



РАДИОМОДУЛЬ ВВОДА-ВЫВОДА

# СПЕКТР 868 iO

*Руководство по эксплуатации*

**БАКП.464421.002 РЭ**

Версия Руководства: **01.02**  
Последнее изменение: 06.03.2023



**ООО «РАТЕОС»** Москва, Зеленоград  
+7(499) 990-9716, [www.rateos.ru](http://www.rateos.ru), [rateos@rateos.ru](mailto:rateos@rateos.ru)

© **ООО «РАТЕОС»**

Все права защищены

ООО «РАТЕОС» прилагает все усилия для того, чтобы информация, содержащаяся в этом документе, являлась точной и надёжной. Однако, ООО «Ратеос» не несет ответственности за возможные неточности и несоответствия информации в данном документе, а также сохраняет за собой право на изменение информации в этом документе в любой момент без уведомления. Для получения наиболее полной и точной информации ООО «Ратеос» рекомендует обращаться к последним редакциям документов на сайте [www.rateos.ru](http://www.rateos.ru).

ООО «Ратеос» не несет ответственности за возможный прямой и косвенный ущерб, связанный с использованием своих изделий.

ООО «Ратеос» не передаёт никаких прав на свою интеллектуальную собственность.

Все торговые марки, упомянутые в данном документе, являются собственностью их владельцев.



## Содержание

<b>1</b>	<b>ИСТОРИЯ ВЕРСИЙ</b> .....	<b>7</b>
<b>2</b>	<b>ОБЩАЯ ИНФОРМАЦИЯ</b> .....	<b>8</b>
2.1	Назначение и общие сведения .....	8
2.2	Основные принципы работы модуля.....	8
2.3	Сопутствующее оборудование и материалы.....	8
<b>3</b>	<b>РАЗЪЁМЫ, ОРГАНЫ УПРАВЛЕНИЯ И ИНДИКАЦИИ</b> .....	<b>9</b>
3.1	Индикаторы.....	9
3.1.1	Индикатор <i>MODE</i> .....	9
3.1.2	Индикатор <i>RX/TX</i> .....	10
3.2	Разъёмы, клеммы, кнопка .....	10
3.2.1	Сброс модуля. Индикация версии ПО .....	11
<b>4</b>	<b>УСТАНОВКА И ПОДКЛЮЧЕНИЕ ВНЕШНЕГО ОБОРУДОВАНИЯ</b> .....	<b>12</b>
4.1	Установка модуля.....	12
4.2	Подключение антенн .....	12
4.2.1	Герметизация разъёмных соединений.....	12
4.3	Питание модуля.....	13
4.4	Интерфейсы RS-485, USB.....	14
4.4.1	Интерфейс <i>RS-485</i> .....	14
4.4.2	Интерфейс <i>USB</i> . Установка драйвера шины <i>USB</i> на ПК .....	15
4.4.3	Индикация ошибок при работе по последовательным портам .....	15
<b>5</b>	<b>НАСТРОЙКА МОДУЛЕЙ</b> .....	<b>16</b>
5.1	Конфигурация параметров.....	16
5.1.1	Подключение к ПК для конфигурации .....	16
5.2	Работа модулей в качестве прозрачного радиоудлинителя RS-485/USB.....	16
5.2.1	Основные принципы.....	16
5.2.2	Параметры активного порта ( <i>\$COM</i> ) .....	17
5.2.3	Скорость обмена данными между модулями в эфире ( <i>\$AR</i> ) .....	17
5.2.4	Рабочий канал ( <i>\$CH</i> ).....	18
5.2.5	Параметры входящих данных <i>RS485/USB</i> для передачи в эфир ( <i>\$DAT</i> ).....	18
5.3	Проверка модулей и стыковка с оборудованием перед установкой на объект .....	19
<b>6</b>	<b>РЕЖИМЫ РАБОТЫ МОДУЛЯ</b> .....	<b>20</b>
6.1	Режим загрузчика .....	20
6.1.1	Вход в командный режим по умолчанию ( <i>BOOT_DEF</i> ).....	20
6.1.2	Режим Тест ( <i>BOOT_TEST</i> ).....	20
6.1.3	Режим смены программного обеспечения ( <i>BOOT_UPD_RS485/ BOOT_UPD_USB</i> ).....	21
6.1.4	Альтернативная конфигурация ( <i>BOOT_CFG</i> ) .....	21
6.1.5	Установка активного порта ( <i>BOOT_SET_RS485/ BOOT_SET_USB</i> ) .....	23
6.1.6	Включение режима "Маяк" ( <i>BOOT_BEACON_OFF/ BOOT_BEACON_ON</i> ) .....	23
6.2	Режим "КОМАНДНЫЙ" .....	24
6.2.1	Пароль на командный режим .....	25
6.3	Режимы передачи данных.....	26
6.3.1	Режим передачи данных «Прозрачный» .....	26
6.3.2	Особенности режима передачи данных «Прозрачный» .....	26
6.4	Ретрансляция пакетов.....	27
6.5	Режимы работы по эфиру (гарантии доставки данных адресату) .....	28
6.5.1	Адресация .....	28
6.5.2	Широковещательный режим .....	28
6.5.3	Индивидуальный режим («точка-точка»).....	29
6.5.4	Особенности применения режима «точка-точка» .....	29
6.6	Буферизация принятых из эфира данных. Конкатенация данных.....	30
<b>7</b>	<b>ФОРМАТ ПАКЕТА В ЭФИРЕ</b> .....	<b>32</b>
7.1	Помехоустойчивое кодирование .....	32
7.2	Перемежение .....	33
7.3	Мультипликация данных .....	34
<b>8</b>	<b>РАСШИРЕННЫЕ ВОЗМОЖНОСТИ ПРИЁМА И РЕТРАНСЛЯЦИИ</b> .....	<b>35</b>
8.1	Приём ретранслированных пакетов .....	37
8.2	Игнорирование пакетов.....	38
<b>9</b>	<b>ФИЛЬТРАЦИЯ ДАННЫХ</b> .....	<b>40</b>

9.1	Фильтрация исходящих данных .....	41
9.2	Фильтрация входящих данных .....	41
9.3	Фильтрация данных при ретрансляции .....	41
<b>10</b>	<b>ПРИМЕРЫ ОРГАНИЗАЦИИ СЕТЕЙ.....</b>	<b>42</b>
10.1	Режим «ТОЧКА–ТОЧКА» .....	42
10.1.1	<i>Режим точка-точка.....</i>	<i>42</i>
10.1.2	<i>Режим точка-точка с повторителем.....</i>	<i>42</i>
10.2	Режим «ШИРОКОВЕЩАТЕЛЬНЫЙ» .....	43
10.2.1	<i>Широковещательная сеть без гарантии доставки сообщений.....</i>	<i>43</i>
10.2.2	<i>Широковещательная сеть с гарантией доставки сообщений.....</i>	<i>43</i>
10.2.3	<i>Широковещательная сеть с повторителем.....</i>	<i>43</i>
<b>11</b>	<b>ШИФРОВАНИЕ ДАННЫХ.....</b>	<b>44</b>
11.1	Шифрование AES.....	44
11.2	Шифрование XOR.....	44
<b>12</b>	<b>МОДУЛЬ ВВОДА ВЫВОДА (IO).....</b>	<b>45</b>
12.1	Общие сведения .....	45
12.2	Входы IN1...IN4.....	45
12.3	Выходы OUT1...OUT4 .....	46
12.4	Тест входов/выходов .....	46
12.5	Клемма 1WIRE .....	46
12.6	Адресация модулей .....	47
12.7	Режим IO_Slave.....	47
12.8	Режим IO_Master.....	48
12.9	Режим IO_Master. Мультиплексирование входов на выходы.....	49
12.10	Режим IO_Trans.....	50
12.11	Безопасное состояние выходов .....	51
12.11.1	<i>Индикаторы и реле ALARM.....</i>	<i>51</i>
12.11.2	<i>Безопасное состояние в режиме IO_Slave .....</i>	<i>53</i>
12.11.3	<i>Безопасное состояние в режиме IO_Master .....</i>	<i>53</i>
12.12	Функции входов .....	54
12.12.1	<i>"Дребезг" контакта.....</i>	<i>54</i>
12.12.2	<i>Детектор частоты.....</i>	<i>54</i>
12.12.3	<i>Защелка.....</i>	<i>54</i>
12.13	Функции выходов .....	56
12.13.1	<i>Выходной импульс.....</i>	<i>56</i>
12.14	Опрос по RS485 .....	56
12.15	Фильтр для ретрансляции \$IORPT .....	57
12.16	Типовые схемы включения.....	58
12.16.1	<i>Трансляция 4-х сухих контактов в обе стороны (точка-точка).....</i>	<i>58</i>
12.16.2	<i>Трансляция 4-х входов на 4 удаленных выхода. Трансляция 4-х удаленных входов на выходы.....</i>	<i>58</i>
12.16.3	<i>Трансляция 4-х входов на выходы произвольного количества модулей IO_SLAVE.....</i>	<i>59</i>
12.16.4	<i>Трансляция на выходы одного модуля входов нескольких удаленных модулей.....</i>	<i>60</i>
12.16.5	<i>Управление модулями IO_SLAVE внешним "мастером" .....</i>	<i>60</i>
<b>13</b>	<b>УДАЛЕННОЕ КОНФИГУРИРОВАНИЕ .....</b>	<b>61</b>
13.1	Удалённое конфигурирование в командном режиме.....	61
13.2	Удалённое конфигурирование по modbus .....	62
<b>14</b>	<b>ТЕСТИРОВАНИЕ СВЯЗИ.....</b>	<b>63</b>
14.1	Тестирования связи в командном режиме .....	63
14.2	Тестирования связи в прозрачном режиме .....	63
14.3	Режим Маяк.....	64
<b>15</b>	<b>ТЕХНОЛОГИЧЕСКИЕ ПАРАМЕТРЫ. РЕГИСТРЫ RG .....</b>	<b>65</b>
<b>16</b>	<b>КОМАНДЫ УПРАВЛЕНИЯ МОДУЛЕМ.....</b>	<b>66</b>
16.1	\$DMP(R) — вывод профиля (удалённого) модуля.....	66
16.2	\$IEE— сброс параметров по умолчанию.....	66
16.3	\$CH — изменение рабочего частотного канала.....	67
16.4	\$PWR — установка мощности передатчика .....	67
16.5	\$AR — скорость передачи данных по эфиру.....	67
16.6	\$MYID — изменение собственного адреса модуля .....	67
16.7	\$TXID — изменение адреса вызываемого модуля .....	67

16.8	\$FEC — тип FEC для информационных пакетов в эфире .....	67
16.9	\$COM — параметры последовательного порта .....	68
16.10	\$EXCHP — смена порта .....	68
16.11	\$MDA — режим работы модуля.....	68
16.12	\$ACKT — время ожидания подтверждения в режиме «точка-точка» .....	68
16.13	\$DAT — тип протокола входящих данных.....	68
16.14	\$MAXP — максимальное число пакетов в буфере .....	68
16.15	\$EOT — тайм-аут приема пакета в буфер на передачу .....	69
16.16	\$PLEN — нарезка входящих данных.....	69
16.17	\$EOC — символ передачи данных.....	69
16.18	\$RESPT — время задержки отправки подтверждения в режиме «точка-точка» .....	69
16.19	\$RETRY — число ретрансляций пакетов, требующих подтверждения .....	69
16.20	\$RPT — номер повторителя модуля .....	69
16.21	\$RID — адрес пакета разрешенного для ретрансляции .....	69
16.22	\$PID — адрес пакета для расширенного приёма .....	70
16.23	\$LID / \$XID — вывод / удаление адресов для ретрансляции и расширенного приёма ..	70
16.24	\$xFL — способ фильтрации для данных и ретрансляции.....	71
16.25	\$xFLDMP — вывод значений фильтров .....	71
16.26	\$xFL+ — включение диапазона значений в фильтр .....	71
16.27	\$xFL- — исключение диапазона значений из фильтра .....	71
16.28	\$FLS — запись значений фильтров в энозу.....	71
16.29	\$BCN — максимальное число одинаковых широковещательных пакетов.....	71
16.30	\$BCT — время между передачами широковещательных пакетов .....	71
16.31	\$RPRF — чтение основного профиля удалённого модуля .....	72
16.32	\$RPIO — чтение профиля IO удалённого модуля .....	72
16.33	\$WPRF — запись основного профиля удалённого модуля.....	72
16.34	\$WPIO — запись профиля IO удалённого модуля.....	72
16.35	\$EPS — начать редактирование удалённого профиля .....	72
16.36	\$EPE — остановить редактирование профиля удалённого модуля.....	72
16.37	\$UPD - смена программного обеспечения .....	72
16.38	\$RCHK — опрос удалённого модуля.....	72
16.39	\$RFIND — поиск устройств в эфире.....	72
16.40	\$TEST — запрос тестовых пакетов .....	73
16.41	\$DRG — вывод всех технологических параметров .....	73
16.42	\$RG — запись технологических параметров модуля .....	73
16.43	\$IRG — инициализация технологических параметров по умолчанию.....	73
16.44	\$R — сброс локального модуля.....	73
16.45	\$RST— сброс удалённого модуля.....	73
16.46	\$S — запись внутренних переменных модуля в ЭНОЗУ.....	74
16.47	\$E — выход из командного режима работы.....	74
16.48	\$LOG — вывод журнала событий в эфире модуля .....	74
16.49	\$SCAN — сканирование эфира .....	74
16.50	\$KEA/KEH — установка пароля AES шифрования.....	74
16.51	\$SK — установка пароля XOR шифрования.....	74
16.52	\$NAM — задание имени модулю .....	75
16.53	\$CFL— установка/снятие пароля на командный режим.....	75
16.54	\$CFU— ввод пароля на командный режим.....	75
16.55	\$LOCK — регистр блокировок.....	75
16.56	\$DMPLOCK — вывод регистра блокировок .....	75
<b>17</b>	<b>КОМАНДЫ УПРАВЛЕНИЯ МОДУЛЕМ ВВОДА ВЫВОДА (IO) .....</b>	<b>76</b>
17.1	\$IOIEE – инициализация модуля IO в начальное состояние .....	76
17.2	\$IORIN – сброс значений входов .....	76
17.3	\$IOROUT – сброс значений выходов.....	76
17.4	\$IOMASTER – ввод модуля IO в режим IO_MASTER .....	76
17.5	\$IOSLAVE – ввод модуля IO в режим IO_SLAVE .....	77
17.6	\$IOTRANS– ввод модуля IO в режим IO_TRANS.....	77
17.7	\$IOADR – базовый адрес модуля .....	77
17.8	\$IONDX – индекс модуля .....	77

17.9	\$IORPT – фильтр для ретрансляции .....	77
17.10	\$IOF02 – список модулей IO_Slave, опрашиваемых функцией F02 .....	77
17.11	\$IOF23 – подмена команды F02 на F23 .....	77
17.12	\$IOF15 – список IO_Slave модулей, опрашиваемых функцией F15 .....	77
17.13	\$IOF15BC – широкопередаточный режим функции F15 .....	77
17.14	\$IOQPT – способ опроса IO_Slave устройств .....	77
17.15	\$IOPT – пауза между опросами .....	78
17.16	\$IOSM – маскирование выходов .....	78
17.17	\$IOSFT – таймаут пропадания связи .....	78
17.18	\$IOSFV – конфигурация выходов в безопасном состоянии .....	78
17.19	\$IOALR – режим безопасного состояния выходов .....	79
17.20	\$IOMXO – мультиплексор входов в режиме IO_Master .....	79
17.21	\$IODBN – счетчик дрейфа входов .....	79
17.22	\$IOINF – конфигурация детектора частоты .....	80
17.23	\$IOFR0/1 – значение компаратора детектора частоты .....	80
17.24	\$IOINL – защелка входов .....	80
17.25	\$IOUTM – генерация выходного импульса .....	80
17.26	\$IOUTxT – длительность выходного импульса .....	80
17.27	\$IOINT – передача в эфир состояний входов при изменении состояния одного из .....	80
17.28	\$IOMOD – режим работы модуля ввода/вывода .....	80
<b>18</b>	<b>КАРТА РЕГИСТРОВ MODBUS .....</b>	<b>81</b>
18.1	Чтение входов .....	81
18.2	Установка выходов .....	81
18.3	Чтение профиля .....	82
18.4	Запись профиля .....	82
18.5	Чтение состояния модуля .....	82
18.6	Запись состояния модуля .....	83
<b>19</b>	<b>ОТЛАДОЧНЫЙ ПОРТ .....</b>	<b>84</b>
<b>20</b>	<b>ВОПРОСЫ И ОТВЕТЫ .....</b>	<b>85</b>
<b>21</b>	<b>ГАБАРИТНЫЕ РАЗМЕРЫ .....</b>	<b>86</b>
<b>22</b>	<b>ОСНОВНЫЕ ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ .....</b>	<b>87</b>

# 1 ИСТОРИЯ ВЕРСИЙ

**Версия Руководства: 1.02**

С этой версии начинается история.

## 2 ОБЩАЯ ИНФОРМАЦИЯ

### 2.1 НАЗНАЧЕНИЕ И ОБЩИЕ СВЕДЕНИЯ

Радиомодуль «СПЕКТР 868 IO Lora» БАКП.464426.004 (далее по тексту — модуль) представляет собой устройство для приёма/передачи данных по радиоканалу со скоростью 980,1760,2700,3125,4500,5470 и 9300 бод в диапазоне частот 868 МГц при выходной мощности 25 мВт, что позволяет использовать его без получения разрешений органов ГосСвязьНадзора.

Модуль предназначен для использования в системах удаленного управления и/или сбора данных в качестве оконечного устройства, к входам которого подключаются датчики типа "сухой контакт", а к выходам — исполнительные устройства.

В режиме Slave модуль является пассивным устройством, он выполняет полученные по радио команды в протоколе Modbus на считывание состояния своих входов и на управление своими выходами.

В режиме Master модуль сам формирует и отправляет по радио команды в протоколе Modbus, управляя тем самым удаленными модулями, работающими в режиме Slave. В таком режиме обеспечивается трансляция состояния входов одного модуля (Master) на выходы удаленных модулей (Slave) и наоборот.

Модуль способен работать в различных режимах с развитой системой адресации, позволяя пользователю максимально гибко использовать его при построении различных конфигураций сетей беспроводной передачи данных: точка–точка, точка–много точек.



Модуль является технически сложным электронным устройством. Конфигурация, установка и эксплуатация модуля должна производиться персоналом с достаточной квалификацией.

### 2.2 ОСНОВНЫЕ ПРИНЦИПЫ РАБОТЫ МОДУЛЯ

Модуль осуществляет все необходимые функции для обмена данными по радиоканалу:

- управление встроенным приёмопередатчиком диапазона 868 МГц (установка частоты, мощности, скорости данных в эфире и др.);
- получение и буферизацию данных по интерфейсам RS-485 или USB;
- формирование пакетов данных для отправки в эфир;
- введение адресации, помехоустойчивого кодирования и перемежения;
- приём и демодуляцию входного радиосигнала и декодирование принятых пакетов;
- буферизация принятых данных и передача их по интерфейсам RS-485 или USB;
- хранение данных о конфигурации модуля и его технологических параметрах в энергонезависимом ОЗУ (ЭНОЗУ) и изменение этих установок в командном режиме.

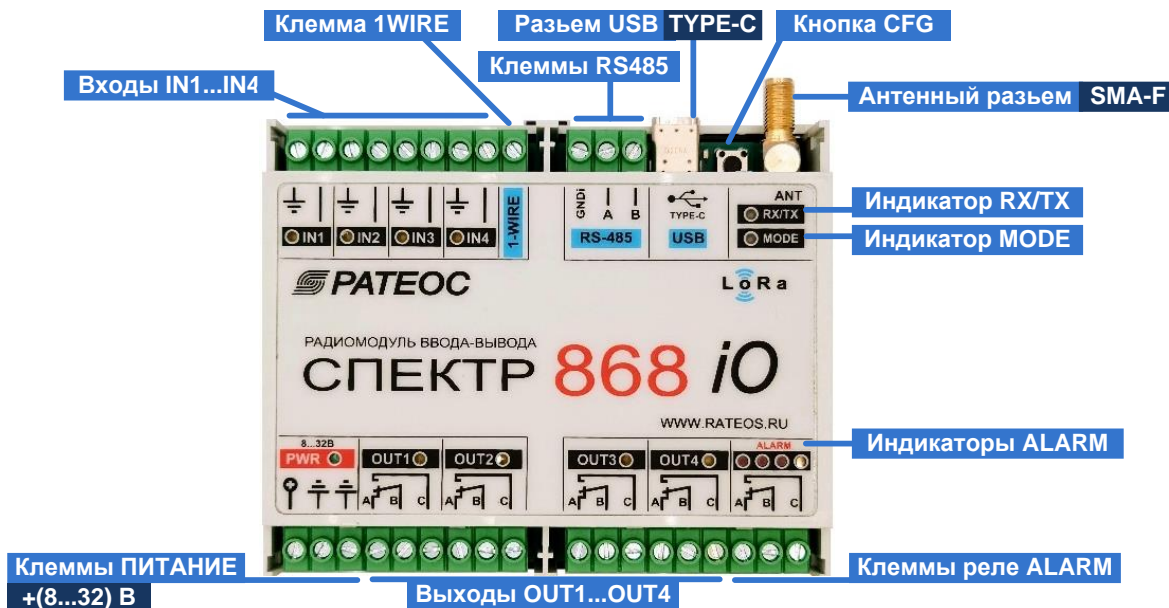
### 2.3 СОПУТСТВУЮЩЕЕ ОБОРУДОВАНИЕ И МАТЕРИАЛЫ

Также можно заказать дополнительное оборудование:

- антенные кабельные сборки необходимой длины и с нужными разъёмами на концах для подключения антенн к модулю (некоторые антенны поставляются уже с кабелем);
- устройства защиты от грозовых разрядов (грозоразрядники);
- УТР кабель (две витые пары в общей оболочке) нужной длины и подходящего под кабельный ввод диаметра для подключения к модулю питания и RS-485;
- источники питания для модулей;
- материалы для герметизации разъёмных соединений;



## 3 РАЗЪЁМЫ, ОРГАНЫ УПРАВЛЕНИЯ И ИНДИКАЦИИ



### 3.1 ИНДИКАТОРЫ

#### 3.1.1 ИНДИКАТОР MODE

В нормальном режиме (режим передачи данных), двухцветный индикатор MODE показывает следующие состояния:



**Не горит.** Входной буфер данных пуст. Модулю нечего передавать в эфир (предыдущие данные переданы)



**Горит зелёным** Во входном буфере модуля есть данные для передачи.



**Мигает красным** 30 70 30 70 ... (время в мс)

Ошибка при приёме данных по последовательному порту, вызванная **несовпадением** параметров портов модуля и внешнего оборудования или **перепутанными** местами линиями A и B интерфейса RS-485

**Мигает красным** 300 200 300 200 ... (время в мс)

Входной буфер данных модуля переполнен (поток данных на последовательный порт модуля слишком велик, модуль не успевает передавать данные в эфир, из-за этого данные могут быть потеряны) — такая ситуация может быть следствием того, что пропускная способность модуля по эфиру ниже, чем плотность данных, подаваемых в модуль



В режиме передачи данных, индикатор MODE **не должен** мигать красным. Если такое происходит, то проблема исключительно в "стыке" между модулем и внешним оборудованием (радиочасть тут не причём).



