала упаковываются старшие байты. Структура заголовка пакета приведена ниже:

Id отправителя 1 байт	Id получателя 1 байт	Тип пакета 1 байт	Длина пакета 4 байта	Нулевой байт 1 байт
--------------------------	-------------------------	----------------------	-------------------------	------------------------

Для длины пакета существует ограничение в 2 КБ, если пришло значение с длиной большей, то пакет считается повреждённым.

Таблица идентификаторов отправителей и получателей:

Идентификатор	Описание
0	IPU_GATE_RF
1	Внешняя система

Таблица типов пакетов:

Типа пакета	Идентификатор	Отправитель	Получатель
Передача накачки	2	IPU_GATE_RF	Внешняя система
Передача статусов	3	Внешняя система	IPU_GATE_RF
Передача пустой накачки	4	IPU_GATE_RF	Внешняя система

Контрольная сумма

В качестве контрольной суммы используется CRC-16 (блок контрольной суммы занимает 2 байта) с начальным значением 0xffff, вычисленная по следующей таблице: static uint16_t crc_ccitt_tab[] =

```
{  
    0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5, 0x60c6, 0x70e7,  
    0x8108, 0x9129, 0xa14a, 0xb16b, 0xc18c, 0xd1ad, 0xe1ce, 0xf1ef,  
    0x1231, 0x0210, 0x3273, 0x2252, 0x52b5, 0x4294, 0x72f7, 0x62d6,  
    0x9339, 0x8318, 0xb37b, 0xa35a, 0xd3bd, 0xc39c, 0xf3ff, 0xe3de,  
    0x2462, 0x3443, 0x0420, 0x1401, 0x64e6, 0x74c7, 0x44a4, 0x5485,  
    0xa56a, 0xb54b, 0x8528, 0x9509, 0xe5ee, 0xf5cf, 0xc5ac, 0xd58d,  
    0x3653, 0x2672, 0x1611, 0x0630, 0x76d7, 0x66f6, 0x5695, 0x46b4,  
    0xb75b, 0xa77a, 0x9719, 0x8738, 0xf7df, 0xe7fe, 0xd79d, 0xc7bc,  
    0x48c4, 0x58e5, 0x6886, 0x78a7, 0x0840, 0x1861, 0x2802, 0x3823,  
    0xc9cc, 0xd9ed, 0xe98e, 0xf9af, 0x8948, 0x9969, 0xa90a, 0xb92b,  
    0x5af5, 0x4ad4, 0x7ab7, 0x6a96, 0x1a71, 0x0a50, 0x3a33, 0x2a12,  
    0xdbfd, 0xcbbc, 0xfbff, 0xeb9e, 0x9b79, 0x8b58, 0xbb3b, 0xab1a,  
    0x6ca6, 0x7c87, 0x4ce4, 0x5cc5, 0x2c22, 0x3c03, 0x0c60, 0x1c41,  
    0xedae, 0xfd8f, 0xcdec, 0xddcd, 0xad2a, 0xbd0b, 0x8d68, 0x9d49,  
    0x7e97, 0x6eb6, 0x5ed5, 0x4ef4, 0x3e13, 0x2e32, 0x1e51, 0x0e70,  
    0xff9f, 0xefbe, 0xdfdd, 0xcffc, 0xbf1b, 0xaf3a, 0x9f59, 0x8f78,  
    0x9188, 0x81a9, 0xb1ca, 0xa1eb, 0xd10c, 0xc12d, 0xf14e, 0xe16f,
```

RU.CTMA.00060 94 01

0x1080, 0x00a1, 0x30c2, 0x20e3, 0x5004, 0x4025, 0x7046, 0x6067,
0x83b9, 0x9398, 0xa3fb, 0xb3da, 0xc33d, 0xd31c, 0xe37f, 0xf35e,
0x02b1, 0x1290, 0x22f3, 0x32d2, 0x4235, 0x5214, 0x6277, 0x7256,
0xb5ea, 0xa5cb, 0x95a8, 0x8589, 0xf56e, 0xe54f, 0xd52c, 0xc50d,
0x34e2, 0x24c3, 0x14a0, 0x0481, 0x7466, 0x6447, 0x5424, 0x4405,
0xa7db, 0xb7fa, 0x8799, 0x97b8, 0xe75f, 0xf77e, 0xc71d, 0xd73c,
0x26d3, 0x36f2, 0x0691, 0x16b0, 0x6657, 0x7676, 0x4615, 0x5634,
0xd94c, 0xc96d, 0xf90e, 0xe92f, 0x99c8, 0x89e9, 0xb98a, 0xa9ab,
0x5844, 0x4865, 0x7806, 0x6827, 0x18c0, 0x08e1, 0x3882, 0x28a3,
0xcb7d, 0xdb5c, 0xeb3f, 0xfb1e, 0x8bf9, 0x9bd8, 0xabbb, 0xbb9a,
0x4a75, 0x5a54, 0x6a37, 0x7a16, 0x0af1, 0x1ad0, 0x2ab3, 0x3a92,
0xfd2e, 0xed0f, 0xdd6c, 0xcd4d, 0xbdaa, 0xad8b, 0x9de8, 0x8dc9,
0x7c26, 0x6c07, 0x5c64, 0x4c45, 0x3ca2, 0x2c83, 0x1ce0, 0x0cc1,
0xef1f, 0xff3e, 0xcf5d, 0xdf7c, 0xaf9b, 0xbfba, 0x8fd9, 0x9ff8,
0x6e17, 0x7e36, 0x4e55, 0x5e74, 0x2e93, 0x3eb2, 0x0ed1, 0x1ef0