**Горит красным** «Командный» режим

При включении питания:






**2 раза мигает зелёным** Выбран активный порт (для передачи данных) USB



**3 раза мигает зелёным** Модуль находится в режиме "МАЯК"

### 3.1.2 ИНДИКАТОР RX/TX

В режиме передачи данных, индикатор RX/TX («ПРИЁМ/ПЕРЕДАЧА») отображает состояние приёмопередатчика модуля:

 RX/TX	Передача данных в эфир (Красный)
 RX/TX	Прием данных из эфира (Зеленый)
 RX/TX	Нет активности в эфире (не горит)

В остальных случаях используется как вспомогательный индикатор режимов.

## 3.2 РАЗЪЁМЫ, КЛЕММЫ, КНОПКА

### Антенный разъём

Высокочастотный коаксиальный разъём типа SMA-F с волновым сопротивлением 50 Ом для подключения внешних антенн.

### Клеммы и индикатор ПИТАНИЕ

Клеммы для подачи питания на модуль. Индикатор загорается при наличии питания.

### Клеммы RS485

На клеммы RS-485 выведены цепи А и В последовательного интерфейса RS-485. Также доступен сигнал GNDi (изолированная "земля").

### Разъём USB

Разъём для подключения к USB шине ПК для конфигурации и отладки. Тип разъема TYPE-C. Для того чтобы ОС Windows поддерживала модуль, необходимо установить драйверы шины USB, которые можно скачать с сайта [www.rateos.ru](http://www.rateos.ru)

Рядом с разъёмом находится светодиодный индикатор, который загорается всегда при подключении к ПК в не зависимости установлены драйверы или нет.

Следует помнить, что из-за ограничений на длину USB кабеля (не более 3...4 метров) модуль нельзя устанавливать далеко от компьютера.

### Клемма 1WIRE

К клемме подключаются устройства с интерфейсом 1WIRE (см. раздел "[Клемма 1WIRE](#)").

### Клеммы и индикаторы IN1...IN4

К четырем парам клемм IN1...IN4 подключаются внешние дискретные датчики типа «сухой контакт» (см. раздел "[Входы IN1...IN4](#)")

### Клеммы и индикаторы OUT1...OUT4

К четырем группам клемм OUT1...OUT4 подключаются внешние исполнительные устройства (см. раздел "[Выходы OUT1...OUT4](#)")

### Клеммы и индикатор ALARM

На клеммы выведены сухие контакты реле ALARM, которое используется для сигнализации о пропадании связи и представляет собой переключающее электромагнитное реле, позволяющее коммутировать нагрузку с максимально допустимым током 4 А при напряжении 250 В 50 Гц или 4 А при постоянном напряжении 24В.

При пропадании связи реле АВАРИЯ срабатывает (замыкаются клеммы В и С и размыкаются клеммы В и А).

Индикаторы ALARM используются для индикации аварии в зависимости от режима работы модуля ввода-вывода.

При включении питания на индикаторах ALARM кратковременно отображается режим работы модуля IO MASTER или IO SLAVE (см. раздел "[Индикаторы и реле ALARM](#)")

### Кнопка CFG

Кнопка CFG используется для переключения режимов работы модуля.

### 3.2.1 СБРОС МОДУЛЯ. ИНДИКАЦИЯ ВЕРСИИ ПО

С помощью кнопки CFG можно осуществить сброс модуля не переключая питания. Это может быть удобно, например, для перехода в режим загрузчика в уже смонтированном модуле.

Для входа в режим сброса необходимо максимум в течении 3сек 2 раза войти в командный режим и выйти из него нажимая на кнопку CFG, а на 3-й раз нажать и удерживать ее более 3 сек.



После этого, индикатор MODE начнет мигать в течении 3с. Если до истечения 3с отпустить кнопку, то модуль просто пересбросится. По истечении 3с модуль также пересбросится, но кнопка при этом будет нажата, что будет эмуляцией входа в режим загрузчика (как при включении питания с нажатой кнопкой CFG).

В момент мигания индикатора MODE, на индикаторах ALARM отображается версия встроенного ПО в двоичном коде. Версия 1.00 - ни один индикатор не горит, версия 1.01 - горит самый правый индикатор и т.д.

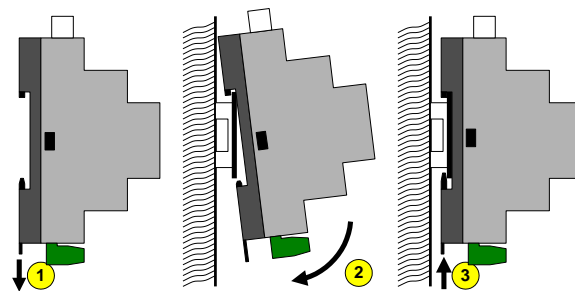
## 4 УСТАНОВКА И ПОДКЛЮЧЕНИЕ ВНЕШНЕГО ОБОРУДОВАНИЯ

### 4.1 УСТАНОВКА МОДУЛЯ

Для установки модуля следует:

- отвести вниз фиксирующую защелку;
- установить модуль на рейку так, чтобы выступы в верхней части корпуса попали за край рейки;
- вернуть фиксирующую защелку вверх.

Модули не имеют защиты от воздействий окружающей среды, поэтому при необходимости должны устанавливаться в шкаф, обеспечивающий нужную степень защиты.



Перед тем как устанавливать модули на объекты, рекомендуем сначала проверить наличие радиосвязи между ними и необходимую логику работы в лабораторных условиях. Такая проверка «на столе» позволит обнаружить и оперативно устранить возможные проблемы.

### 4.2 ПОДКЛЮЧЕНИЕ АНТЕНН

Совместно с модулем применяются различные антенны диапазона 868 МГц с волновым сопротивлением 50 Ом, отличающиеся направленностью, усилением, конструкцией и т. д.

В комплект поставки антенны не входят, заказывайте нужные антенны отдельно.

Выбор антенн определяется взаимным расположением объектов в системе, расстояниями между ними, условиями местности и т. д. Широкий ассортимент антенн и рекомендации по их выбору доступны на сайте [www.rateos.ru](http://www.rateos.ru).

Дальность связи между модулями зависит от различных факторов, основными из которых являются (перечислены по уменьшению важности):

- характер местности;
- параметры и место установки используемых антенн;
- помеховая обстановка в радиоэфире;
- параметры модуля (мощность, скорость данных в эфире).

На практике в максимально комфортных условиях (прямая видимость, направленные антенны с усилением 10...12 dBi, отсутствие помех) можно рассчитывать на дальность до 10...15 км.

Для подключения антенны в модулях используется разъём SMA-F.

Для соединения модуля с антенной следует применять кабель с волновым сопротивлением 50 Ом, например RG-58. При этом не рекомендуется использовать слишком длинный (более 10...15 м) кабель, иначе он внесет существенные потери в высокочастотный сигнал, что приведет к уменьшению дальности связи. Если условия эксплуатации требуют применения более длинного кабеля, используйте кабель с низкими потерями, например RG-213 или RG-8.

Допускается работа модуля без антенны или с отрезком провода вместо антенны (например, при лабораторных экспериментах в пределах комнаты, когда чувствительности модулей достаточно и для работы без антенны).

#### 4.2.1 ГЕРМЕТИЗАЦИЯ РАЗЪЁМНЫХ СОЕДИНЕНИЙ

Модуль в исполнении IP65 рассчитан на установку в незащищённом от осадков месте, при этом антенна подключается к нему через разъёмное соединение вне зоны

герметизации. В таких условиях следует обеспечить дополнительную герметизацию разъёмного антенного соединения.

Герметизировать следует винтовое соединение антенного разъёма модуля с гайкой ответного разъёма SMA-M на антенном кабеле, а также место обжима этих разъёмов на кабель.

Для герметизации используйте специальную термоусадочную трубку (термофит) с клеевым наполнителем. Отрежьте трубку такой длины, чтобы она покрыла все разъёмное соединение «с запасом» в 2...3 см.

Термоусадочная трубка подходит для герметизации разъёмов и кабелей с небольшим отличием в диаметре, как в случае кабеля RG-58 и разъёма SMA у модуля. Если же необходимо герметизировать кабель с разъёмом, чьи диаметры сильно отличаются (например, тот же кабель RG-58 и разъём TNC и N-типа, часто используемый на антеннах) термоусадка уже не подойдет.

Для таких случаев рекомендуем использовать электроизоляционную мастику («сырую резину») или самовулканизирующуюся клейкую ленту на основе этиленпропиленовой резины (ЭПР).



Для герметизации следует плотно, с натяжением при намотке и перекрытием намотать электроизоляционную мастику или самовулканизирующуюся клейкую ленту на разъёмное соединение (начиная с антенного кабеля и до самого корпуса модуля). Намотку осуществить сначала в одном, затем, не прерываясь, в другом направлении.

Поверх слоя мастики или самовулканизирующейся ленты рекомендуем наложить слой ПВХ изоленты. Используйте качественную изоленту, рассчитанную на широкий диапазон температур, иначе она быстро потеряет свои свойства.



Не используйте для герметизации «обычную» термоусадку (термомфит) без клеевого слоя — она не обеспечит требуемой герметичности. Такую термоусадку можно применить только вместо ПВХ изоленты поверх слоя мастики или самовулканизирующейся ленты.



Не используйте для герметизации обычную изоленту — со временем под действием окружающей среды клеевой слой теряет свои свойства и слои изоленты начинают отслаиваться.



Не используйте для герметизации ацетатные герметики — они «разъедают» цветные металлы и разъём не прослужат и нескольких месяцев.

### 4.3 ПИТАНИЕ МОДУЛЯ

Модуль питается от внешнего источника постоянного напряжения в диапазоне от 8 В до 32 В и мощностью не менее 2 Вт. Обычно используют промышленные источники питания на 12 В или 24 В.

Если для подачи питания используется длинный (более 50 метров) кабель, рекомендуется использовать источник питания на 24 В, чтобы падение напряжения в кабеле не привело к снижению напряжения на клеммах модуля ниже 8 В.

В режиме «Приём» модуль потребляет около 0,5 Вт (40 мА при напряжении питания 12 В и 20 мА при напряжении питания 24 В).

Потребление в режиме «Передача» зависит от установленной выходной мощности передатчика, как показано в таблице ниже.

Программная установка мощности	Мощность передатчика	Потребление в режиме «Передача»		
		Мощность	Ток при питании +12 В	Ток при питании +24 В
\$PWR=0	25 мВт	1 Вт	80 мА	50 мА
\$PWR=1	100 мВт	1,3 Вт	110 мА	55 мА
\$PWR=2	250 мВт	1,7 Вт	140 мА	70 мА
\$PWR=3	350 мВт	2,2 Вт	170 мА	90 мА

Небольшое потребление и широкий диапазон напряжений модуля позволяют использовать для питания модуля уже имеющиеся в системе источники, питающие другое оборудование — не обязательно приобретать для модулей отдельные источники питания.

Модуль начинает работать сразу после подачи питания.

#### 4.4 ИНТЕРФЕЙСЫ RS-485, USB

В модуле два независимых равно функциональных порта - RS485 и USB. В процессе работы один порт используется для передачи пользовательских данных (активный порт), на другой выводится отладочная информация. По умолчанию, RS485 порт предназначен для передачи данных, а на USB выводится отладочная информация и по нему осуществляется конфигурация в командном режиме.

Параметры порта для передачи данных (скорость, чётность/нечётность, количество стоповых бит и др.) задаются командой \$COM в командном режиме. Значение по умолчанию: **9600 8N1**.

Отладочный порт **ВСЕГДА** настроен на 57600 8N1 и изменить эти параметры нельзя.



- Выбор интерфейса для передачи данных производится с помощью команд загрузчика BOOT\_SET\_485/ BOOT\_SET\_USB.
- Конфигурация возможна по любому порту.

##### 4.4.1 ИНТЕРФЕЙС RS-485

При использовании интерфейса RS-485 подключайте внешние устройства к контактам А и В модуля. Для подключения рекомендуется использовать витую пару. Длина кабеля может составлять сотни метров. Интерфейс RS-485 работает в полудуплексном режиме. В модуле интерфейс имеет **гальваническую развязку**.

Терминальный резистор в модуле не предусмотрен. В случае необходимости, можно использовать внешний резистор подключив его к контактам А и В.

За функционирование интерфейса кроме стандартных настроек скорости, четности и т.д., также отвечают 2 технологических параметра - 485\_PRE\_TMR и 485\_PST\_TMR.

После активации шины (внутренним сигналом трансивера RS485) модуль выжидает время 485\_PRE\_TMR после чего, начинает передавать данные в шину. По окончании передачи последнего байта, модуль выжидает время 485\_PST\_TMR и после этого деактивирует шину. Оба параметра задаются кратно времени передачи 1 бита на текущей

скорости порта и программируются регистрами \$RG00 и \$RG01 (см. раздел "[Технологические параметры. Регистры RG](#)")



По умолчанию, параметры 485\_PRE\_TMR и 485\_PST\_TMR настроены для обеспечения нормальной работы шины RS485 и менять или задавать их **нет необходимости**.

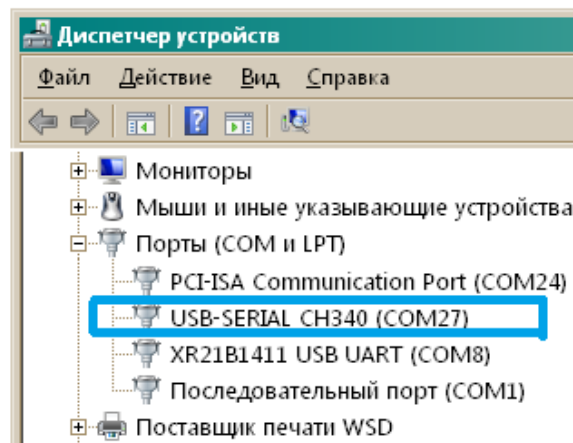
#### 4.4.2 ИНТЕРФЕЙС USB. УСТАНОВКА ДРАЙВЕРА ШИНЫ USB НА ПК

Для установки драйвера необходимо запустить программу CH340SER.EXE и следуя инструкциям установить драйвер. Далее подключить модуль с помощью кабеля USB TYPE-C.

В Диспетчере устройств Windows должен появиться виртуальный COM-порт под названием "**USB-SERIAL CH340 (COMxx)**", где xx - номер виртуального порта. Его нужно открывать в терминальной программе при конфигурировании модуля



К шине USB компьютера одновременно можно подключать только один модуль, иначе возникнет конфликт оборудования.



#### 4.4.3 ИНДИКАЦИЯ ОШИБОК ПРИ РАБОТЕ ПО ПОСЛЕДОВАТЕЛЬНЫМ ПОРТАМ

В модуле предусмотрена индикация ошибок при работе с внешним оборудованием по интерфейсам в режиме передачи данных ( см. раздел "[Индикатор MODE](#)").

## 5 НАСТРОЙКА МОДУЛЕЙ

### 5.1 КОНФИГУРАЦИЯ ПАРАМЕТРОВ

Модуль имеет несколько режимов работы, позволяющих использовать его при построении систем передачи данных с разнообразным внешним оборудованием, поэтому перед эксплуатацией может потребоваться установка параметров, определяющих:

- режим адресации (широковещательный, групповой или «точка-точка»);
- параметры встроенного приёмопередатчика (рабочая частота, мощность и т.д.);
- скорость данных в эфире и на последовательных интерфейсах;
- другие параметры, определяющие работу модуля.

Параметры хранятся в энергонезависимой памяти (ЭНОЗУ) модуля и восстанавливаются при включении питания. Таким образом сконфигурированный ранее модуль готов к работе в заданном режиме при подаче на него питания.

По умолчанию, модуль готов к работе и в большинстве случаев **не требует начальной конфигурации**. Заводские (по умолчанию) настройки следующие:



- широковещательная адресация (\$TXID=FFF)
- активный порт RS485. Скорость 9600,8N1
- Частотный канал 868,765 КГц (\$CH=00)
- Скорость в эфире 3125 бит/сек (\$AR=3). Минимальное время ожидания ответа для внешнего оборудования при данной скорости должно быть 2 секунды.
- Мощность в эфире 25 мВт (\$PWR=0)
- конфигурация по USB (настройки порта 57600,8N1)

#### 5.1.1 ПОДКЛЮЧЕНИЕ К ПК ДЛЯ КОНФИГУРАЦИИ

Конфигурация модуля осуществляется по любому интерфейсу с помощью персонального компьютера (ПК). Для конфигурации можно использовать любую терминальную программу для ПК (например Pcomm). Установка параметров осуществляется подачей текстовых команд. Подробная информация описана в разделе ["Режим "КОМАНДНЫЙ"](#).

### 5.2 РАБОТА МОДУЛЕЙ В КАЧЕСТВЕ ПРОЗРАЧНОГО РАДИОУДЛИНИТЕЛЯ RS-485/USB

#### 5.2.1 ОСНОВНЫЕ ПРИНЦИПЫ

Прозрачность при передаче данных означает, что от внешнего оборудования не требуется обрамлять данные дополнительной информацией перед передачей в модуль. Модуль, получив данные по последовательному порту сам добавляет заголовок и проверочную информацию в соответствии со своим протоколом формируя при этом **пакет**, который передается в эфир. На приемной стороне модуль анализирует все данные из эфира на предмет пакетов своего протокола и выдает на порт только пользовательские данные, удаляя при этом всю лишнюю информацию.

Внешнее оборудование, работающее по интерфейсам RS-485 или RS-232, будет работать и через радиомодули.

Протоколы, по которым работает внешнее оборудование (ModBus, Бolid и т.д.) практически не имеют значения, поскольку модули будут прозрачны для этих протоколов. Все это касается как простых систем, когда нужно связать пару устройств с помощью двух модулей, так и более сложных систем сбора данных из множества удалённых объектов.

Стоит отметить, что имеются некоторые оговорки касательно прозрачности модулей, и в некоторых случаях после замены проводов RS-485 или RS-232 на радиомодули система откажется сразу же заработать. К счастью, эти возможные проблемы в работе оборудования без особого труда преодолимы (см. раздел ["Проверка модулей и стыковка с оборудованием перед установкой на объект"](#)).





При работе в прозрачном широковещательном режиме все модули системы будут иметь одинаковые настройки, а их роль в системе (мастер или слейв, ведущий или ведомый и т.д.) будет определяться только ролью внешнего оборудования, а не настройками самих модулей. Таким образом, все модули в таких системах равнозначны и взаимозаменяемы.

Если Вы не уверены, что модуль имеет заводские установки, рекомендуем сбросить его настройки в заводские командой \$IEE — это гарантирует известное исходное состояние модуля.

### 5.2.2 ПАРАМЕТРЫ АКТИВНОГО ПОРТА (\$COM)

Для корректного стыка с внешним оборудованием по активному порту (например, RS-485) необходимо, чтобы параметры порта модуля **совпадали** с параметрами порта внешнего оборудования. У модуля по умолчанию 9600-8N1.

Модуль показывает ошибки при приёме данных от внешнего оборудования по активному порту (см. раздел "[Индикатор MODE](#)") — это можно использовать для диагностики неправильной конфигурации и/или ошибок при подключении последовательных портов. Параметры активного порта устанавливаются командой \$COM.

### 5.2.3 СКОРОСТЬ ОБМЕНА ДАННЫМИ МЕЖДУ МОДУЛЯМИ В ЭФИРЕ (\$AR)

Скорость обмена данными в радиоэфире между модулями никак не связана со скоростью обмена данными между модулями и внешним оборудованием по последовательному порту: модули могут работать с оборудованием на одной скорости, а в эфире между собой — на любой другой. Для выбора скорости в эфире следует придерживаться следующих критериев:

- Если данные через модемы передаются по протоколам "запрос-ответ" (modbus, болид и т.д.), то временные характеристики задаются именно протоколами, а именно тайм-аутом ожидания ответа на запрос - чем больше тайм-аут, тем "дольше" может передаваться запрос-ответ. В этом случае, скорость в эфире нужно выбирать из принципа "ниже скорость - больше дальность" и, соответственно, отрегулировать тайм-аут ожидания ответа у "мастера" внешнего оборудования.  
Если скорость опроса не будет устраивать и есть "запас" по дальности, то можно попробовать увеличить скорость в эфире.
- Если нужно передавать данные преимущественно в одну сторону с максимальной пропускной способностью, то следует увеличивать скорость в эфире.

Если объем данных по последовательному порту будет выше способности модема передавать его в эфир с заданной скоростью или же тайм-аут ожидания ответа у внешнего оборудования "мастер" будет меньше, чем время прохождения "запрос-ответ" в эфире, то внутренний буфер модема постепенно заполнится и начнется потеря данных.