};

С использованием следующей функции:

```
static uint16_t GetCrc( uint16_t wCrc , BYTE* pBuf , int nSize )
```

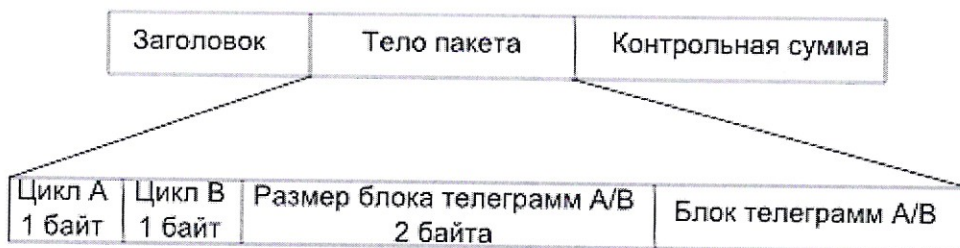
```
{  
    WORD crc = wCrc;  
    while( nSize-- ) crc = (WORD)( (crc << 8) ^ crc_ccitt_tab[ ( crc >> 8 ) ^ *pBuf++ ] );  
    return crc;  
}
```

В контрольной сумме учитываются заголовок пакета и его тело.

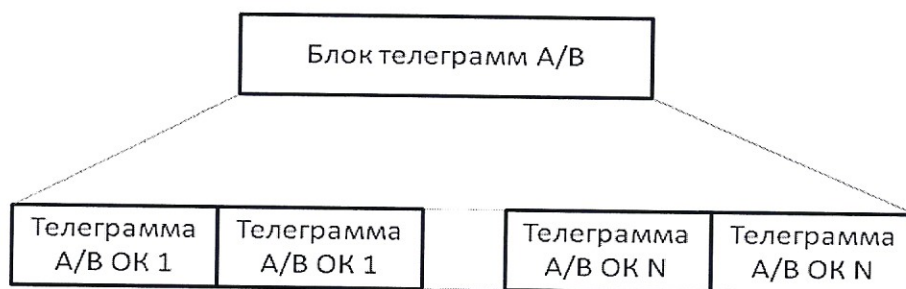
Телеграмма передачи приказа от Ebilock 950 R4

Ebilock 950 R4 на каждом цикле (продолжительность цикла 0.6 секунды) отправляет активному IPU_GATE_RF накачку. Как только активный IPU_GATE_RF получает накачку, он немедленно передаёт этот приказ внешней системе и пассивному шлюзу. Если IPU_GATE_RF соединён с внешней системой более чем по одному каналу связи, то идентичный приказ передается по всем подсоединённым каналам.

Телеграмма передачи накачки (приказа) имеет следующий формат:



В блоке безопасных телеграмм для каждого используемого объектного контроллера должны быть сохранены обе безопасные телеграммы, т.е. и телеграмма А и телеграмма В. В блоке телеграмм телеграмма А и телеграмма В для каждого объектного контроллера располагаются последовательно:



Значения счётчиков циклов А и В должны совпадать со значениями соответствующих счётчиков циклов в безопасных телеграммах А и В соответственно. Если нет совпадения, тогда отбрасывается безопасная телеграмма с не совпавшим счётчиком.

При приходе телеграммы от IPU_GATE_RF внешняя система первоначально проверяет контрольную сумму телеграммы передачи приказа, идентификаторы отправителя и получателя, тип пакета. Если найдено несоответствие, то телеграмма отбрасывается и не учитывается (считается, что она не приходила).

На следующем этапе выполняется проверка счётчиков циклов А (ct_a) и циклов В (ct_b) телеграммы передачи приказа, для которых должны быть выполнены следующие условия:

1. Значения ct_a увеличивается на 1 с каждым новым приказом, допустимо увеличение на 2;
2. Значение ct_b уменьшается на 1 с каждым новым приказом, допустимо уменьшения на 2;
3. Выполнено условие согласованности циклов $ct_a + ct_b = 0xff$

Если хоть одно из условий для счётчиков циклов ct_a/ct_b не выполнено в телеграмме приказа, то внешняя система считает, что до этого был разрыв связи.