При этом следует понимать, что под скоростью обмена данными в эфире понимается «физическая» скорость, тогда как «информационная» (полезная) скорость при этом ниже физической, поскольку «полезные» данные от внешних устройств перед передачей в эфир дополняются служебными - заголовок, помехоустойчивое кодирование и т.д.

Скорость обмена данными в эфире устанавливается командой \$AR:

AR	Скорость в эфире, бит/сек	Ширина полосы, КГц
0	980	125
1	1760	125
2	2700	62.5
3	<b>3125 (по умолчанию)</b>	<b>125</b>
4	4500	62.5
5	5470	125
6	9300	125
7	22000	125

#### 5.2.4 РАБОЧИЙ КАНАЛ (\$CH)

Модуль позволяют менять рабочую частоту в пределах 4-х каналов с шагом 125 КГц. Все модули в одной системе должны работать на одной частоте. Канал меняется командой \$CH.

КАНАЛ	ЧАСТОТА, МГц
00 (по умолчанию)	868,765
01	868,890
02	869,015
03	869,140

#### 5.2.5 ПАРАМЕТРЫ ВХОДЯЩИХ ДАННЫХ RS485/USB ДЛЯ ПЕРЕДАЧИ В ЭФИР (\$DAT)

Модем имеет буфер на исходящие (пользовательские) данные размером 2048 байт. Состояние буфера индицирует светодиод MODE – загорается **зелёным** при наличии в буфере данных для отправки в эфир, при заполнении буфера - **мигает красным**.

В режиме "точка - точка", светодиод MODE гаснет только в том случае, если получено подтверждение от удаленного модуля на все данные в буфере. Если удаленный модуль выключен или недоступен по радио, светодиод MODE будет гореть постоянно, а новые входящие данные приведут к переполнению.

В режиме "широковещательный" светодиод MODE гаснет сразу же после передачи последнего пакета в эфир. К переполнению буфера может привести только непрерывный поток данных со скоростью значительно отличающийся от скорости в эфире.

Для обеспечения прозрачности передачи данных необходимо задать условия, при которых модемом формируются пакеты для передачи в эфир. Этими условиями будет обозначаться режим или протокол входящих данных. Возможны следующие режимы:

##### EOT (End Of Time)

##### \$EOT или \$PLEN

Признаком формирования модемом пакета для передачи в эфир является тайм-аут после приема последнего байта данных - как только тайм-аут превышает заданное значение, данные "защелкиваются" в буфере в пакет и становятся в очередь на передачу в эфир. Пришедшие следом данные будут буферизироваться и относиться уже к следующему пакету для передачи в эфир. Тайм-аут задается командой \$EOT.

В случае, когда данные могут представлять собой непрерывный поток информации без пауз или же размер единоразовых данных предназначенных для передачи превышает внутренний буфер модема, *дополнительным* признаком формирования пакета для передачи в эфир служит параметр \$PLEN, который задает размер в байтах  $x \cdot 8$  при превышении которого, уже принятые ранее в буфер данные без пауз "защелкиваются" для передачи в эфир. Таким образом, условиями для формирования пакета для передачи в эфир в данном режиме являются тайм-аут \$EOT по приему последнего байта **ИЛИ** размер принятых данных превысил значение \$PLEN. Параметр \$PLEN активизируется только при не нулевом значении.

##### EOC (End Of Char)

##### \$EOC

В режиме EOC признаком конца данных и, соответственно, сигналом формирования модемом пакета для передачи в эфир является специальный символ EOC. Символ задается командой \$EOC. Следует отметить, что непрерывный поток заканчивающийся символом EOC не должен превышать размер буфера модема. В этом случае данные не будут переданы в эфир по причине переполнения буфера и будут удалены.

##### RTU (Modbus RTU)

##### \$EOT

В режиме RTU признаком конца данных является тайм-аут \$EOT (аналогично режиму EOT). Одновременно с приемом, осуществляется подсчет контрольной суммы (CRC) данных по стандарту протокола MODBUS RTU. По истечении \$EOT данные "защелкиваются" в очередь для передачи в эфир. Перед началом передачи в эфир, пакет проверяется на корректность контрольной суммы - если она не совпала, то пакет удаляется и в эфир не передается. Данный режим удобно применять когда модем подключен к сети RS485 в которой есть трафик, часть которого не нужно передавать в эфир. В данном режиме осуществляется фильтрация данных на соответствие пакетов стандарту Modbus RTU по

контрольной сумме. Также, возможна фильтрация по адресам с помощью фильтров на передачу.

### DL8 (DALLAS8)

### \$EOT

Режим аналогичен логике работы режима RTU, только контрольная сумма данных подсчитывается по алгоритму DALLAS CRC8 (используется в протоколах охранно-пожарных систем).



Во всех режимах текущий пакет удаляется из буфера в следующих случаях:

- нет места для приема начала пакета (буфер переполнен);
- в процессе приема данных произошло переполнение буфера;
- произошла ошибка приема байта по порту (четность, нарушение формата и т.д.).

**Удаление пакета будет выполнено после получения признака его конца.**

Например, если во время приема ASCII строки "11122243337" на символе '4' произошло переполнение буфера или символ принят с ошибкой четности или обнаружена ошибка кадра (асинхронного последовательного протокола RS232/485) то, после приема последнего символа '7' и по истечении тайм-аута EOT (например, в режиме протокола EOT) из буфера будет удалена вся строка "11122243337".



Переполнение входящего буфера в большинстве случаев является признаком неправильно настроенной логики работы внешнего оборудования. Как правило, это неверно выставленный тайм-аут ожидания ответа в протоколе внешнего оборудования - запросы шлются намного чаще, чем модуль способен доставить ответ от удаленного оборудования.

В случае, когда тайм-аут внешнего протокола "настроен" для работы по проводной линии (минимальный) и его **невозможно** изменить, для предотвращения заполнения буфера постоянными запросами предусмотрен параметр \$MAXP, который задает максимальное число пакетов которые могут одновременно находиться в буфере. Например, если параметр задан равным 3 и в буфере уже находятся 3 пакета, то новый пришедший 4-ый пакет в буфер не записывается. Стоит заметить, что задание параметра \$MAXP не гарантирует работоспособность системы (где нет регулировки тайм-аута) по эфиру. По умолчанию, параметр \$MAXP отключен (равен 0)

## 5.3 ПРОВЕРКА МОДУЛЕЙ И СТЫКОВКА С ОБОРУДОВАНИЕМ ПЕРЕД УСТАНОВКОЙ НА ОБЪЕКТ

После конфигурации всех модулей системы и перед тем как устанавливать их на объекты, рекомендуем сначала проверить наличие радиосвязи между модулями и добиться успешной стыковки их с оборудованием в *лабораторных условиях*. Такая проверка «на столе» позволит обнаружить возможные проблемы и оперативно устранить их.

Первым делом следует убедиться в работоспособности внешнего оборудования без модулей (по кабелю): настройте и запрограммируйте систему так, чтобы она работала в нужном режиме по проводам.

После этого следует выполнить проверку связи между модулями без внешнего оборудования чтобы убедиться, что модули между собой нормально работают по радио (см. раздел ["ТЕСТИРОВАНИЕ СВЯЗИ"](#)).

Только после этих проверок можно приступать к замене проводов радиомодулями, иначе в случае проблем будет трудно разобраться, в чем дело и что не так.

Модуль показывает ошибки при приеме данных от внешнего оборудования (см. раздел [«Индикатор MODE»](#)) — это можно использовать для диагностики неправильной конфигурации и / или ошибок при подключении RS-485.

Если подключенное оборудование "сходу" не заработало через радиомодули, необходимо ознакомиться с разделами:

["Режим передачи данных «Прозрачный»"](#);

["Особенности режима передачи данных «Прозрачный»"](#);

["Вопросы и ответы"](#).

## 6 РЕЖИМЫ РАБОТЫ МОДУЛЯ

Модуль может находиться в одном из следующих режимов:

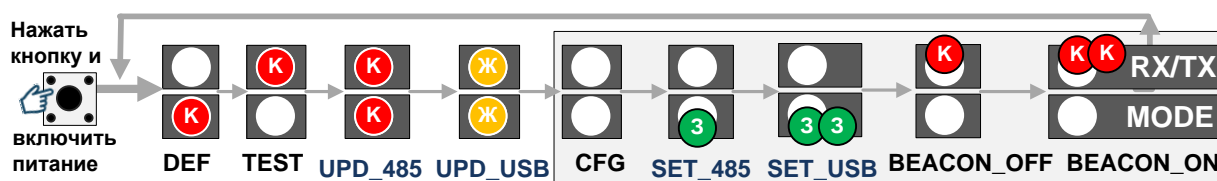
- **Режим загрузчика** – дополнительные режимы (команды) конфигурации при включении питания;
- **Командный** – режим конфигурации с помощью компьютера;
- **Рабочий** – "штатный", режим работы;

### 6.1 РЕЖИМ ЗАГРУЗЧИКА

В этом режиме имеется возможность выполнить операции (команды) с модулем с минимальным участием компьютера, либо без него. Определены следующие команды:

<b>BOOT_DEF</b>	– вход в командный режим по USB порту на скорости 9600 8N1;
<b>BOOT_TEST</b>	– специальный режим для проверки модуля;
<b>BOOT_UPD_485</b>	– обновление ПО по интерфейсу RS485 на скорости 57600 8N1;
<b>BOOT_UPD_USB</b>	– обновление ПО по интерфейсу USB на скорости 57600 8N1;
<b>BOOT_CFG</b>	– минимальная конфигурация модуля без компьютера;
<b>BOOT_SET_485</b>	– установка порта для передачи данных RS485;
<b>BOOT_SET_USB</b>	– установка порта для передачи данных USB;
<b>BOOT_BEACON_OFF</b>	– выключение маяка в эфире;
<b>BOOT_BEACON_ON</b>	– включение маяка в эфире.

Вход в режим осуществляется при зажатии кнопки "CFG" и включении питания. Текущие значения отображаются на индикаторах "RX/TX" и "MODE". Выбор осуществляется отпуская кнопку в нужный момент в соответствии со схемой:



- Удерживать кнопку для выбора значения.
- Отпустить в течении 0.9-1.2с после очередной индикации.



На команды загрузчика выделенная область можно ввести блокировку исполнения битом **bBootCmdLock** в регистре \$LOCK в командном режиме (по умолчанию блокировка выключена).

#### 6.1.1 ВХОД В КОМАНДНЫЙ РЕЖИМ ПО УМОЛЧАНИЮ (BOOT\_DEF)

Индикация для отпускаяния



Вход в командный режим по USB порту с параметрами обмена **9600,8N1**. Способ используется когда не известен активный порт и параметры его настройки (\$COM).

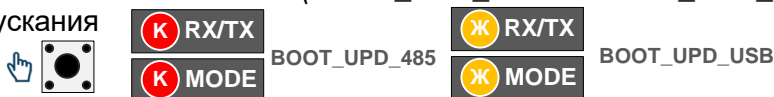
Также, модуль ввода-вывода переходит в тестовый режим (см. раздел [Тест ВХОДОВ/ВЫХОДОВ](#)).

#### 6.1.2 РЕЖИМ ТЕСТ (BOOT\_TEST)

На данный момент функциональность не определена.

### 6.1.3 РЕЖИМ СМЕНЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (BOOT\_UPD\_RS485/BOOT\_UPD\_USB)

Индикация для отпускания



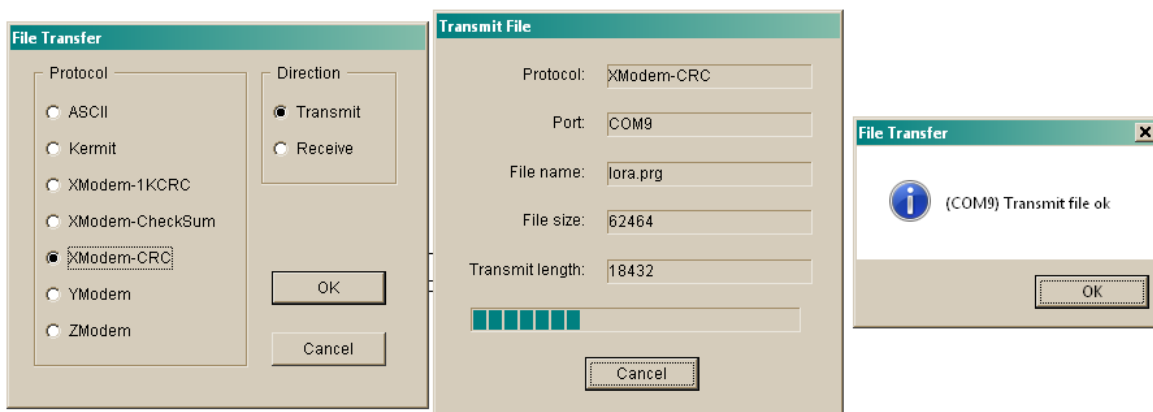
Для смены ПО следует:

- запустить на компьютере терминальную программу с параметрами **57600 8N1**;
- подключить модуль к интерфейсу RS485 или USB;
- перевести модуль в режим смены ПО командой **BOOT\_UPD\_RS485** или **BOOT\_UPD\_USB**. Модуль должен выдать приветствие примерно такого содержания:

```
Spektr 868IO LoRa (c)Rateos 1998-2023 bv1.0;
Erase ...done
Ready rcv by XModem-CRC...CCCCC
```

- сменить ПО можно также из командного режима по команде **\$UPD**.

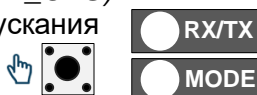
С помощью инструмента «Передача файлов» отправьте в модуль файл с нужной версией встроенного ПО в протоколе Xmodem-CRC и дождитесь окончания передачи.



По окончании передачи, в случае правильного обновления микропрограммы модуль автоматически перезапустится и перейдет в рабочий режим.

### 6.1.4 АЛЬТЕРНАТИВНАЯ КОНФИГУРАЦИЯ (BOOT\_CFG)

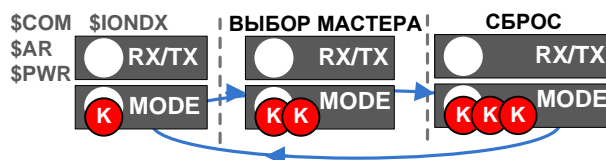
Индикация для отпускания



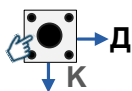
В данном режиме доступно:

- Изменение регистров \$COM, \$AR, \$PWR и \$IONDX;
- Выбор работы в режиме \$IOMASTER
- Сброс настроек по умолчанию.

После отпускания кнопки на индикаторах запускается цикл выбора действий. После каждого состояния необходимо нажать кнопку CFG в течении 1 с, что будет соответствовать выбору действия.



Выход из конфигурации (кроме сброса) отсутствует. Поэтому, после просмотра или редактирования значений необходимо переключить питание. Или, если не нажимать кнопку CFG в течении ~20с, модуль автоматически перезагрузится по бездействию.



В следующих режимах выбора и редактирования данная пиктограмма означает, что требуется нажатие на кнопку **CFG**.

"K" означает **К**ороткое нажатие, "Д" - **Д**линное (больше 1 сек).

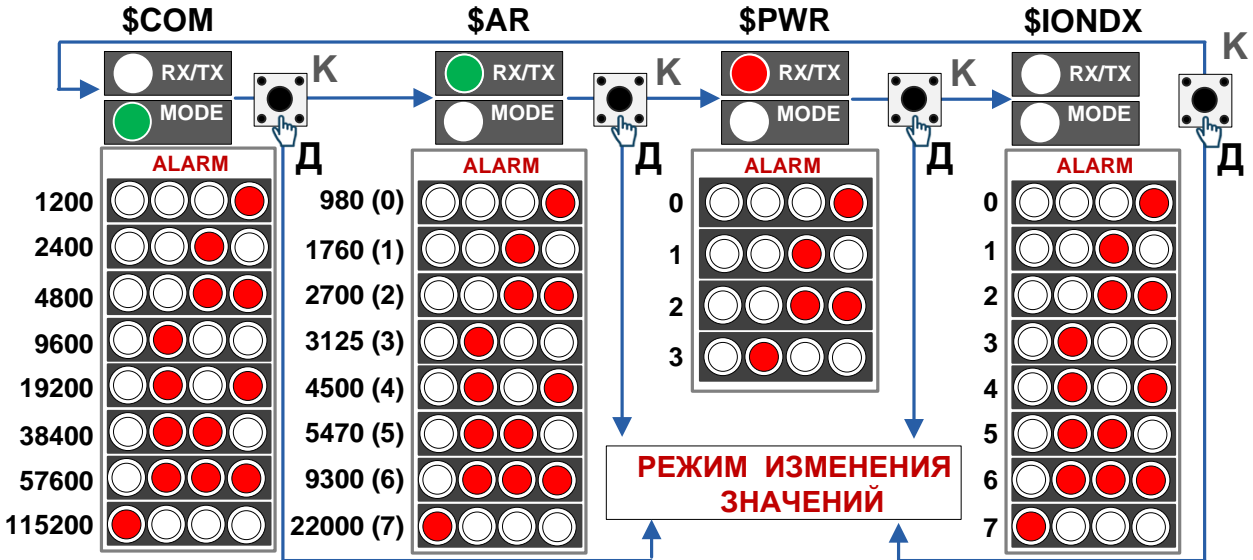
**ИЗМЕНЕНИЕ РЕГИСТРОВ \$COM, \$AR, \$PWR и \$IONDX.**

Для параметра \$COM игнорируется четность, число стоповых бит и размер слова. При изменении параметра в данном режиме четность, число стоповых бит и размер слова устанавливается всегда в значение "8N1".

Если текущее значение \$IONDX больше 7, все светодиоды ALARM загорятся. При сохранении параметра \$IONDX устанавливается режим \$IOSLAVE, базовый адрес \$IOADR устанавливается в значение 0x0A.

Всего доступны два режима - режим просмотра и изменения.

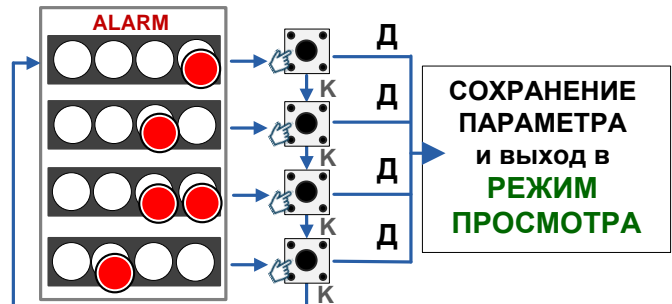
**Режим просмотра.** Текущее значение отображается на светодиодах ALARM в соответствии с таблицей ниже:



**Режим изменения значений.**

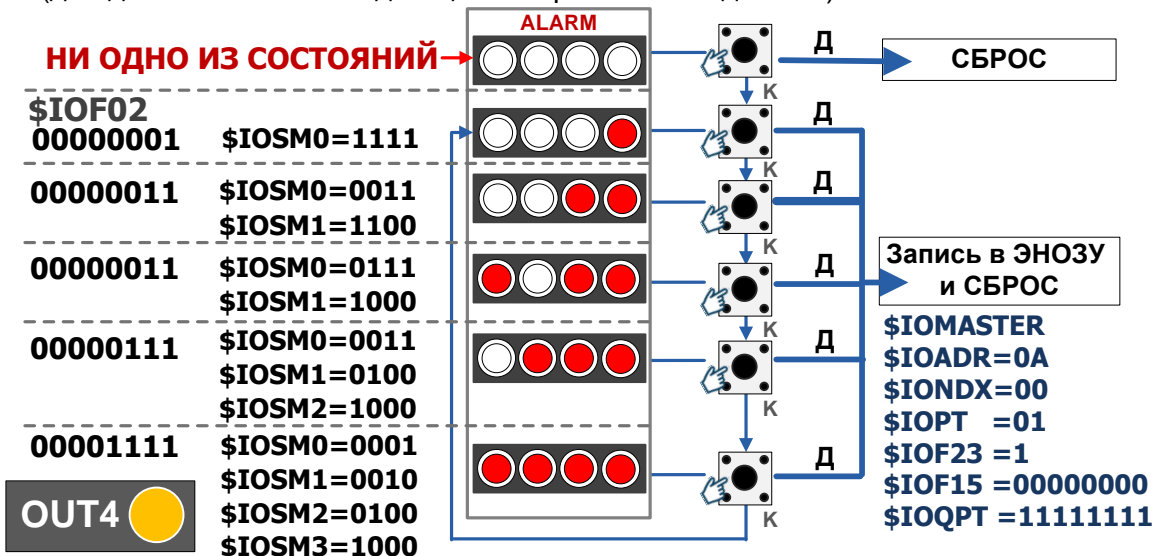
На светодиодах ALARM показываются варианты **миганием красным**.

На рисунке показан пример изменения параметра \$PWR.



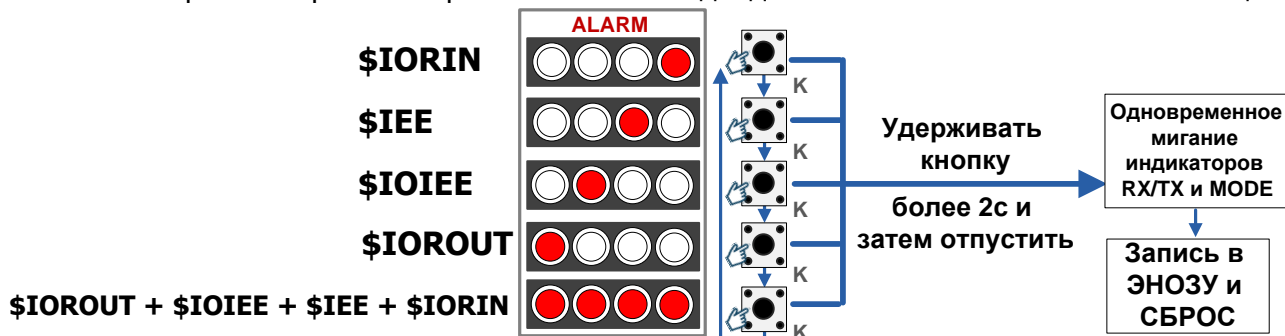
**ВЫБОР МАСТЕРА.**

Выбор работы модуля в режиме \$IOMASTER выбирается в соответствии с таблицей (для дополнительной индикации загорается выход OUT4):



### СБРОС НАСТРОЕК.

Выбор типа сброса отображается на светодиодах ALARM в соответствии с таблицей:



В случае успешного действия, индикаторы RX/TX и MODE начнут одновременно мигать желтым цветом, операция будет выполнена и модуль пересбросится. Если сброс настроек заблокирован (в регистре \$LOCK), то при входе в режим сброса настроек индикаторы RX/TX и MODE сразу начнут мигать красным цветом и модуль просто пересбросится.