Приказы из этой телеграммы не выполняются, а в ответ передается статусная информация о всех объектных контроллерах со старыми значениями счётчиков цикла, после чего устанавливаются новые счётчики циклов.

Если условия для ct_a/ct_b выполнены в телеграмме приказа, тогда внешняя система устанавливает новые значения счётчиков циклов для безопасных телеграмм, после чего передаёт статусную информацию о всех ОК с новыми значениями счётчиков циклов телеграммах A/B, после чего приступает к выполнению приказов, т.е. обрабатывает телеграммы A/B значения счётчиков циклов у которых совпадают с полученными ct_a/ct_b (конечно, для телеграмм A проверяется наличие корректной телеграммы B и только в случае корректности приказ выполняется).

В ответ на телеграмму приказа со счётчиками цикла A/B, внешняя система в нормальной ситуации должна послать телеграмму статусов со счётчиками A-1 и B+1. При восстановлении после разрыва связи, допускается первую телеграмму статуса послать со счётчиками по умолчанию (0x00 и 0xFF). Телеграмма статуса может быть подготовлена в ответ на приказ на предыдущем цикле — тогда счётчики берутся из приказа на предыдущем цикле, или как ответ на приказ на текущем цикле — тогда счётчики берутся A-1 и B+1.

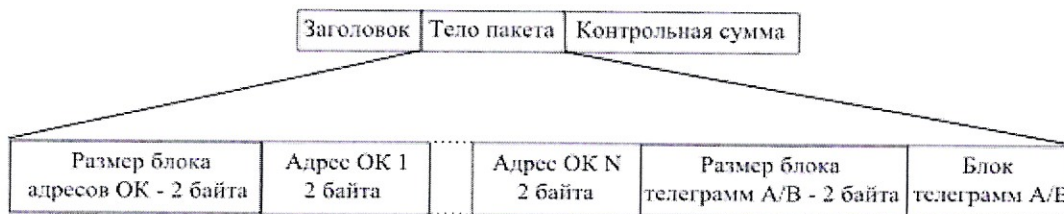
Если один из объектных контроллеров не получал корректную пару безопасных телеграмм A/B более чем 1.5 секунды, то он должен быть переведён в безопасное состояние, а остальные контроллеры работают в соответствии с полученными приказами.

В ответ на пустой приказ, в котором отсутствует тело пакета приказа и имеется соответствующий идентификатор в заголовке пакета, внешняя система должна послать статусную телеграмму с текущими счётчиками A/B по той же линии.

Телеграмма передачи статуса для Ebilock 950 R4

Телеграмма передачи статуса имеет следующий формат:

ИНВ. № 03.14.049
ПОДП. И ДАТА Шифр-дв.16.08.2016



Такая структура статусной телеграммы позволяет избежать необходимости посылать дополнительные конфигурационные телеграммы. В блоке телеграмм А/В для каждого используемого объектного контроллера должны быть сохранены обе безопасные телеграммы, т.е. и телеграмма А и телеграмма В.

Число объектных контроллеров вычисляется как значение сохранённое в поле «Размер блока адресов ОК» делённое на 2.

Статусная телеграмма может передаваться как один раз для полной передачи всех статусов ОК, так и для небольшой группы. Хотя, использование одной телеграммы, передающей все статусы является более эффективным, с точки зрения объёма передаваемых байт. Если объектный контроллер не доступен, то безопасные телеграммы А/В со статусной информацией о его состоянии не передаются (Ebilock 950 R4 автоматически выставит состояние нет связи для этого ОК, если в течении 2-х тактов от него не будет приходить статусная информация).

В случае когда объектный контроллер не получает накачки более чем 1.5 с, он должен быть переведён в безопасное состояние, все остальные контроллеры действуют в соответствии с полученным приказом.

Если объектный контроллер получает телеграмму накачки с некорректным блоком данных, он также должен быть переведён в безопасное состояние.

Временные ограничения при передаче статусов.

Статус должен быть передан немедленно в ответ на приказ. Шлюз IPU_GATE_RF имеет общее ограничение времени в 100 мс на ожидание статуса от внешней системы после передачи приказа, соответственно, статус должен быть полностью передан за это время.

Промежуточное хранение статусов ОК концентратором внешней системы.

В некоторых внешних системах существует внутренняя шина данных ограниченной пропускной способности, связывающая собственно безопасные объектные контроллеры (ОК) внешней системы и концентратор информации (КЦ) внешней системы. В таком случае допустимо использование КЦ внешней системы для временного хранения статусов ОК на протяжении цикла передачи данных.