#### 6.1.5 УСТАНОВКА АКТИВНОГО ПОРТА (BOOT\_SET\_RS485/BOOT\_SET\_USB)

Индикация для отпущания (моргание)



В модуле два независимых равно функциональных порта - RS485 и USB. В процессе работы один порт используется для передачи пользовательских данных (**активный**), на другой выводится отладочная информация (DEBUG порт). По умолчанию, RS485 порт предназначен для передачи данных, а на USB выводится отладочная информация. При необходимости, назначение портов можно поменять местами. Смена осуществляется либо через режим загрузчика, либо в командном режиме регистрами \$RG02-\$RG03 (см. раздел ["Технологические параметры. Регистры RG"](#)) или командой [\\$EXCHP](#).

Активный порт	DEBUG порт	Переход
RS 485	USB	1) По команде BOOT_SET_485 2) \$RG02-\$RG03 = 0xC4,0x85
USB	RS 485	1) По команде BOOT_SET_USB 2) \$RG02-\$RG03 = 0xC2,0x32

Параметры активного порта задаются командой \$COM. Параметры отладочного порта фиксированы (57600 8N1).

Если активным задан порт USB, то при включении питания и переходе в рабочий режим, модуль моргнет 2 раза зеленым цветом на индикаторе "MODE". Индикация активного порта RS485 не осуществляется.

В случае успешного действия, индикаторы RX/TX и MODE начнут одновременно мигать желтым цветом, операция будет выполнена и модуль пересбросится. Если режим загрузчика заблокирован (в регистре \$LOCK), то при входе в режим установки порта индикаторы RX/TX и MODE начнут мигать красным цветом и модуль просто пересбросится.



Нет необходимости каждый раз задавать активный порт. Текущая конфигурация запоминается в ЭНУЗУ модуля.

#### 6.1.6 ВКЛЮЧЕНИЕ РЕЖИМА "МАЯК" (BOOT\_BEACON\_OFF/BOOT\_BEACON\_ON)

Индикация для отпущания (моргание)



Вход в режим "Маяк" осуществляется через загрузчик или в командном режиме регистрами \$RG04-\$RG05 (см. раздел "[Технологические параметры. Регистры RG](#)").

"Маяк"	Переход
ВЫКЛ.	1) По команде BOOT_BEACON_OFF 2) \$RG04-\$RG05 = 0xFF,0xFF
ВКЛ.	1) По команде BOOT_BEACON_ON 2) \$RG04-\$RG05 = 0xBE,0xAC

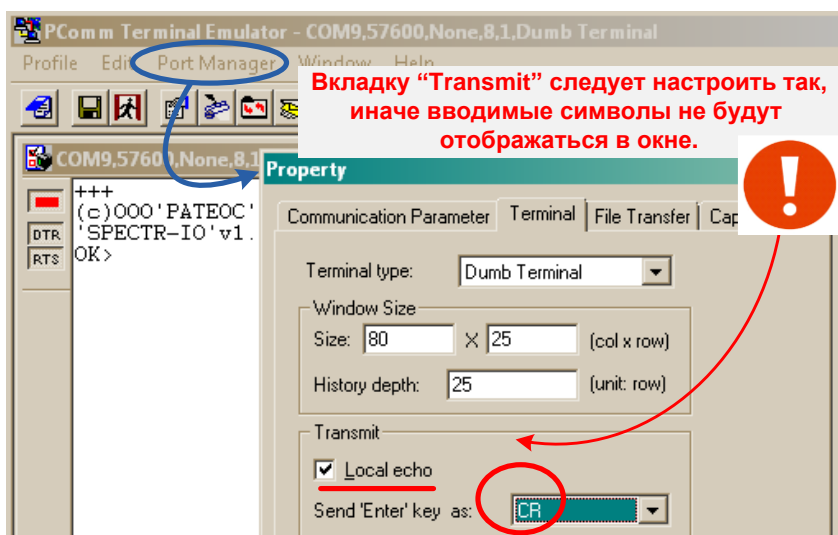
В случае успешного действия, индикаторы RX/TX и MODE начнут одновременно мигать желтым цветом, операция будет выполнена и модуль пересбросится. Если режим загрузчика заблокирован (в регистре \$LOCK), то при входе в данные режимы индикаторы RX/TX и MODE начнут мигать красным цветом и модуль просто пересбросится.

Подробнее о режиме "Маяк" см раздел "[Режим Маяк](#)".

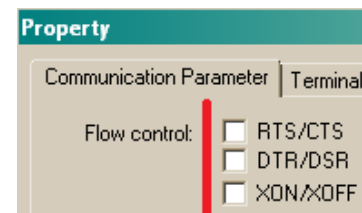
## 6.2 РЕЖИМ "КОМАНДНЫЙ"

Командный режим служит для конфигурации параметров модуля с помощью текстовых команд. Предварительно необходимо установить на компьютер драйверы для работы по USB (см. раздел "[Интерфейс USB. Установка драйвера](#)"). Признаком правильной установки драйверов является появление виртуального COM-порта в диспетчере устройств компьютера при подключении модуля (без восклицательных знаков и т.д.).

Для конфигурации потребуется любая терминальная программа (терминал). Мы рекомендуем терминал Моха PComm, который есть на нашем сайте (<http://rateos.ru/files/PComm210.zip>).



В разделе "Communication Parameter" **отключить** Flow Control:



В командный режим модуль переходит:

- в режиме загрузчика по команде **BOOT\_DEF**. В этом случае конфигурация будет осуществляться по USB порту с параметрами обмена **9600,8N1**. Такой способ удобен, когда неизвестны параметры и тип порта для конфигурации.
- из рабочего режима при получении по любому интерфейсу RS-485 или USB трех последовательных символов «+++». Как минимум за 1 с до начала и 1 с после окончания последовательности «+++» не должно быть никаких других символов. Терминальная программа должна быть настроена на соответствующую скорость. Если вход осуществляется по отладочному порту, то на 57600 8N1. Если по порту передачи данных то на скорости, на который он настроен командой \$COM (по умолчанию - 9600 8N1). После входа данным способом, тип порта автоматически запоминается и используется в дальнейшем для входа по кнопке "CFG".
- из рабочего режима при нажатии кнопки «CFG». В этом случае вход будет осуществляться по порту, по которому ранее был вход через последовательность "+++" или по умолчанию по USB, если последовательность "+++" никогда ранее не вводилась.



Признаком нахождения модуля в командном режиме является загорание индикатора «MODE» **красным** цветом (или **желтым**, если индикатор «MODE» в текущий момент сигнализирует наличие данных в буфере для передачи). На порт выводится примерно следующее приветствие:

```
(c)OOO'PATEOC' 23/02/2023 (17:36)
'SPECTR 868IO LoRa'v1.01 #sn:RL1000001 #name:
OK>
```

Далее, напечатайте в окне терминальной программы команду \$DMP и нажмите клавишу "Enter" — в ответ модуль должен выдать список своих основных параметров.

- Если приветствие не появилось:
- не установлены или установлены некорректно драйвера USB (если вход по USB);
  - выбран не тот COM-порт в программе и т.д.

Если видны нечитаемые символы: параметры COM-порта в программе не соответствуют параметрам порта модуля;

```
OK> $DMP
CH=00 TXID=FFF COM=9600,8N1
PWR=0 MYID=002 DAT=EOT MAXP=0
AR=3 ACKT=016 EOT=0015 PLEN=000 EOC=0AT
BCN=1 RESPT=000 FEC=- (MaxSz=234)
BCT=005 RETRY=000 MDA=00000001 CONCT=000
IOADR=00+IONDX=00 (00h) TFL=---- RPT=-
IORPT=11111111 RFL=---- RTFL=----
MaxTt=715 ms,ACKT >=016
IOSLAVE IOALR=F0
IOSFT=010 IOSFV=----
IOINL=---- IODBN=00000000h
IOINF=00 IOFR[1:0] =0000,0000
IOUTM=---- IOUT[3:0]t=0000,0000,0000,0000
IOINT=00000000
IOMOD=00000000
OK>
```

Все команды, посылаемые в модуль, должны начинаться с префикса \$ (0x24) и заканчиваться символами CR или LF (0x0D,0x0A) – клавиша **Enter**.

После получения команды, возможны следующие реакции модуля:

```
OK> $hello ← Ответ "ER>" - ввод неизвестной команды
ER> $COM=57600,8n1 ← Ответ "OK>" - команда выполнена.
OK> $bct=999 ← Ответ "??>" - команда введена правильно, но
??> ← содержит некорректные параметры.
```



После изменения параметров, необходимо выполнить команду **\$S** (save), чтобы изменения сохранились в энергонезависимой памяти модуля.

Для выхода из командного режима необходимо выполнить команду \$E (exit) или нажать кнопку "CFG".

Для того, чтобы новые изменения вступили в силу, необходимо выполнить команду \$R (reset) или выключить и включить питание заново.

Если Вы не уверены, что модуль имеет заводские установки, рекомендуем сбросить его настройки в заводские командой **\$IEE** — это гарантирует известное исходное состояние модуля. После ввода этой команды модуль автоматически пересбросится и необходимо будет снова войти в командный режим.

### 6.2.1 ПАРОЛЬ НА КОМАНДНЫЙ РЕЖИМ

В модуле имеется возможность установить пароль на изменение настроек в командном режиме. Пароль устанавливается командой **\$CFL**. После установки пароля изменение настроек в командном режиме блокируется. При этом не изменяющие параметры модуля команды (\$DMP, \$LOG, \$SCAN и т.д.), не блокируются. Для разблокировки изменения настроек необходимо ввести команду \$CFU. Разблокировка действует на весь последующий сеанс работы модуля (для повторной блокировки необходимо пересбросить модуль).

## 6.3 РЕЖИМЫ ПЕРЕДАЧИ ДАННЫХ

### 6.3.1 РЕЖИМ ПЕРЕДАЧИ ДАННЫХ «ПРОЗРАЧНЫЙ»

В данный режим модуль переходит сразу после включения питания. В этом режиме:

- все данные, поступившие на последовательный порт модуля, доставляются на последовательный порт адресуемого модуля (или модулей, если используется широковещательный или групповой адрес);
- все данные, принятые из эфира, выдаются на последовательный порт модуля, если адресованы ему.

Слово «прозрачный» в названии режима не означает, что модули передают в радиоэфир непосредственно данные, полученные от внешнего оборудования: в эфире они работают с собственным протоколом, обеспечивающим адресацию, проверку доставки, помехоустойчивое кодирование и т. д.

Прозрачность означает, что данные, поступившие на последовательный порт модуля не нуждаются в какой либо специальной обработке и будут доставлены на последовательный порт (порты) удалённого модуля без изменений. То есть можно говорить о прозрачности на уровне «последовательный порт одного модуля» — «последовательный порт (порты) удалённого (удалённых) модулей».

Благодаря такой прозрачности практически любое внешнее оборудование, работающее по проводам, будет работать и через радиомодули, не замечая, что работает через них — для него работа через модули не будет отличаться от работы по проводам.

Протоколы, по которым работает внешнее оборудование (ModBus, Болид и т.д.) практически не имеют значения, поскольку модули будут прозрачны для этих протоколов.

Все это касается как простых систем, когда нужно связать пару устройств с помощью двух модулей, так и более сложных систем сбора данных из множества удалённых объектов. Никаких специальных ограничений на количество объектов в системе с радиомодулями нет: можно сказать, что если система работает по проводам, она практически наверняка будет работать и через радиомодули.

Передача данных в эфир начинается в зависимости от выбранного типа входящих данных (протокола) и задается параметром [\\$DAT](#). Возможные варианты описаны в разделе "[Параметры входящих данных](#)".

Независимо от приёма данных по интерфейсу модуль принимает данные из эфира. Если принятый пакет адресован ему и информация, содержащаяся в пакете, не содержит ошибок, она записывается во внутренний буфер модуля и выдается на последовательный порт.

### 6.3.2 ОСОБЕННОСТИ РЕЖИМА ПЕРЕДАЧИ ДАННЫХ «ПРОЗРАЧНЫЙ»

Как говорилось ранее, при работе радиомодулей в прозрачном режиме внешнее оборудование «не замечает» их и работает так же, как и по проводам.

При этом стоит помнить о некоторых особенностях передачи данных через радиомодули. Основные причины, по которым система отказывается работать через модули (естественно, предполагается, что осуществлена проверка связи между самими модулями):

- задержка при передаче данных;
- разбиение данных на части;

#### **ЗАДЕРЖКА ПРИ ПЕРЕДАЧИ ДАННЫХ.**

Главное (и наиболее часто приводящее к тому, что оборудование «не хочет» работать через модули) отличие работы через модули от работы по кабелю заключается в задержках при передаче данных. Если при работе по кабелю задержки практически отсутствуют, то при работе через радиомодули эти задержки могут составлять сотни миллисекунд и даже секунды.

Такие задержки возникают из-за того, что путь прохождения данных при работе через модули «удлинняется»: модуль должен принять данные по порту в свой буфер, понять, что пакет данных кончился и можно передавать его в эфир, включить передатчик, осуществить

передачу данных по радио, приёмный модуль должен принять данные в свой буфер, проверить целостность и выдать их на свой последовательный порт.

Для систем, построенных по топологии «звезда» с протоколом опроса удалённых объектов по принципу «запрос мастера — ответ слейва» наличие дополнительных задержек при работе через модули приведет к увеличению времени между отправкой запроса мастером и получению им ответа от слейва. В протоколах таких систем практически всегда определено время (тайм-аут) ожидания мастером ответа от слейва — если ответ не получен за это время, мастер считает, что слейв не отвечает и выдает ошибку.

Вполне может оказаться, что при работе через модули ответы от слейвов приходят с опозданием — когда мастер их уже не ждет и сообщает об отсутствии связи или неполучении ответа от слейва.



Для исправления такой ситуации необходимо увеличивать время ожидания ответов в настройках мастера системы. Как правило (практически всегда), протоколы типа ModBus предусматривают такое увеличение — мастер (пульт, ПЛК, OPC-сервер) должен иметь возможность конфигурации временных параметров протокола.

Минимальное время ожидания ответа должно быть не менее, указанное в подсказке "АСКТ>=" при выводе команды \$DMP. К нему рекомендуется добавить еще 100-300мс.

Если в системе используются ретрансляторы, то задержки при доставке данных через цепочку ретрансляторов будут длиннее (примерно в 2 раза на каждый ретранслятор).

#### РАЗБИЕНИЕ ДАННЫХ НА ЧАСТИ.

Другой вероятной (хотя и гораздо менее частой) причиной отказа работы системы через модули является разбиение последовательности данных на несколько пакетов при передаче их через модули.

Одним из условий формирования пакета данных для передачи в эфир является пауза между приходящими данными - если пауза превышает заданное значение, данные "защелкиваются" и начинается формироваться пакет. Все идущие следом данные в этот пакет уже не попадут и будут переданы следующим пакетом. Длительность паузы задается параметром \$EOT.

К примеру, если на порт модуля приходит поток из 100 байт с паузой после 80-го байта больше чем параметр \$EOT, то в эфире будет 2 пакета данных - 80 и 20 байт.

При этом получатель получит все 100 байт данных (прозрачность модулей в смысле сохранности данных остается), но двумя пакетами по 80 и 20 байт с задержкой между ними. Получатель данных может быть не готов к такому и может воспринять разрыв в пакете данных, как ошибку. Чтобы исключить такую ситуацию, следует увеличить параметр \$EOT.

По умолчанию \$EOT=0015, что соответствует 15 мс и паузе в ~15 байт на скорости по умолчанию 9600. Это достаточно для большинства применений. При увеличении скорости по последовательному порту, параметр \$EOT можно уменьшать, при уменьшении - увеличивать.

Кроме паузы между байтами в потоке данных следует также учитывать максимальный размер данных, которые модуль может передать в 1 пакете. Размер пакета зависит от типа помехоустойчивого кодирования и использовании шифрования. Максимальный размер данных передаваемых в эфир в одном пакете не может превышать 234 байт, поэтому в случае если на порт модуля поступают массивы данных большей длины, модуль при передаче по радио разобьет их на несколько пакетов. При этом приёмный модуль выдаст принятые из эфира данные на свой порт также несколькими «порциями». В этом случае могут возникнуть определенные проблемы с объединением таких пакетов в единый блок. Решение такой проблемы, если она возникает, должно производиться на более высоком уровне системы или с помощью активизации режима конкатенации данных.

С помощью искусственного метода конкатенации можно "склеивать" данные из эфира - см. раздел "[Буферизация принятых данных из эфира. Конкатенация данных](#)"

## 6.4 РЕТРАНСЛЯЦИЯ ПАКЕТОВ

Модуль способен ретранслировать пакеты, не утрачивая своих основных функций.

В системе может быть до 8 ретрансляторов, номера которых задаются командой **\$RPT**. Адреса ретранслируемых пакетов задаются командой **\$RID**.

В каждом пакете, передаваемом в эфир, находится специальное ретрансляционное поле (РП), которое обрабатывается каждым активным ретранслятором. Модуль, работая в режиме ретранслятора, принимая кадр из эфира, анализирует РП и адреса RIDxx. Если в РП отсутствует маркер ретрансляции для данного модуля и адрес в пакете совпал с одним из RIDxx, принятый кадр записывается во внутреннюю ретрансляционную очередь, работающую по принципу FIFO (первый вошел, первый вышел).

Ретрансляция сообщений, находящихся в очереди, имеет меньший приоритет по сравнению с передачей собственных данных модуля. Ретрансляция информационных пакетов происходит только в случае правильного приёма всех данных пакета без ошибок.

Поскольку модуль в эфире работает в полудуплексном режиме, при применении ретрансляторов общая скорость передачи уменьшается прямо пропорционально количеству активных ретрансляторов, задействованных в процессе передачи данных между абонентами.

Для активизации режима ретрансляции достаточно присвоить модулю уникальный ретрансляционный номер (команда **\$RPT**) и ввести хотя бы один адрес отправителя/получателя, пакеты которого необходимо ретранслировать (команда **\$RID**).

Более подробно о дополнительных возможностях по ретрансляции см. в разделе [«Расширенные возможности приема и ретрансляции пакетов»](#).

## 6.5 РЕЖИМЫ РАБОТЫ ПО ЭФИРУ (ГАРАНТИИ ДОСТАВКИ ДАННЫХ АДРЕСАТУ)

### 6.5.1 АДРЕСАЦИЯ

В модуле возможно использование 4096 (000...FFF) адресов. Адреса разделены на индивидуальные, групповые и широковещательный. Адрес содержит 3 цифры в 16-ричном формате.

- Адрес является **широковещательным**, если он равен FFF.
- Адрес является **групповым**, если он содержит цифру F.
- Все остальные адреса являются **индивидуальными**.

Каждый модуль имеет два адреса – адрес отправителя (собственный) и адрес получателя. Адрес отправителя задаётся командой **\$MYID**, получателя — командой **\$TXID**.

Адрес получателя может быть индивидуальным, групповым или широковещательным.

Адрес отправителя может быть только индивидуальным. Два или более модуля не могут иметь одинаковый адрес MYID.

Модули в сети могут быть объединены в группы - часть цифр их «собственного» адреса должны быть одинаковыми. Например, адреса 120, 121,...12E образуют группу. Для передачи данных всем адресатам данной группы необходимо адресу получателя присвоить значение 12F.

Пакеты, передаваемые в эфире, содержат информацию об адресах, на основании этой информации каждый принявший пакет модуль может судить о «принадлежности» и «назначении» данного пакета. Таким образом, нет необходимости в отдельном признаке способа распределения данных между модулями («точка-точка», «групповой» или «широковещательный»), режим работы задаётся только адресами. Например, если один из модулей имеет TXID=12F, его пакеты будут «принимать» все модули, адреса MYID которых начинаются с 12. Если же, например, адрес TXID=205, его пакеты будет «принимать» только модуль с адресом MYID=205.

Модуль может работать в эфире в двух основных режимах – широковещательный (или групповой) и индивидуальный («точка-точка»).

### 6.5.2 ШИРОКОВЕЩАТЕЛЬНЫЙ РЕЖИМ

Модуль автоматически работает в широковещательном режиме, если параметр TXID не является индивидуальным. В этом случае модуль отправляет в эфир «широковещательные» пакеты, которые «слышат» все модули (или группа модулей).



В широковещательном режиме гарантия доставки пакета данных адресату отсутствует.

Это не является проблемой если протоколы, по которым работает внешнее оборудование, обеспечивает контроль доставки данных на своём уровне, а именно так и есть в подавляющем большинстве случаев (промышленные протоколы ModBus и им подобные).

Вероятность доставки пакетов может быть увеличена вспомогательными методами:

- помехоустойчивым кодированием (см. раздел "[Формат пакет в эфире](#)");
- мультипликацией данных (см. раздел "[Мультипликация данных](#)");
- снижением скорости передачи в эфире (см. раздел "[Скорость обмена данными между модулями в эфире \(\\$AR\)](#)").