Диаграмма передачи данных выглядит при этом следующим образом:

Время	ОК внешней системы	КЦ внешней системы	Шлюз IPU_GATE_RF	Ebilock950 R4
От 100мс до 300мс в зависимости от наличия помех при передаче. Обычное значение — 100 мс.				Начало цикла 1. Приказ послан из ЦП Ebilock950
			Приказ цикла 1 получен шлюзом и передан на внешнюю систему	
		Концентратор передаёт полученные ранее от ОК статус (счётчики - от цикла 0) шлюзу IPU_GATE_RF, а также передаёт полученный приказ на ОК		
			Статус передаётся на ЦП Ebilock950	
От 300мс до 500мс. Обычное значение — 500 мс.	ОК обрабатывает полученный приказ цикла 1			ЦП Ebilock950 обрабатывает статус.
	ОК передаёт текущий статус для цикла 1 на концентратор	КЦ получает статус для цикла 1 и буферизирует его		
	ОК передаёт текущий статус для цикла 1 на концентратор	КЦ получает статус для цикла 1 и буферизирует его		
	ОК передаёт текущий статус для цикла 1 на концентратор	КЦ получает статус для цикла 1 и буферизирует его		
	ОК передаёт текущий статус для цикла 1 на концентратор	КЦ получает статус для цикла 1 и буферизирует его		Начало цикла 2. Приказ послан из ЦП Ebilock950
	ОК передаёт текущий статус для цикла 1 на концентратор	КЦ получает статус для цикла 1 и буферизирует его	Приказ цикла 2 получен шлюзом и передан на внешнюю систему	
		Концентратор передаёт полученные ранее от ОК статус (счётчики - от цикла 1) шлюзу IPU_GATE_RF, а также передаёт полученный приказ цикла 2 на ОК		

Таким образом, ЦП получает статусы, сформированные объектным контроллером за время Т до получения очередного приказа. Время Т определяется внутренним циклом объектных контроллеров внешней системы. Это время должно быть меньше времени цикла как минимум вдвое. В системе ОК OCS950, используемых штатно в системе Ebilock950, время цикла объектных контроллеров — 110мс.

Предполагается, что ОК внешней системы не просто повторяет концентратору сформированную при получении приказа телеграмму статуса, а должен генерировать статус в зависимости от своего текущего состояния. Используя при этом счётчики А и В из последнего полученного приказа.

Например, в системе объектных контроллеров OCS950 состояние контактов реле опрашивается каждые 110мс. Соответственно, ЦП Ebilock950 получает информацию о переключении контролируемого реле с задержкой в 110мс + 600мс.

Для систем, которые не содержат внутреннего концентратора информации и непосредственно генерируют статус в ответ на полученный приказ (без задержки или с задержкой в сотые доли секунды), данная диаграмма неприменима. Такие системы должны генерировать статус со счётчиками А-1 и В+1 в ответ на приказ со счётчиками А и В.

Правила упаковки данных

При хранении данных, биты нумеруются следующим образом

Нумерация битов в байте							
7	6	5	4	3	2	1	0
Старший							Младший

Пусть есть двухбайтовое значение А, которые представляется двумя байтами А1 и А2, где А1 – старший байт (15-8 биты), а А2 – младший байт (7-0 биты). Тогда А, будет сохранено в последовательности битов следующим образом:

$$\boxed{A} = \boxed{A1} \boxed{A2}$$

При сохранении четырёхбайтного значения В аналогичном образом сначала сохраняется сначала старший байт В1 (31-24 биты), затем В2 (23-16 биты), В3 (15-8 биты) и В4 (7-0 биты), как показано ниже:

RU.CFMA.00060 94 01

B	=	B1	B2	B3	B4
---	---	----	----	----	----

При хранении многобайтовых данных байты сохраняются по старшинству от старших к младшим.

В данном документе, были рассмотрены основные концепции взаимодействия внешней системы с Ebilock 950 R4, которое осуществляется по средствам пары шлюзов IPU_GATE_RF, соединённых с парой блоков-коммуникаторов внешней системы, по принципу каждый с каждым, что позволяет осуществлять резервированную передачу данных по независимым физическим линиям.

Передача данных осуществляется по бинарному протоколу, пакеты которого при передаче по RS интерфейсам дополнительно кодируются с применением метода HDLC, обеспечивающего однозначную идентификацию пакета в потоке байт.

При взаимодействии используется простая процедура коммуникации и достаточно простой формат пакетов данных, что делает её достаточно легко реализуемой при минимальных усилиях разработчиков внешней системы.

ИНВ. № 03.14.049
ПОДП. И ДАТА 28.12.2016

[illegible]