### 6.5.3 ИНДИВИДУАЛЬНЫЙ РЕЖИМ («ТОЧКА-ТОЧКА»)

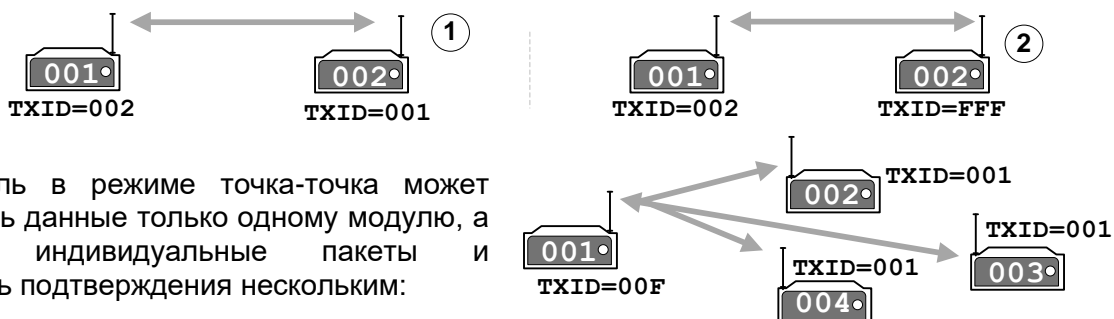
Для работы в индивидуальном режиме («точка-точка») параметр TXID не должен являться широковещательным или групповым. В этом случае модуль отправляет в эфир «индивидуальные» пакеты, которые принимает только модуль, параметр MYID которого равен TXID передающего модуля.



При получении «индивидуального» пакета модуль автоматически отправляет подтверждение о его приёме отправителю пакета. Отправитель же при неполучении такого подтверждения повторяет пакет. Таким образом, в режиме «точка-точка» имеются **гарантии** доставки данных.

Получая индивидуальный пакет, модуль создает "соединение" с этим модулем, начиная при этом вести статистику приёма пакетов от него.

На каждый правильно принятый информационный пакет отправляется короткое подтверждение АСК (если данные приняты с ошибками, подтверждение не отправляется). Удаленный модуль при этом может быть как в режиме точка-точка (1), так и в широковещательном режиме (2):



Модуль в режиме точка-точка может передавать данные только одному модулю, а получать индивидуальные пакеты и отправлять подтверждения нескольким:

Если скорость выдачи данных на последовательный порт много ниже общей скорости поступления данных из эфира и приёмные буферы заполнены, передающему модулю не передаётся подтверждение о приёме пакета, что вынуждает его повторить передачу пакета данных через время, задаваемое параметром \$ACKT. Таким образом, потеря данных из-за несоответствия скоростей исключена.

### 6.5.4 ОСОБЕННОСТИ ПРИМЕНЕНИЯ РЕЖИМА «ТОЧКА-ТОЧКА»

Если предполагается использовать модули для связи только двух объектов между собой, то на первый взгляд более логичным выглядит использование режима «точка-точка», а не широковещательного, поскольку в режиме «точка-точка», в отличие от широковещательного режима, модули «гарантируют» доставку данных — отправив данные, модуль ждёт от адресата подтверждения о получении, и в случае неполучения подтверждения будет повторять отправку данных до получения подтверждения.

Тем не менее, когда внешнее оборудование использует протоколы с собственной адресацией (Болид, Modbus и подобные), лучше даже в системах из 2-х объектов использовать широковещательный режим (без подтверждений).

Дело в том, что подтверждение доставки на уровне модулей вносит в работу системы элемент непредсказуемости и неуправляемости, что в некоторых случаях может привести к неработоспособности системы. Это связано с тем, что на работу протокола обмена данными на уровне внешнего оборудования накладывается собственный протокол обмена данными на уровне модулей.

Как правило, даже в случае системы из двух объектов, модули работают по принципу «запрос-ответ». Мастер периодически посылает запросы и ждет на них ответ от слейва. Представим, что по какой-то причине один из запросов мастера не был доставлен с первого раза (помехи в радиоканале) и модуль, запрограммированный на работу в режиме «точка-точка» начинает повторять его несколько раз, чтобы гарантировать доставку. Мастер при этом уже не дождался ответа на этот запрос и посылает в модуль следующий запрос, который попадает в буфер модуля и ждет там своей очереди на отправку (модуль занят доставкой предыдущего запроса).

Таких запросов может накопиться несколько, и при возобновлении связи все они будут переданы в эфир, приняты слейвом, и слейв ответит на все эти запросы. Мастер может быть не готов к такой ситуации — он уже и «забыл» про не отвеченные запросы, а тут они будут получены.

В худшем случае (при достаточно продолжительном пропадании связи) может случиться также ситуация, когда полностью заполнится буфер модуля и данные на входе в модуль будут просто потеряны.

Таким образом, при использовании режима «точка-точка» возможны ситуации, когда время получения ответа на запрос становится непредсказуемым. Кроме этого, есть вероятность выхода системы из-под контроля и потери данных — модули могут быть заняты обеспечением гарантии доставки уже не нужных данных, а нужные данные будут становиться ненужными или вовсе теряться.

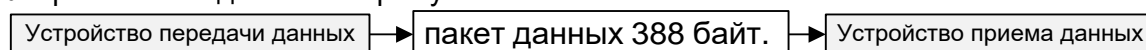
В широкополосном же режиме неуправляемых и необъяснимых ситуаций, когда модули «живут своей жизнью», обеспечивая гарантированную доставку данных, возникнуть не может. Худшее, что может случиться в случае проблем со связью — на некоторые запросы не будут получены ответы, а при появлении связи работа системы автоматически быстро возобновится.

## 6.6 БУФЕРИЗАЦИЯ ПРИНЯТЫХ ИЗ ЭФИРА ДАННЫХ. КОНКАТЕНАЦИЯ ДАННЫХ

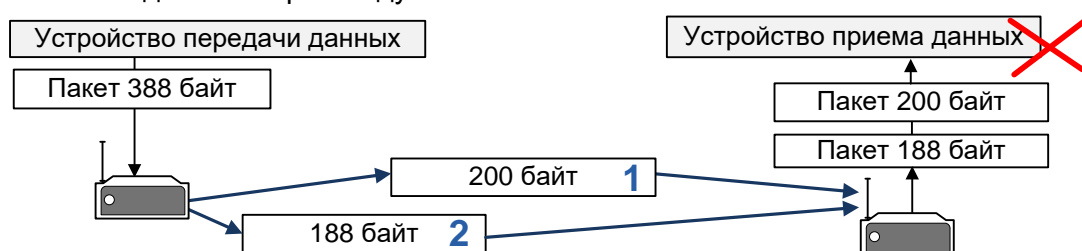
На принятые из эфира данные в модуле предусмотрен буфер. Информационный пакет от каждого модуля направляется в индивидуальный буфер. В случае если скорость обмена по эфиру намного превышает скорость обмена по последовательному порту это предотвращает потерю данных. Данные приёмных буферов последовательно передаются на последовательный порт модуля в порядке поступления их из эфира.

Некоторые протоколы передачи данных не подразумевают тайм-аут между символами внутри непрерывного сообщения. Размер непрерывного сообщения может превышать максимальный размер пакета, передаваемого в эфир модулем. Модуль будет разбивать это сообщение на части. В таком случае, данные могут быть доставлены корректно, но с паузами между частями. Эти данные не будут восприняты приемным оборудованием или будут восприняты как некорректные.

Устройства соединены напрямую:



Устройства соединены через модули:



Модуль не может передать все 388 байт одним пакетом. Он разбивает его на части и каждую часть маркирует специальными маркерами 'L' (Last) или 'M' (More). Приемный

модуль, при активации данной функции может их конкатенировать (объединить) в один пакет (по признакам L/M) и выдать в порт единым целым.

Для функции конкатенации в модуле выделены 2 буфера по 900 (общий размер 1 пакета для конкатенации не может превышать 900 байт).

Конкатенация активизируется параметром \$CONCT, который является таймером с градацией в 300мс. Таймер задает время, в течении которого считается, что еще ожидается прием пакета с признаком 'L' (последний).

Конкатенация данных происходит только в том случае, если параметр \$CONCT отличен от 0, иначе данные выдаются на последовательный порт модуля по мере поступления из эфира.

При разрешении конкатенации данных выдача принятых данных на последовательный порт модуля происходит в следующих случаях:

- **буфер полон.** Приходящие данные имеют признак M и их общий объем, таким образом, превышает или равен размеру буфера;
- **получен маркер "L" (признак последних данных).**

Если по истечении времени \$CONCT не получен признак 'L', данные в буфере считаются не актуальными и на порт не выдаются.

## 7 ФОРМАТ ПАКЕТА В ЭФИРЕ.

Максимальный *общий* размер пакета в эфире составляет 256 байт. Перед передачей в эфир данные проходят следующие этапы предварительной обработки:

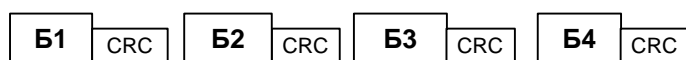
- проверка входных данных на формат 7 бит. Сжатие данных в случае положительного результата тестирования;
- разбиение пакета на 32-байтные блоки и вычисление 8-битной контрольной суммы для каждого блока (CRC8);
- добавление избыточности (помехоустойчивое кодирование и перемежение) в случае активации этой функции;

При приёме данных из эфира модуль создает переменную маркеров правильности приёма каждого блока для каждого текущего информационного пакета: после процедур декодирования информации вычисляется контрольная сумма каждого блока и сравнивается с полученной. Если контрольные суммы совпадают, маркер для этого блока устанавливается в «1», и блок копируется во внутренний буфер. Иначе маркер устанавливается в «0», а блок игнорируется. Если после обработки пакета все маркеры равны «1», данные считаются корректными и передаются в порт модуля. Иначе ожидается повторный приём данного пакета (если это подразумевается установленным режимом работы модулей). При приёме другого (нового) информационного пакета переменная маркеров сбрасывается в «0».

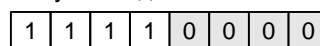
Применение «технологии» маркеров уменьшает время достоверной и гарантированной передачи информации в случае непрохождения пакета с первого раза.

Иллюстрация работы маркеров приводится на рисунке ниже.

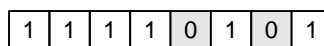
Информационный пакет. Длина 120 байт



Начальное состояние маркеров при получении длины пакета

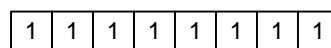
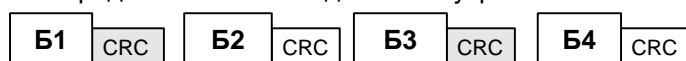


① Передача. Искажение данных внутри блоков 2 и 4



Состояние маркеров после 1-го приема

② Передача. Искажение данных внутри блоков 1 и 3



Состояние маркеров после 2-го приема

Как видно из рисунка, после «первой» передачи всего пакета контрольная сумма у блоков 2 и 4 не совпадает с принятой, а блоки 1 и 3 приняты без ошибок. После «повторной» передачи блоки 1 и 3 приняты с ошибками, в то время как блоки 2 и 4 приняты без ошибок. Благодаря маркерам для «восстановления» полного пакета в данном случае потребовалась только одна повторная передача, поскольку нет необходимости повторять пакет до тех пор, пока все блоки одновременно будут приняты корректно.

Использование маркеров эффективно в любых режимах работы модуля по эфиру, кроме широковеЩательного, при условии, что информационный пакет передаётся один раз (параметр [\\$BCN](#) равен 1).

### 7.1 ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ

Для уменьшения повторных передач информационных пакетов и, следовательно, для увеличения пропускной способности радиоканала, а также для более надёжной передачи данных кроме использования подтверждений, можно использовать прямое исправление ошибок. Для этого в модуле реализованы несколько способов помехоустойчивого кодирования (FEC): код Рида-Соломона (RS) и код Хэмминга (HAM). Характеристики кодов приведены в следующей таблице:



Код	Скорость	Характеристики
RS(7,5)	0,714	Каждое кодовое слово состоит из 15 информационных (i) и 6 проверочных (p) бит. Исправление 3 (информационных) бита.
RS(7,3)	0,429	Кодовое слово состоит из 9i+12p бит. Исправление 6 бит.
RS(15,11)	0,733	Кодовое слово состоит из 44i+16p бит. Исправление 8 бит.
RS(15,9)	0,600	Кодовое слово состоит из 36i+24p бит. Исправление 12 бит.
HAM(12,8)	0,667	Кодовое слово состоит из 8i+ 4p бит. Исправление 1 бит.
НЕТ	1	-

Использование FEC приводит к снижению «информационной» скорости данных в эфире (скорость кода). Например, при использовании кода RS(7,3) при «физической» скорости в эфире 9600 бод получим «информационную» скорость  $9600 \times 0,429 = 4118$  бод.

Как упоминалось выше, максимальный размер данных в эфире включая заголовок, не может превышать 256 байт. Использование FEC подразумевает введение избыточности поэтому, размер полезных данных, которые можно передать в одном пакета также уменьшается. Ниже приведена таблица зависимости максимального размера данных в одном пакете от FEC в порядке увеличения размера пакета.

Код FEC	Максимальный размер данных, байт
RS(7,3)	101
RS(15,9)	139
HAM(12,8)	158
RS(7,5)	168
RS(15,11)	170
Без FEC	234

Если пользовательский объем данных при формировании пакета не уместится в максимальный размер при выбранном коде FEC, модуль автоматически меняет FEC для увеличения максимального размера пакета в соответствии с таблицей выше.

Например, если используется код RS(7,3), то при передаче 120 байт код FEC для текущего пакета изменится с RS(7,3) на RS(15,9). Или, если необходимо передать 235 байт и код FEC не используется, то в эфире будет 2 пакета - 234 байт и 1 байт.

Тип помехоустойчивого кодирования задается командой \$FEC.



Признак используемого кода передается в заголовке пакета, поэтому нет необходимости устанавливать одинаковый тип кода на всех модулях.

## 7.2 ПЕРЕМЕЖЕНИЕ

На практике часто искажаются не отдельные биты, а целые последовательности информационных бит (затухание и переотражение сигнала, кратковременные активные помехи), поэтому при использовании FEC эффективно применение перемежения информационных и проверочных символов.

Процесс перемежения заключается в передаче сначала первых элементарных символов каждого кодового слова, потом вторых, третьих и так далее. В случае возникновения пакетной ошибки после процедуры деперемежения ошибки равномерно распределяются в каждом кодовом слове. При этом повышается вероятность исправления ошибок в принятых данных.

Код	Исправляемых ошибок без перемежения, бит	Исправляемых ошибок с перемежением в блоке на 32 байта	
		бит	Байт
RS (7,5)	3	54	6,75
RS (7,3)	6	174	21,75
RS (15,11)	8	48	6
RS (15,9)	12	96	12
HAM(12,8)	1	32	4
Нет	0	0	0

При выборе типа FEC и перемежения необходимо учитывать характер помех в эфире, а также конкретное приложение или режим работы модуля. Например, в режиме «точка-

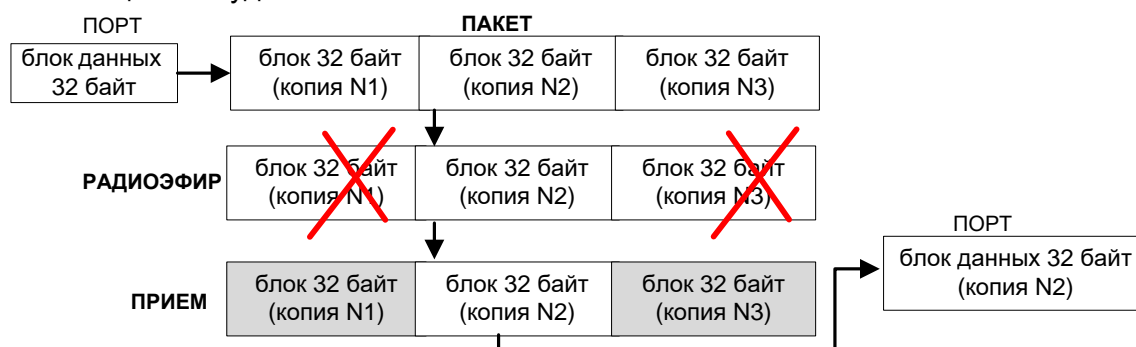
точка», когда неправильно принятый пакет будет ретранслирован, как правило, целесообразно использовать менее мощные коды или вовсе обойтись без них. В режиме же «точка-много точек» (широковещательный) гарантия доставки данных отсутствует, и для повышения вероятности доставки оправдано использование того или иного типа FEC. Включение/выключение процедуры в команде `$FEC`.

### 7.3 Мультипликация данных

Если размер пользовательских данных для текущего пакета значительно меньше максимального размера пакета в эфире при выбранном коде FEC, модуль копирует данные по блокам заполняя всю возможную длину пакета. Это увеличивает вероятность приема информации - на приемном конце модуль по контрольной сумме выбирает правильный блок и откидывает некорректный. В следующей таблице приведена зависимость мультипликации от размера данных и выбранного FEC.

FEC	Размер данных, байт	Размер мультипликации
НЕТ	< 33	3
	< 65	2
	< 97	2
	> 96	нет
HAM(12,8)	< 33	3
	< 65	2
	> 64	нет
RS(15,9)	< 36	3
	> 36	нет
RS(15,11)	< 33	3
	< 66	2
	> 65	нет
RS(7,3)	< 31	3
	> 30	нет
RS(7,5)	< 33	3
	< 66	2
	> 65	Нет

Например, для кода RS(7,5) размер мультипликации 3 означает, что если пользовательские данные для текущего пакета будут от 1 до 32 байт, то они будут дублированы 3 раза в одном пакета. Если размер данных превышает 65 байт, то мультипликации не будет.



Возможность мультипликации задается битом `$MDA.DataMultEnb`.

**По умолчанию мультипликация включена.**



При включенном шифровании данных AES, мультипликация автоматически отключается. Также, при мультипликации не осуществляется анализ данных в текущем пакете на 7 битное значение.

## 8 РАСШИРЕННЫЕ ВОЗМОЖНОСТИ ПРИЁМА И РЕТРАНСЛЯЦИИ

Расширенные возможности для ретрансляции и приёма пакетов подразумевают дополнительный анализ пакета по полученному ретрансляционному полю (РП) с целью его дальнейшей ретрансляции и/или приёма. Любой пакет, предназначенный для расширенной ретрансляции или приёма, идентифицируется адресом, который занимает одну ячейку во внутренней таблице модуля. Всего может быть запрограммировано до 8 различных ячеек. В адресе ячейки может быть как индивидуальный адрес, так и маска на подгруппу или группу. Наличие маски означает, что не будет происходить сравнение старших и/или младших значений адресов полученного пакета и адреса ячейки, а окончательное решение будет основываться на сравнении частей адресов, не «закрытых» маской. Каждая ячейка может содержать любой идентификатор адреса (ретрансляция или приём). Ячейки необходимо распределять между идентификаторами, исходя из реальной необходимости, и по возможности назначать адресацию в радиосети таким образом, чтобы была возможность введения не индивидуальных адресов, а групповых или широковещательных. Идентификатор каждой ячейки автоматически присваивается при выполнении команд **\$RID** и **\$PID**.

Каждая ячейка может содержать специальную маску для анализа РП пакета. При программировании маски предусмотрены две логические операции: «ИЛИ» (символ «|») и «И» (символ «&»). Если маской необходимо выделить один или несколько ретрансляторов, указанных в РП пакета, то используется операция «ИЛИ», если группу – операция «И». Если пакет от ретранслятора (группы ретрансляторов) должен быть обработан, необходимо в команде программирования маски указать знак обработки «+», иначе — «-».

При программировании адресов \$RIDxx вводятся следующие параметры:

- признак адреса повторяемого пакета (адрес отправителя/адрес получателя);
- признак разрешения на повтор пакета с РП, равным «0»;
- специальная маска и логическая операция, которая определяет действие между запрограммированной маской и полученным РП пакета.



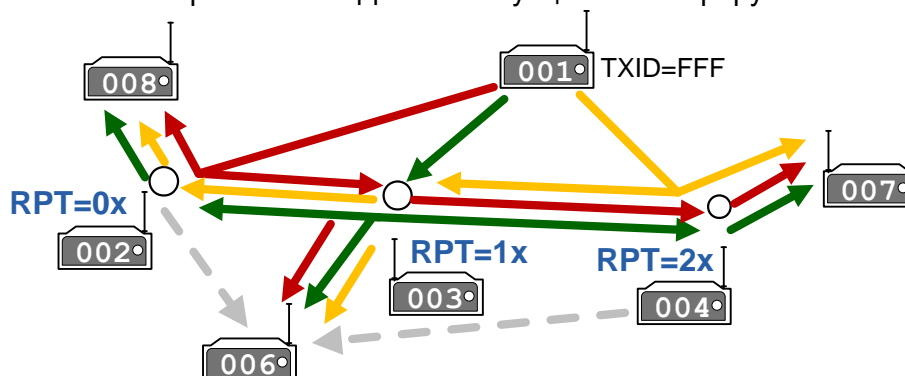
Программировать анализ маски РП следует только в том случае, если между конечным абонентом и цепочкой ретрансляторов существует «прямая видимость» в радиоэфире, вследствие чего может увеличиваться трафик служебных пакетов из-за множественной ретрансляции.



Необходимо иметь в виду, что активация анализа маски снижает надёжность сети, т.к. выход из строя ретрансляционного узла может привести к неработоспособности остального тракта передачи данных.

Рассмотрим пример необходимости анализа РП пакета на предмет повторения, учитывая специальную маску.

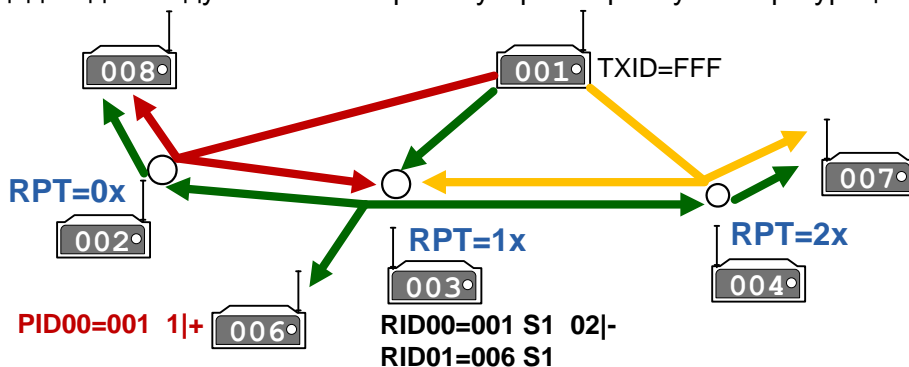
В случае если ретрансляторы находятся в прямой видимости друг от друга, могут иметь место лишние повторы пакетов. Данная ситуация иллюстрируется на следующем рисунке:



Модули 2, 3 и 4 «слышат» друг друга и работают в режиме ретрансляторов пакетов для модулей 8, 6 и 7 от базового 1.

Рассмотрим прохождение широковещательного пакета от модуля 1 до модуля 6. В данной ситуации модуль 3 повторит прямой пакет от модуля 1 и пакеты, ретранслированные модулями 2, 4. Следовательно, в направлении модуля 6 будет 3 пакета от модуля 1. Также, на модуль 6 возможно прохождение (пунктирная линия) прямых пакетов от ретрансляторов 0 и 2. Эфир будет забит лишними пакетами и скорее всего, такая система работать не будет.

Зададим для модуля 6 и 3 теперь такую расширенную конфигурацию:



Для модуля 3 конфигурация расшифровывается следующим образом :

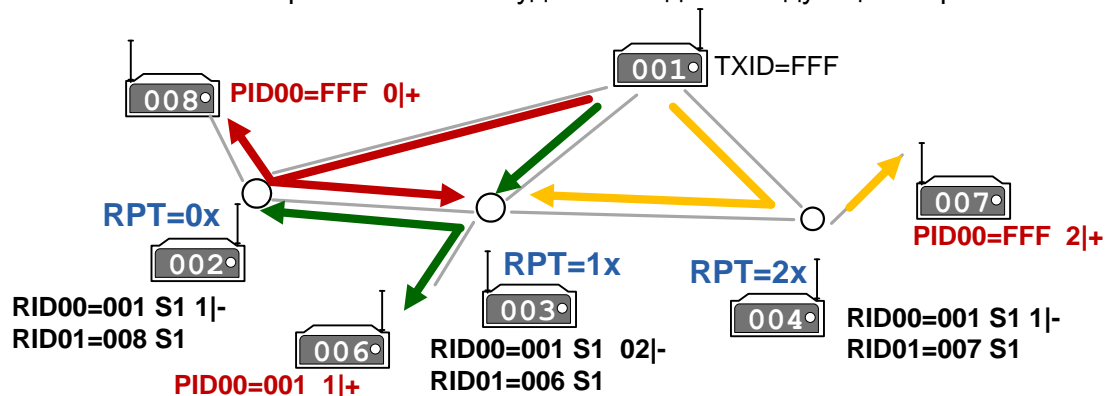
**RID00=001 S1 02|-** Не повторять пакеты от модуля 1, если они прошли через ретранслятор 0 ИЛИ 2

**RID01=006 S1** Повторять пакеты от модуля 6

Для модуля 6 конфигурация расшифровывается следующим образом:

**PID00=001 1|+** Принимать пакеты от модуля 1 только, если он прошел через ретранслятор 1

Окончательная настройка системы будет выглядеть следующим образом:



Если маска активирована и запрограммирована *на ретрансляцию* пакетов и совпадает с РП принятого пакета, подлежащего ретрансляции, окончательное решение о ретрансляции данного пакета принимается после анализа адресов RIDxx.

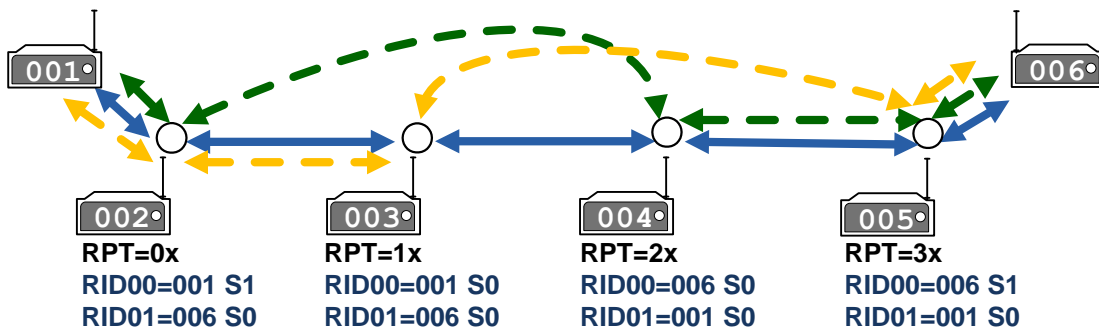
Если маска активирована и запрограммирована *на не ретрансляцию* пакетов и совпадает с РП принятого пакета, подлежащего ретрансляции, анализ адресов RIDx не происходит и пакет, подлежащий ретрансляции, не ретранслируется.

Примеры программирования масок:

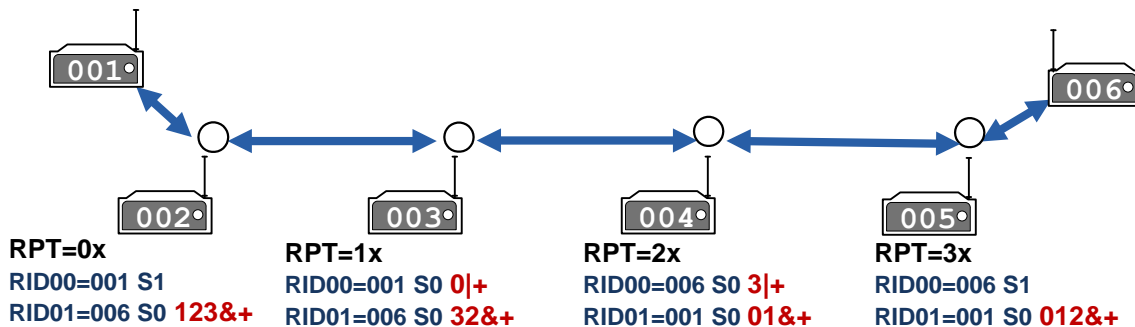
Команда	Значение
<b>RID01=001 S1 57&amp;+</b>	Ретранслировать пакеты от модуля 1, только если данный пакет уже ретранслирован ретрансляторами 5 И 7. В любом другом случае данный пакет не ретранслируется.
<b>RID07=001 D1 57&amp;-</b>	Не ретранслировать пакеты, предназначенные модулю 1, которые уже ретранслированы ретрансляторами 5 И 7. В любом другом случае данный

- RID01=001 S0 07|+** пакет ретранслируется. Ретранслировать пакеты от модуля 1, только если данный пакет уже был ретранслирован ретрансляторами 0 ИЛИ 7. В любом другом случае данный пакет не ретранслируется.
- RID01=00F S1 07|-** Не ретранслировать пакеты от группы абонентов (001, 002...00E), которые уже были ретранслированы ретрансляторами 0 ИЛИ 7. В любом другом случае пакет от данной группы абонентов ретранслируется.

Еще один пример (повторители в линию) показан на рисунке ниже:



Необходимо обеспечить связь между модулями 1 и 6. Для этого понадобилось 4 ретранслятора. При планировании обнаружилось частичное прохождение радиосигнала между модулями 2-4 и 3-5. Если запрограммировать ретрансляторы стандартными средствами (без анализа масок), то в эфире будут лишние пакеты, которые могут привести к не работоспособности системы.



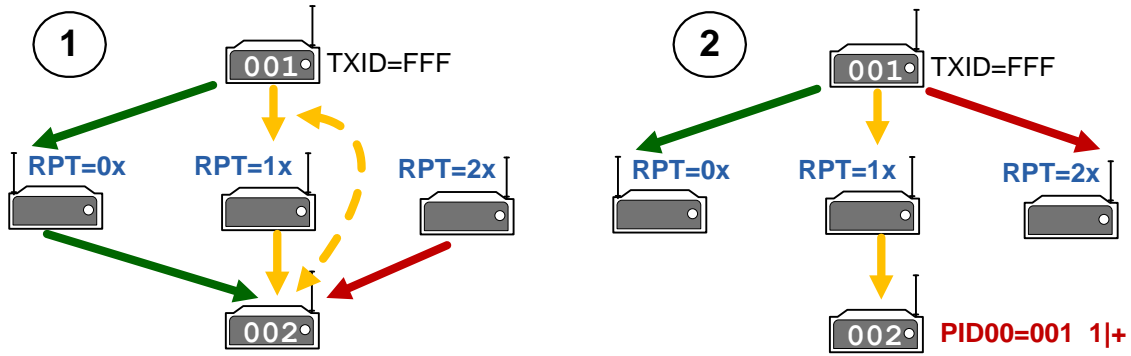
Введение соответствующих масок у ретрансляторов позволило строго определить путь прохождения пакетов между модулями 1 и 6 не обращая внимание если ли прямое прохождение сигнала между модулями ретрансляторами или нет .

### 8.1 ПРИЁМ РЕТРАНСЛИРОВАННЫХ ПАКЕТОВ

При прохождении пакетов между модулями через сеть повторителей (ретрансляторов) желательно исключить дублирование от соседних ретрансляторов некоторых служебных пакетов процесса обмена информацией, т.к. они требуют немедленной реакции, и поэтому возможно заполнение эфира лишними пакетами, что уменьшает общую пропускную способность канала.

Для исключения приёма не нужных пакетов имеется возможность установить маску на приём/игнорирование пакетов только от определенных ретрансляторов или группы ретрансляторов. Программирование приёма пакета от определенных ретрансляторов осуществляется командой **\$PIDxx**. Всего может быть запрограммировано до 8 значений PID. Синтаксис ввода и логика маски аналогична маске при анализе РП процесса ретрансляции пакетов.

Рассмотрим пример:



Допустим, что в данном примере неизвестно, как запрограммированы ретрансляторы 0,1 и 2. Но известно, что модуль 2 находится в радиовидимости всех 3-х. Также, есть частичное прямое прохождение пакетов между модулями 1 и 2. Необходимо, чтобы пакеты от модуля 1 воспринимались модулем 2 по **единственному** пути - прошедшие через ретранслятор 1.

После ввода модулю 2 значение **PID00=001 1|+**, модуль будет обрабатывать пакеты **только** прошедшие через ретранслятор 1. Все остальные будут игнорироваться.

Примеры программирования маски:

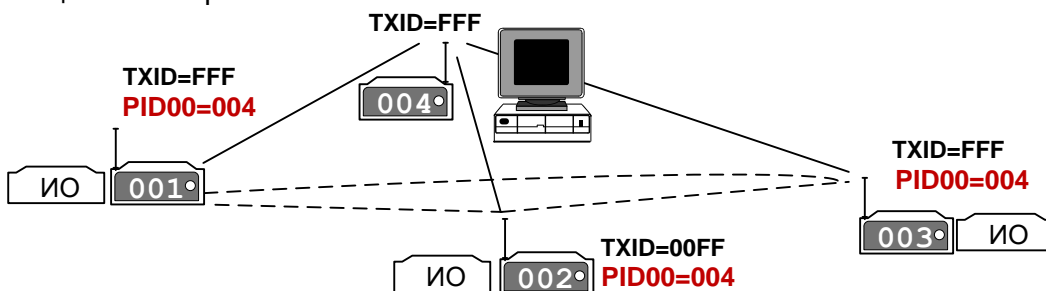
Команда	Значение
<b>\$PID00=001 57&amp;+</b>	Принимать пакеты от модуля 1, только если он был ретранслирован ретрансляторами 5 И 7. В любом другом случае данный пакет игнорируется.
<b>\$PID03=004 57&amp;-</b>	Не принимать пакеты от модуля 4, который был ретранслирован ретрансляторами 5 И 7. В любом другом случае данный пакет принимается.
<b>\$PID07=011 07 +</b>	Принимать пакеты от модуля 011, только если он уже был ретранслирован ретрансляторами 0 ИЛИ 7. В любом другом случае данный пакет игнорируется.
<b>\$PID03=00F 07 -</b>	Не принимать пакеты от группы 00, который был ретранслирован ретрансляторами 0 ИЛИ 7. В любом другом случае данный пакет принимается.

## 8.2 ИГНОРИРОВАНИЕ ПАКЕТОВ

В модуле реализована возможность игнорирования пакетов от определенных абонентов или приёма пакетов только от определенного абонента (группы абонентов). Данное свойство удобно использовать при организации радиосети в широкополосном режиме с одной или несколькими базовыми станциями, когда приём пакетов от абонентов, не являющихся базовой станцией, нежелателен. Адрес базовой станции задаётся командой **\$PIDxx**, в которой отсутствует маска анализа РП. Значение 0xF в адресе PID означает, что при анализе PID данная часть адреса не будет анализироваться, т.е. имеется возможность замаскировать целую группу абонентов.

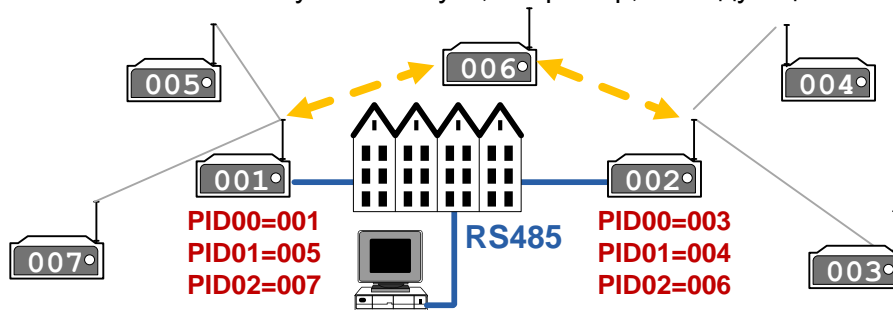
Программирование базовых станций также может потребоваться, в случае если несколько модулей, соединенных в сеть RS-485, работают на несколько базовых станций. В этом случае получение широкополосного пакета может вызвать коллизию в сети, хотя пакет может быть предназначен только одному абоненту, подключённому в сеть RS-485.

Рассмотрим пример конфигурации сети с одной базовой станцией, работающей в широкополосном режиме.



Базовая станция 4 передаёт запрос на модули 1, 2 и 3. Каждое ИО (исполнительное оборудование), получив свой запрос, передаёт ответ в модуль. Этот ответ также принимают все модули в сети и передают в свой порт. Если, например, для ИО модуля 3 необходимо не допустить получение информации от ИО других модулей, достаточно установить в модуле 3 значение PID, равное 0004. В этом случае на последовательный порт модуля 3 будет поступать информация только от базовой станции 4.

Коллизии в сети RS-485 могут возникнуть, например, в следующей системе:



Объекты расположены по разные стороны большого здания. При использовании одной базовой станции охватить всю сеть сбора информации не представляется возможным. Поэтому было принято решение установить два базовых модуля 1 и 2 по разные стороны здания. В данной ситуации выяснилось, что пакет от модуля 6 проходит как в модуль 1, так и в модуль 2. Оба модуля при этом одновременно выдают эти пакеты в сеть RS-485, что вызывает коллизию. Установка соответствующих значений PID на модулях 1 и 2 решает проблему. В данной конфигурации модуль 1 будет игнорировать данные от модуля 6. Информация от модуля 6 будет поступать в центральный пункт только через модуль 2.



Если модулем получен пакет, содержащий информацию для удалённой конфигурации, он обрабатывается независимо от того, находится ли адрес отправителя пакета в списке активированных адресов виртуальных базовых станций или нет.

## 9 ФИЛЬТРАЦИЯ ДАННЫХ

В модуле реализованы простые фильтры пакетов на предположении о том, что в популярных телеметрических протоколах первый байтом идет адрес устройства. В фильтр можно включить диапазон разрешенных значений от 0x00 до 0xFF - если значение задано (разрешено) пакет проходит через фильтр (передается в эфир, принимается из эфира или ретранслируется), иначе удаляется. Применение фильтров позволяет включить модем в общую сеть RS485 и не передавать в эфир весь трафик (выдавать на порт) или ретранслировать только конкретные пакеты. Конечно, для корректной фильтрации необходимо знать тип данных.

*По умолчанию, все фильтры отключены.*

Фильтры задаются в командном режиме следующими командами:

**\$xFL=cccc**      Задаёт способ фильтрации cccc (способы фильтрации описаны далее);  
**\$xFLDMP**        Вывод адресов разрешенных для фильтрации;  
**\$xFL+aabb**      Включение диапазона адресов aa-bb для разрешения фильтрации;  
**\$xFL-aabb**      Исключение диапазона адресов aa-bb из разрешения для фильтрации;  
**\$FLS**            Запись адресов для фильтрации в ЭНОЗУ;

где, x - 'T' фильтр исходящих данных, 'R' - фильтр входящих данных, 'RT' - фильтр для ретрансляции.

Способ фильтрации (\$xFL) хранится в общем профиле модема и может быть изменен удаленно по эфиру. Сами же адреса в профиле не хранятся и не могут быть изменены удаленно. Соответственно, для записи способа фильтрации в ЭНОЗУ необходимо выполнить команду \$S (для адресов - \$FLS). Возможны следующие варианты фильтрации:

----	Фильтр отключен (значение по умолчанию).
RTUA	Фильтр настроен на пакеты протокола Modbus RTU. Если адрес присутствует в списке разрешенных значений, то пакет проходит через фильтр. <b>Иначе удаляется.</b> Также, пакет удаляется если не соответствует пакету протокола Modbus RTU. Соответствие основывается на корректности контрольной суммы пакета (CRC).
DL8A	Аналогичен RTUA, только CRC пакета рассчитывается по алгоритму DALLAS CRC8.
1BYT	Фильтр проверяется 1-ый байт пакета на разрешенные значения без привязки к какому-либо протоколу (разрешенные значения задаются в адресах для фильтрации).
NONE	Фильтр запрещает все пакеты.

В адресе пакета может быть закодировано направление передачи. К примеру, если в протоколе Modbus RTU адрес 0x04 присутствует в запросе и ответе, то в других запрос может содержать адрес 0x04, а ответ 0x84.

Для понимания процесса фильтрации на примере фильтра исходящих данных в эфир приведем следующую простую схему:



В существующую проводную систему необходимо включить удаленный и локальный модуль IO. Без фильтрации пакетов весь трафик между мастером и устройствами с адресами 0x10 и 0x11 передавался бы в эфир. После задания на модуле IO\_Slave с адресом 0x12 фильтр только на пакет 0x13, в эфире будут запросы только к модулю IO\_Slave с адресом 0x13. После этого останется только установить на мастере корректный тай-аут ожидания ответа с учетом задержки передачи данных в эфире.

Также, на модуле IO\_Slave 0x12 необходимо будет задать время \$EOT - на скорости 9600 значений 5-10мс будет достаточно. Даже если оно будет больше чем время между запросом и ответом, например, между мастером и Slave 0x10 и этот "склеенный" запрос-



ответ попадает в буфер модуля IO\_Slave, он отбросится по причине не совпадения контрольной суммы.

### 9.1 ФИЛЬТРАЦИЯ ИСХОДЯЩИХ ДАННЫХ

Фильтр входящих данных по последовательному порту (исходящих данных предназначенных для передачи в эфир) может принимать следующие значения:

----, RTUA, DL8A, 1BYT, NONE

При значении фильтра NONE, пакеты поступившие по последовательному порту не передаются в эфир вообще.

Следует отметить, что контрольная сумма по Modbus RTU/ DALLAS CRC8 у пакета в буфере подсчитывается всегда поэтому, если задан фильтр RTUA или DL8A, то не обязательно задавать соответствующий тип протокола входящих данных (команда \$DAT).

### 9.2 ФИЛЬТРАЦИЯ ВХОДЯЩИХ ДАННЫХ

Фильтр входящих данных из эфира может принимать следующие значения:

----, RTUA, DL8A, 1BYT, NONE

При значении фильтра NONE, данные полученные из эфира не выдаются на последовательный порт вообще.

### 9.3 ФИЛЬТРАЦИЯ ДАННЫХ ПРИ РЕТРАНСЛЯЦИИ

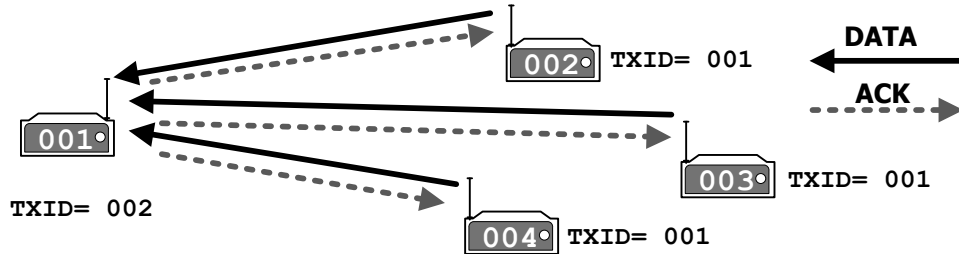
Фильтр ретранслируемых данных может принимать следующие значения:

----, RTUA, DL8A, 1BYT

# 10 ПРИМЕРЫ ОРГАНИЗАЦИИ СЕТЕЙ

## 10.1 РЕЖИМ «ТОЧКА-ТОЧКА»

### 10.1.1 РЕЖИМ ТОЧКА-ТОЧКА.

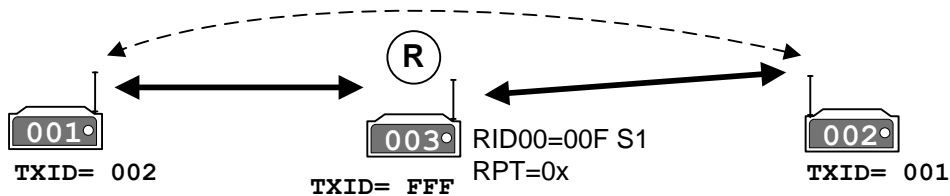


В данном режиме гарантируется доставка данных адресату. Отослав «индивидуальный» пакет, модуль ожидает подтверждения приёма со стороны получателя и повторяет пакет при неполучении подтверждения по истечении тайм-аута  $\$ACKT$ . В данном режиме принимающий модуль способен подтверждать полученный пакет как коротким кадром (ACK), так и своим информационным пакетом (если таковой имеется).

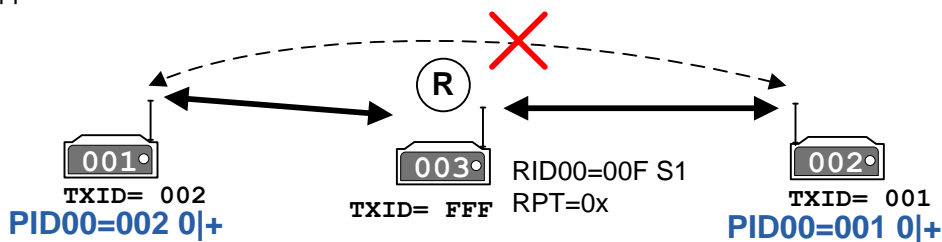
Следует заметить, что в данном режиме модуль может передавать свои данные только одному адресату, но поддерживать несколько виртуальных соединений в режиме «точка-точка» (отсылать короткие кадры подтверждения).

### 10.1.2 РЕЖИМ ТОЧКА-ТОЧКА С ПОВТОРИТЕЛЕМ.

Режим аналогичен режиму «точка-точка» без повторителя. Режим повторителя у модуля 003 активизируется установкой параметра  $\$RPT$  и заданием фильтра  $\$RID$ . Такая конфигурация не исключает приёма модулями 001 и 002 «прямых» пакетов друг от друга, что может привести к избыточным пакетам в эфире:



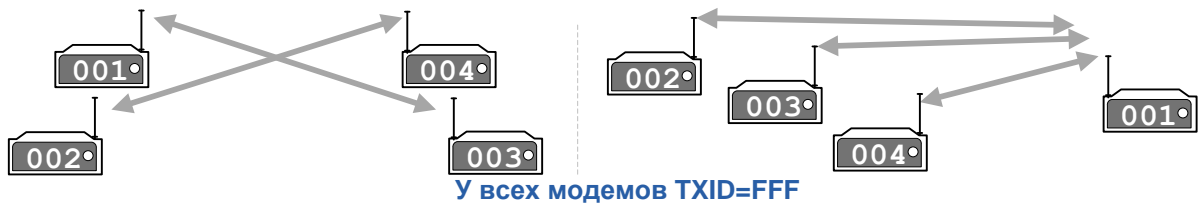
Поэтому, при вводе повторителя в систему крайне рекомендуется устанавливать маски расширенного приёма всем модулям создавая таким образом "виртуальный" путь прохождения пакетов:



В этом случае, модуль 001 будет реагировать на пакеты от модуля 002 только, если он прошел через повторитель и будет игнорировать прямые пакеты. Тоже самое верно и для модуля 002.

## 10.2 РЕЖИМ «ШИРОКОВЕЩАТЕЛЬНЫЙ»

### 10.2.1 ШИРОКОВЕЩАТЕЛЬНАЯ СЕТЬ БЕЗ ГАРАНТИИ ДОСТАВКИ СООБЩЕНИЙ

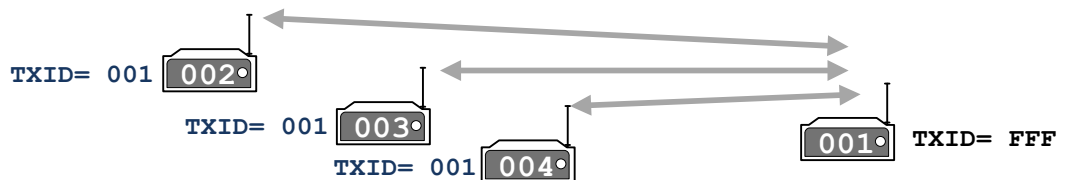


Модули 001, 002, 003 и 004 являются равноправными членами сети и могут принимать пакеты друг от друга.



Несмотря на то, что при данном адресном режиме нет гарантии доставки сообщений, он является предпочтительным почти во всех случаях, т.к. механизм доставки сообщений реализуется на более высоком уровне - на уровне протоколов оборудования, подключенных к модулям. При "нормальной" связи, лишние пакеты подтверждений от самих модулей могут "загрузить" радиозэфир, тем самым снизить пропускную способность канала. Данная адресация установлена **ПО УМОЛЧАНИЮ**

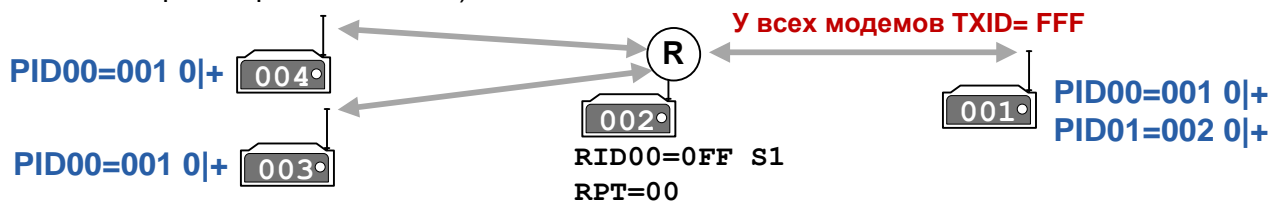
### 10.2.2 ШИРОКОВЕЩАТЕЛЬНАЯ СЕТЬ С ГАРАНТИЕЙ ДОСТАВКИ СООБЩЕНИЙ



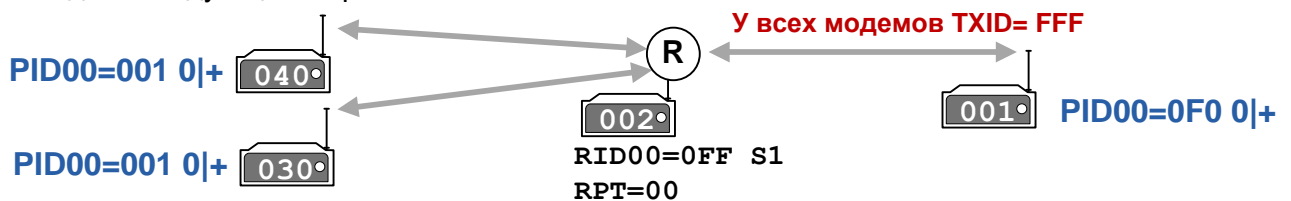
Конфигурация аналогична предыдущему варианту, за исключением того, что модули 002, 003 и 004 входят в адресный режим с базовой станцией при передаче данных (с гарантией доставки).

### 10.2.3 ШИРОКОВЕЩАТЕЛЬНАЯ СЕТЬ С ПОВТОРИТЕЛЕМ

Широковещательная сеть с повторителем показана на следующем рисунке (для модулей 001, 004 и 003 крайне желательно установить маски расширенного приема, чтобы исключить прием прямых пакетов):



Как видно из рисунка, с введением дополнительных модулей усложняется конфигурация масок (для модуля 001). Однако, с изменением адресов MYID модулей 003 и 004 можно воспользоваться заданием групповой маски. Предыдущий вариант будет выглядеть следующим образом:



# 11 ШИФРОВАНИЕ ДАННЫХ

В модуле реализовано 2 метода шифрования - AES128 с 16 битным ключом и простое XOR шифрование. Методом AES шифруются только пользовательские данные. Методом XOR весь пакет целиком.



Ключи AES и XOR нельзя узнать. Их можно либо удалить, либо изменить  
По умолчанию, ключи AES и XOR **НЕ ЗАДАНЫ**.

Признаком наличия ключей является метка **\*KEY** (для ключа AES)  
или метка **\*SK** (для ключа XOR) в результатах вывода команды \$DMP.

## 11.1 ШИФРОВАНИЕ AES

Шифрование AES осуществляется с помощью 16 битного ключа. Ключ задается в командном режиме с помощью команд \$KEA (в символах) или \$KEN (в 16-ричном формате).

Данные шифруются блоками по 16 байт. В конце блоков добавляется 32 битная контрольная сумма. После этого, зашифрованный блок передается на кодирование для передачи в эфир (добавляется заголовок, FEC, разбивка на блоки и т.д).

У каждого модуля должен быть одинаковый ключ. В случае, если после дешифрования не совпадает 32 битная контрольная сумма, данные в порт не выдаются.

Ввиду того, что для механизма AES данные должны быть кратны 16 и требуется дополнительная контрольная сумма, максимальный размер пользовательских данных передаваемых в эфир в одном пакете уменьшается. Далее приведена таблица максимального размера данных для различных комбинаций FEC и AES

Код FEC	Максимальный размер данных без AES, байт	Максимальный размер данных с AES, байт
RS(7,3)	101	92
RS(15,9)	139	124
HAM(12,8)	158	140
RS(7,5)	168	156
RS(15,11)	170	156
НЕТ	234	220

## 11.2 ШИФРОВАНИЕ XOR

Шифрование XOR осуществляется с помощью 6 байтного ключа методом операции XOR. Ключ задается в командном режиме с помощью команды \$SK (Simple Key). Данным методом шифруется весь пакет целиком в эфире, включая заголовок. Если на удаленном модуле ключ SK будет задан неверно, то пакет будет дешифрован как "мусор" и не будет декодирован. Данное свойство будет полезно, когда в пределах радиовидимости текущей сети появляется аналогичная другая. Это может исключить некорректное поведение, например, встроенного IO модуля модуля.

Ключ SK не обладает какой либо криптостойкостью, но будет полезен для минимальной защиты. Также, он не изменяет максимальную длину информационного пакета в отличии от AES.

## 12 Модуль ввода вывода (IO)

### 12.1 ОБЩИЕ СВЕДЕНИЯ

Состояние входов/выходов модуля может быть прочитано/установлено по радиоканалу или по активному интерфейсу командами Modbus ASCII/RTU (распознаются автоматически). Форматы сообщений и карта регистров приведена в разделе "[Карта регистров Modbus](#)".

Для того чтобы модуль корректно воспринимал Modbus команды по радиоканалу, необходимо, чтобы команда умещалась в 1 пакет, передаваемый в эфир. Например, функция **F01** в протоколе Modbus RTU имеет длину 8 байт; если модуль получит данную команду частями (в виде двух пакетов длиной, скажем, 3 и 5 байт), она будет проигнорирована. Данный факт необходимо учитывать, если в качестве устройства Master выступает «связка» OPC сервера и модуля в качестве радиоудлинителя. В этом случае у модуля необходимо настроить параметр **\$EOT** в сторону увеличения (на формирование modbus запросов самим модулем, параметр EOT не влияет).

Модуль ввода/вывода может работать в 3-х режимах:

**IO\_Slave** – пассивный режим, модуль принимает по радио или активному порту и выполняет запросы и команды в протоколе Modbus. **Включен по умолчанию.**

**IO\_Master** – модуль сам формирует в цикле запросы по радио или активному порту команды в протоколе Modbus для опроса и управления удаленными модулями, работающими в режиме IO\_Slave, благодаря чему обеспечивается трансляция состояния входов удаленных модулей на выходы мастера и наоборот – входов мастера на выходы удаленных модулей.

**IO\_Trans** – (прозрачное транслирование) пассивный режим, модуль принимает по радио или активному порту и выполняет запросы и команды в протоколе Modbus. Аналогичен режиму IO\_Slave. В дополнении к нему, выходы модуля устанавливаются значениями, которые передаются во всех информационных пакетах.



Режим IO\_Master включается командой **\$IOMASTER**  
 Режим IO\_Slave включается командой **\$IOSLAVE**  
 Режим IO\_Trans включается командой **\$IOTRANS**

### 12.2 Входы IN1...IN4

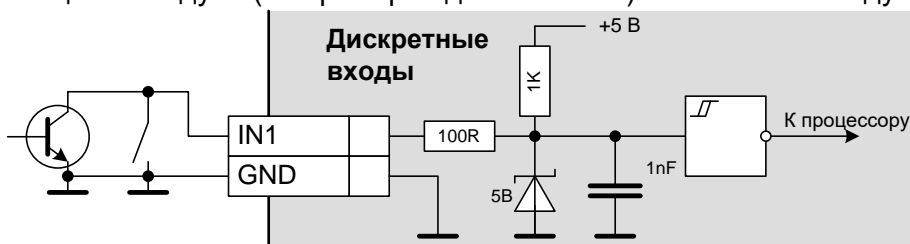
К четырем парам клемм IN1...IN4 подключаются внешние дискретные датчики типа «сухой контакт» (выключатели, кнопки, концевые выключатели, контакты реле или датчики различного типа с транзисторным выходом) или транзисторные ключи n-p-n типа с открытым коллектором.

Каждый вход имеет светодиодный индикатор состояния, расположенный у соответствующей пары клемм. Индикатор загорается при замыкании клемм.



Входы модуля не имеют гальванической развязки ни между собой, ни с внутренними цепями (контакты GND всех входов соединены внутри модуля). Внутреннее сопротивление контактов подключаемых к входам не должно превышать 100 Ом.

Схема подключения внешних цепей к входам IN1...IN4, а также внутренняя схема входных цепей модуля (на примере одного канала) показана на следующем рисунке.



### 12.3 Выходы OUT1...OUT4

К четырем группам клемм OUT1...OUT4 подключаются внешние исполнительные устройства.

Выходы OUT1...OUT4 представляют собой переключающие электромагнитные реле, позволяющие коммутировать нагрузку с максимально допустимым током 4 А при напряжении 250 В 50 Гц или 4 А при постоянном напряжении 24В. На клеммы выведены сухие контакты реле со следующими обозначениями:

- **A** – нормально замкнутый;
- **B** – перекидной;
- **C** – нормально разомкнутый.

Каждый выход OUT1...OUT4 имеет собственный светодиодный индикатор состояния, расположенный непосредственно у соответствующей группы клемм. Индикатор загорается при срабатывании реле (замыкании клемм B и C и размыкание клемм B и A).

Выходы реле модуля гальванически развязаны между собой и внутренними цепями.

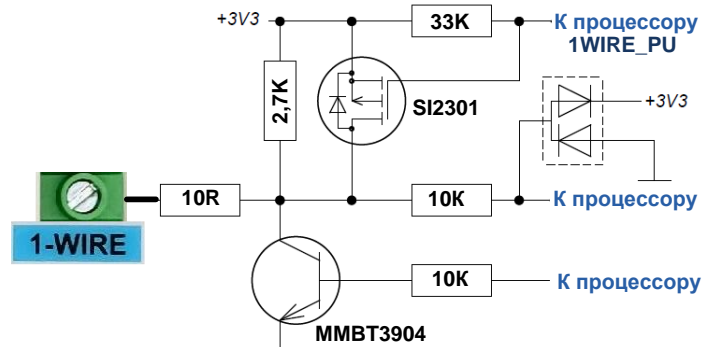
### 12.4 ТЕСТ ВХОДОВ/ВЫХОДОВ

Работоспособность входов/выходов может быть проверена по команде BOOT\_DEF загрузчика. При переходе в режим BOOT\_DEF модуль IO становится не активным. В режиме тестирования при замыкании соответствующего входа, замыкается соответствующий выход и индикатор ALARM.

### 12.5 КЛЕММА 1WIRE

Схема выходного блока интерфейса 1WIRE показана на рисунке.

Интерфейс не имеет отдельного выхода питания. Внешние устройства должны работать либо по "паразитному" питанию линии 1WIRE, либо иметь внешнее.



Встроенной схемы достаточно для подключения, например, датчиков температуры без внешнего питания. В момент преобразования (измерение температуры) включается внутренний сигнал 1WIRE\_PU.

Логика работы интерфейса 1WIRE в составе модуля ввода-вывода на данный момент не определена. Доступна только тестовая команда \$IO1WDMP при вводе которой в командном режиме осуществляется поиск и отображение подключенных устройств.

```

OK> $IO1WDMP
OK> press 's' to stop...
__1_wire__:
temp dev: |-----| Список датчиков температуры
1) DS18S20 t=+24 02766FC9
2) DS18S20 t=+24 02768905
3) DS18S20 t=+24 0276629B
4) DS18S20 t=+24 02767E07
5) DS18B20 t=+24 043BC589
6) DS18B20 t=+24 043BB599 |-----| Серийный код датчика
7)
8) Тип датчика Температура
butn dev: |-----| Список ключей iButton
1) 0EF3548B
2)
3) |-----| Серийный код ключа
4)
    
```

Модуль поддерживает до 8-ми датчиков температуры (DS18B20 или DS18S20) и до 4-х ключей iButton.

**Реализация логики работы интерфейса 1WIRE возможна по ТЗ заказчика.**

## 12.6 АДРЕСАЦИЯ МОДУЛЕЙ

Для корректной работы модуля необходимо назначить ему адрес в системе Modbus. Адрес задается двумя командами \$IOADR и \$IONDX. Команда \$IOADR задает базовый адрес, а команда \$IONDX смещение от базы. Итоговый адрес модуля будет равен \$IOADR+\$IONDX. Если сумма адресов превышает значение 255 (0xff), то итоговый адрес высчитывается по модулю 256. К примеру, если IOADR=F0h, а IONDX=31h, то итоговый адрес будет равен 21h. Также, итоговый адрес не должен быть равен 0.

Модуль в режиме IO\_Master осуществляет опрос модулей IO\_Slave по адресам: \$IOADR+0, \$IOADR+1, \$IOADR+2..... \$IOADR+7. Соответственно, числа 0,1,2...7 называются индексом модулей IO\_Slave.

Модуль IO\_Master может иметь любой базовый адрес + смещение. В некоторых случаях даже совпадать с итоговым адресом одного из IO\_Slave (если в системе нет никаких других устройств мастеров ведущих опрос модулей).

Адреса модулей IO\_Slave могут иметь разную комбинацию \$IOADR+\$IONDX. Главное, чтобы итоговый адрес был уникальным и попадал в диапазон опроса IO\_Master (IOADR+0.....IOADR+7).



Далее, для упрощения чтения всегда будет предполагаться, что модули IO\_Slave и IO\_Master по умолчанию имеют **одинаковый** адрес IOADR=0A, а адресация модулей IO\_Slave будет оперироваться в **индексах**. К примеру, адрес модуля IO\_Slave с адресом IOADR+IONDX=0A+00=0Ah имеет индекс **0 (n0)**, с адресом IOADR+IONDX=0A+01=0Bh имеет индекс **1 (n1)** и т.д.



Модуль в режиме IO\_Master обеспечивает индикацию отсутствия связи только с модулями IO\_Slave имеющих индекс 0, 1, 2 или 3. Поэтому, если требуется индикация отсутствия связи, то необходимо задавать адрес модулю IO\_Slave так, чтобы он попал в диапазон индексов 0...3 со стороны модуля IO\_Master.

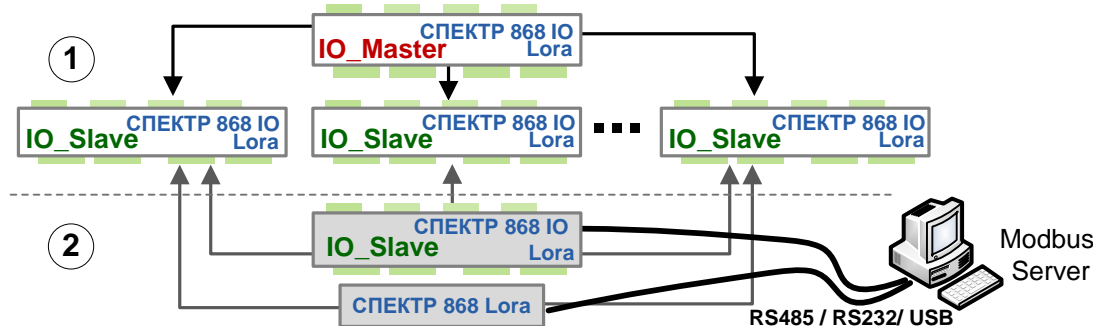
## 12.7 РЕЖИМ IO\_SLAVE

В этом режиме (активируется командой \$IOSLAVE) модули являются пассивными – они прослушивают радиоэфир или активный порт и выполняют приходящие по нему команды в протоколе Modbus.

Если в системе используется только модули IO, то общее количество устройств IO\_SLAVE должно быть не больше 8. Каждому модулю должен быть присвоен уникальный адрес (базовый + индекс). Больше настраивать модуль IO\_SLAVE кроме как радиочасти и функций входов/выходов для корректного опроса его модулем IO\_MASTER **НЕ НУЖНО**.

На рис.1 ниже показана схема работы системы из одного «мастера» и трех удаленных модулей «Спектр 868 IO Lora» в режиме IO\_Slave. При этом в качестве «мастера» используется модуль «Спектр 868 IO Lora» в режиме IO\_Master.

На рис.2 в качестве «мастера» может быть использован Modbus сервер подключенный по активному порту либо к модулю «Спектр 868 IO Lora» в режиме IO\_Slave, либо к радиомодему «Спектр 868 Lora». В таком варианте, число опрашиваемых модулей «Спектр 868 IO Lora» в режиме IO\_Slave может быть более 8 (вся емкость modbus адресов).



## 12.8 РЕЖИМ IO\_MASTER

В режиме IO\_Master модуль самостоятельно в цикле опрашивает функцией F02/F23 состояние входных контактов включенных в опрос удаленных модулей IO\_Slave, получает ответы от них и транслирует это состояние на свои выходы или другие модули IO\_Slave. Помимо этого, модуль может транслировать состояние своих входных контактов на выходы удаленных модулей IO\_Slave функцией F15/F23.

Для настройки опроса у модуля IO\_Master необходимо задать следующие параметры:

<b>\$IOF15</b>	Набор индексов модулей IO_Slave которым будет посылаться запрос на установку <b>ВЫХОДОВ</b> функцией 0Fh. Например, значение 310 означает, что IO_Master будет генерировать запросы функцией F15 с адресами IOADR+0, IOADR+1 и IOADR+3;
<b>\$IOF15BC</b>	Флаг. Установка в "1" означает, что команда на установку выходов, будет содержать широковещательный адрес (сразу всем). В этом случае, модуль(модули) IO_Slave не будут формировать ответ;
<b>\$IOF02</b>	Набор индексов модулей IO_Slave которым будет посылаться запрос на чтение <b>ВХОДОВ</b> функцией 02h. Смысл аналогичен набору параметра \$IOF15;
<b>\$IOF23</b>	Флаг. Установка в "1" означает подмену функции F02 функцией F23 т.е. вместо запроса на чтение входов, будет посылаться запрос сразу не чтение входов и установку выходов.



Если не нужно каких то специфических требований, оптимально задать только IOF02 и установить флаг IOF23 в 1 (IOF15 задавать не нужно). В этом случае, при опросе модуля IO\_Slave будет всего 2 пакета.

<b>\$IOQPT</b>	Путь опроса (QueryPath). Задаёт путь, через который IO_Master будет опрашивать устройство IO_Slave. Задаётся битовой маской. Каждый номер бита соответствует индексу модуля. Если бит равен '1', то опрос осуществляется по радио. Если бит равен '0' - по активному порту. <b>По умолчанию, опрос всех модулей ведётся по радио.</b>
<b>\$IOPT</b>	Пауза между опросами (Pause Time). Задаёт паузу в секундах после опроса очередного модуля IO_Slave.
<b>\$IOSMx</b>	Маска на установку выходов. Задаёт маскирование входов IO_Slave при выводе их на выходы IO_Master. Если не задать маскирование, то при опросе модуля IO_Slave выходы IO_Master будут постоянно менять свое значение в зависимости от значений входов очередного IO_Slave. Маска задается для каждого модуля IO_Slave (с индексами 0...7) отдельно.
<b>\$IOMXO</b>	Мультиплексор выходов для модулей IO_Slave. По умолчанию, на выходы всех модулей IO_Slave участвующих в опросе, транслируются входы IO_Master. С помощью мультиплексора можно задать на какой выход любого устройства (кроме IO_Master) какой будет транслироваться вход с любого другого устройства. Например, для модуля IO_Slave n5, можно указать, что на его выход OUT4 будет транслироваться вход IN2 с модуля IO_Slave n0, на выход OUT3 будет транслироваться вход IN4 с модуля IO_Slave n3 и т.д.

Рассмотрим принцип работы модуля IO\_Master на конкретном примере. Имеем 3 удаленных объекта и пункт управления. Необходимо обеспечить управление выходами на 2-х удаленных объектах по входам на пункте управления, а также трансляцию входов всех трех удаленных объектов на выходы в пункте управления.

Для решения понадобится один модуль в режиме IO\_Master (в пункте управления) и 3 модуля в режиме IO\_Slave (на удаленных объектах).

Первым делом необходимо присвоить каждому модулю IO\_Slave уникальные адреса. В примере они будут соответствовать индексам 0, 1 и 2. Теперь программируем IO\_Master:

<b>\$IOF23 = 1</b>	Включена функция F23 (чтение-установка) за 1 запрос
<b>\$IOPT = 1</b>	Пауза после опроса очередного модуля 1с
<b>\$IOQPT = 11111111</b>	Опрос по радио
<b>\$IOF15 =</b>	Функции F15 не нужна (используется "оптимизированная" функция F23)
<b>\$IOF02 = 210</b>	Задаем в соответствии с задачей применение функции "02 Read Input Status" (чтение состояние дискретных входов) для всех 3-х удаленных модулей с индексами 0,1 и 2.
<b>\$IOSM0 = 0001</b>	Задаем игнорирование состояний всех входов, кроме IN1, для удаленного
<b>\$IOSM1 = 0010</b>	модуля n0, всех входов, кроме IN2, для удаленного модуля n1 и всех



**\$IOSM2 = 0100** входов, кроме IN3, для удаленного модуля n2.  
 Теперь:  
 выход OUT1 мастера будет управляться входом IN1 модуля n0,  
 выход OUT2 мастера – входом IN2 удаленного модуля n1,  
 выход OUT3 мастера – входом IN3 удаленного модуля n2.

Схема работы модуля в режиме IO\_Master для данного примера показана на схеме:

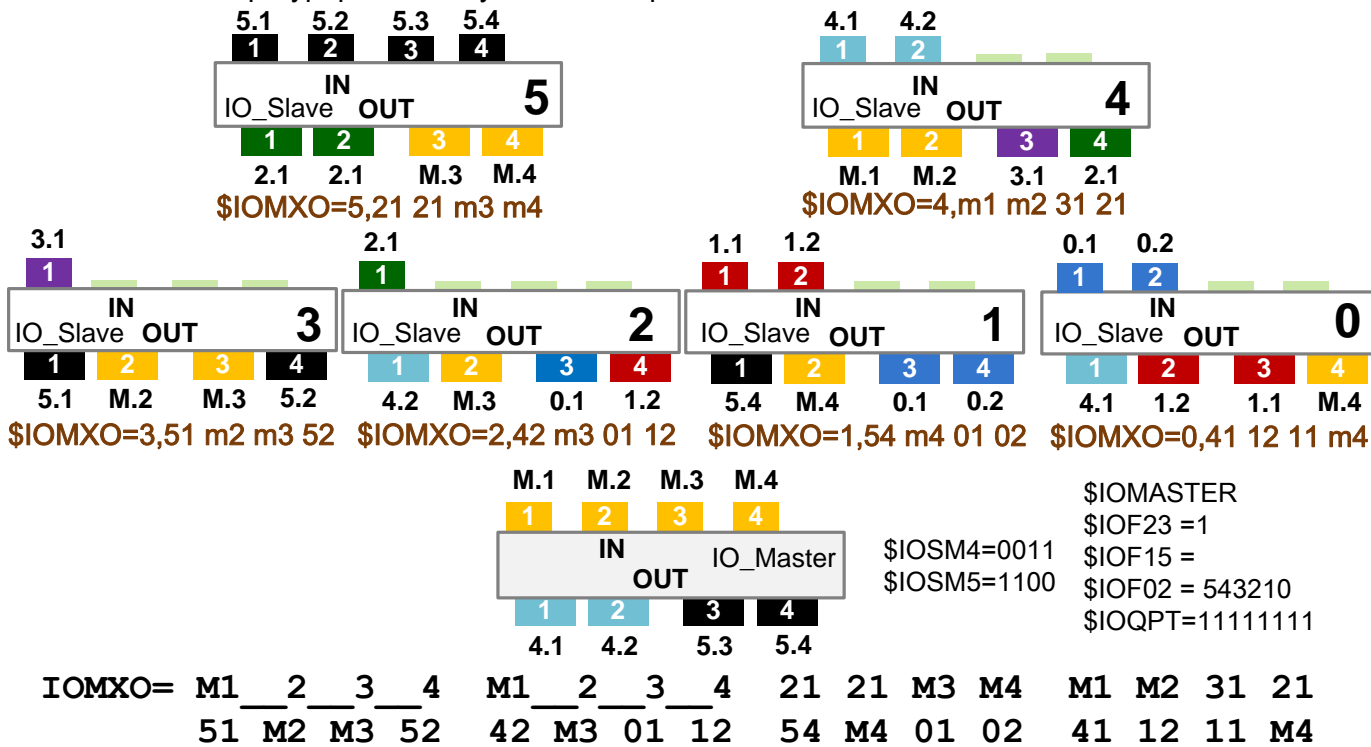


Таким образом, полный цикл работы модуля IO\_Master будет состоять из 3х запросов и 3х ответов. Если выключить флаг IOF23 и установить соответствующий параметр IOF15 для установки выходов, то цикл бы состоял уже из 12 пакетов (6 запросов и 6 ответов).

После передачи очередной команды модуль IO\_Master выдерживает паузу 1 секунду.

### 12.9 РЕЖИМ IO\_MASTER. МУЛЬТИПЛЕКСИРОВАНИЕ ВХОДОВ НА ВЫХОДЫ.

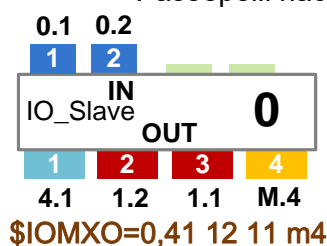
В режиме IO\_Master имеется возможность мультиплексировать входы одних модулей IO\_Slave на выходы других модулей IO\_Slave участвующих в опросе. Мультиплексирование выходов самого модуля IO\_Master недоступно. Далее приведена абстрактная схема, для понимания конфигурирования мультиплексора:



Алгоритм конфигурирования следующий:

- Составляется схема в соответствии с которой, необходимо "соединить" входы модуля X на выходы модуля Y.
- Выход маркируется как X.N, где X - индекс модуля IO\_Slave, N - номер входного контакта от 1 до 4.
- В модуле IO\_Master вводится начало команды "\$IOMXO=Y,".
- Далее, перечисляется конфигурация (без разделительных точек) начиная с первого выхода.
- Правильность ввода всегда можно проверить командой \$DMP - напротив параметра IOMXO идет перечисление мультиплекса для каждого IO\_Slave начиная с индекса 7, далее 6, 5, 4. На следующей строке для модулей 3-2-1-0.

Разберем настройку из рисунка выше для модуля IO\_Slave n0.



На OUT1 направить с IO\_Slave n4 вход IN1. Маркируем выход 4.1  
 На OUT2 направить с IO\_Slave n1 вход IN2. Маркируем выход 1.2  
 На OUT3 направить с IO\_Slave n1 вход IN1. Маркируем выход 1.1  
 На OUT4 направить с IO\_Master вход IN4. Маркируем выход M.4

Таким образом, конфигурация мультиплексора для модуля IO\_Slave n0 записывается как 41 12 11 m4. Итоговая команда для него будет \$IOMXO=0, 41 12 11 m4

Модуль при опросе входов устройств IO\_Slave всегда запоминает их состояние в ЭНОЗУ поэтому, при переключении ему питания всегда доступны последние значения входов ранее опрошенных модулей.

Если в момент работы, какой то модуль IO\_Slave становится не доступен в течении времени больше, чем параметр \$IOSFT, то в команде установки выходов (если в значениях есть входы этого недоступного модуля) IO\_Master посылает специальный признак сообщающий, что входы от недоступного модуля не актуальны. Модуль IO\_Slave получивший такой признак не обновляет соответствующий выход и включает таймер \$IOSFT по истечении которого, происходит "Авария" (если, конечно, такой режим у модуля IO\_Slave установлен см. раздел "[Индикаторы и реле ALARM](#)").

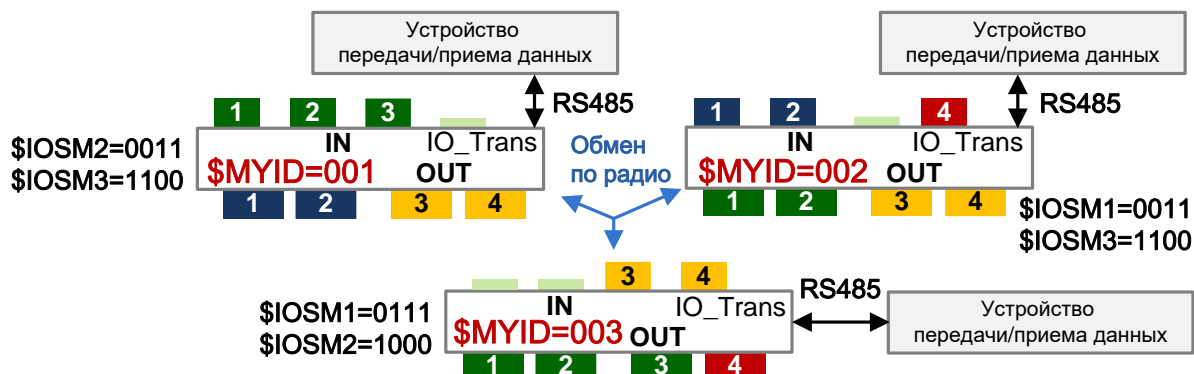
## 12.10 РЕЖИМ IO\_TRANS

Как говорилось выше, режим IO\_Trans (активируется командой \$IOTRANS) аналогичен режиму IO\_Slave. Важное отличие заключается в следующем: в заголовке каждого пакета в эфире передается состояния 4-х входов модуля ввода/вывода при получении которых, модуль в режиме IO\_Trans транслирует их на свои выходы. Чтобы, например, при получении широкополосных пакетов выходы не переключали свое состояние от разных модулей, осуществляется фильтрация адресов с помощью масок \$IOSMx.

Номер в маске \$IOSMx соответствует младшей цифре параметра \$MYID удаленного модуля. Таким образом, принимать значение входов можно только от модулей, у которых значения \$MYID равны от xx0 до xx7 (т.к. масок IOSMx всего 8). Если сравнивать принцип работы маски \$IOSMx в режиме IO\_Master, то в режиме IO\_Master маска задается для индекса модуля IO\_Slave, в то время как у модуля в режиме IO\_Trans маска задается по младшей цифре параметра \$MYID удаленного модуля.

Задавать modbus адрес в режиме IO\_Trans не обязательно (если не нужно управлять модулем modbus командами).

Схема работы в режиме IO\_Trans поясняется следующим рисунком:



Система состоит из 3-х модулей с адресами \$MYID 001,002 и 003. С помощью масок \$IOSMx задаем "маршрут" установки выходов:

MYID=001 \$IOSM2=0011 от модуля с MYID=002 выставить выходы OUT1 и OUT2;  
\$IOSM3=1100 от модуля с MYID=003 выставить выходы OUT3 и OUT4;

MYID=002 \$IOSM1=0011 от модуля с MYID=001 выставить выходы OUT1 и OUT2;  
\$IOSM3=1100 от модуля с MYID=003 выставить выходы OUT3 и OUT4;

MYID=003 \$IOSM1=0111 от модуля с MYID=001 выставить выходы OUT1,OUT2 и OUT3;  
\$IOSM2=1000 от модуля с MYID=002 выставить выход OUT4;

Направлять входы на выходы можно только с одинаковыми номерами - IN1 на OUT1, IN2 на OUT2 и т.д.



Обновление выходов в режиме \$IO\_Trans зависит от "активности" обмена данными в эфир между подключенными к модулям устройствами. Если нет обмена, то и выходы обновляться **не будут**.

## 12.11 БЕЗОПАСНОЕ СОСТОЯНИЕ ВЫХОДОВ

В процессе обмена между модулем IO\_Master и IO\_Slave может возникнуть ситуация, когда по каким-то причинам пропадает связь. В этом случае заданное ранее (когда связь еще была) состояние выходов может быть неактуальным или даже опасным для подключенных к ним исполнительных устройств. Например, может оказаться, что прошла команда на включение исполнительного устройства, а команда на выключение уже не может пройти из-за проблем со связью. Таким образом, исполнительное устройство останется включенным на неопределенное время, что может быть недопустимым.

Для предотвращения таких ситуаций, предусмотрена возможность задания "безопасного" состояния выходов. При отсутствии связи в течение заданного промежутка времени выходы модуля установятся в заранее заданное "безопасное" состояние. Для модуля в режиме IO\_Slave пропаданием связи считается отсутствие запросов, получаемых от мастера, а для модуля IO\_Master – отсутствие ответов от удаленных модулей (для каждого из удаленных модулей собственный счетчик времени).

С безопасным состоянием выходов связаны следующие регистры:

**\$IOALR** Режим индикации ALARM (IO ALaRm)

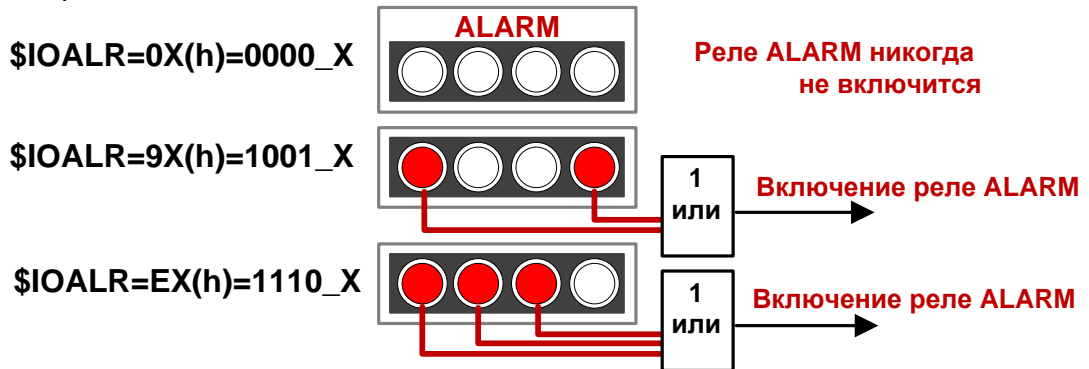
**\$IOSFT** Допустимое время отсутствия связи, в секундах (IO SaFe Time);

**\$IOSFV** Безопасные состояния для выходов (IO SaFe Value). Для модуля в режиме IO\_Slave задает маску выходов, для которых разрешено использование безопасного состояния. Для модуля в режиме IO\_Master задает маску индексов модулей IO\_Slave, для которых используется безопасное состояние

### 12.11.1 ИНДИКАТОРЫ И РЕЛЕ ALARM.

Модули имеют четыре индикатора и выход (реле) ALARM (авария), которые используются для отображения отсутствия связи. Под отсутствием связи подразумевается либо отсутствие в течение заданного времени запросов от мастера (в режиме IO\_Slave или IO\_Trans), либо отсутствие в течение заданного времени ответов от удаленных модулей (в режиме IO\_Master).

Реле ALARM включается в зависимости от набора включенных индикаторов ALARM и маски в регистре \$IOALR по логическому "ИЛИ". Маска во всех режимах задается в старших 4х битах регистра \$IOALR.



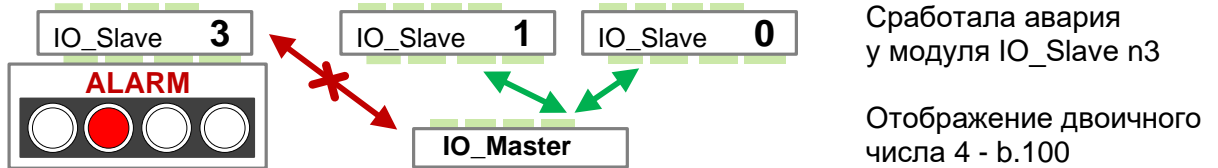
При включении питания, режим модуля кратковременно отображается на индикаторах ALARM (само реле ALARM при этом не переключается):

Режим	Отображение
\$IOSLAVE	индекс модуля плюс 1 двойным миганием в течении 0.5 с
\$IOTRANS	индекс модуля плюс 1 тройным миганием в течении ~0.9 с
\$IOMASTER	индексы опрашиваемых устройств IO_Slave одинарным миганием в течении 0.4 с (если индексы попадают в диапазон от 0 до 3).

**IO\_SLAVE.**

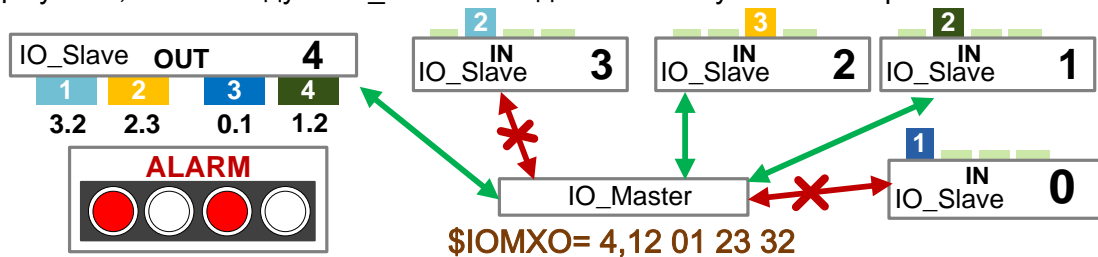
Режим аварии задается в младших 4-х битах регистра \$IOALR.

**\$IOALR[3:0] = 0.** В данном режиме, индикация аварии (по истечении времени IOSFT) включается, когда нет опроса от модуля "IO\_Master". На индикаторах ALARM выставляется двоичный индекс модуля IO\_Slave плюс 1



**\$IOALR[3:0] = 1.** Каждый из 4-х индикаторов указывает какой выход не обновлялся в течении времени IOSFT. Данный режим может быть включен, когда используется мультиплексирование выходов в модуле IO\_Master.

Допустим, что на модуль IO\_Slave n4 задано такое мультиплексирование:



Модуль IO\_Master транслирует значение выходов на IO\_Slave n4 таким образом:

- На OUT4 вход IN2 модуля IO\_Slave n1
- На OUT3 вход IN1 модуля IO\_Slave n0
- На OUT2 вход IN3 модуля IO\_Slave n2
- На OUT1 вход IN2 модуля IO\_Slave n3

Программировать IO\_Slave на отсутствие связи с модулем IO\_Master в данном случае будет не слишком информативно т.к. реальна ситуация, когда пропадет связь модуля IO\_Master с модулями 1,0,2,3. Если у IO\_Master пропадает связь с каким либо модулем IO\_Slave, в команде установки выходов F23 будет сброшен соответствующий бит-маска, которая будет означать, что модулю IO\_Slave n4 запрещено обновлять данный выход. По истечении времени IOSFT соответствующий индикатор ALARM включится. Таким образом, можно будет судить в какой именно ветке маршрутизации произошла авария (обрыв связи).

В данном примере отсутствует связь между модулем IO\_Master и модулями IO\_Slave с индексами 3 и 0. Это означает, что в течении времени \$IOSFT выходы 1 и 3 данного модуля не обновлялись.

Если пропадет связь с самим IO\_Master, то у модуля IO\_Slave в данном случае зажгутся все индикаторы ALARM.

#### IO\_TRANS.

Младшие 4 бита регистра \$IOALR не имеют значения.

#### IO\_MASTER.

При пропадании связи с любым из модулей IO\_Slave так же сработает реле на выходе ALARM и загорится один из индикаторов ALARM – в зависимости от того, с каким модулем пропала связь. При пропадании связи с модулем IO\_Slave n0 загорится первый индикатор, при пропадании связи с n1 загорится второй индикатор, с n2 – третий, с n3 – четвертый.

Если связь с модулем IO\_Slave при очередном опросе есть, соответствующий индикатор ALARM кратковременно загорается красным без воздействия на реле ALARM. Данное свойство можно использовать для визуального контроля наличия связи с удаленным модулем.

### 12.11.2 БЕЗОПАСНОЕ СОСТОЯНИЕ В РЕЖИМЕ IO\_SLAVE

За конфигурацию безопасного состояния в режиме IO\_Slave отвечают 3 параметра - \$IOSFT, \$IOSFV, и \$IOALR.

Параметр **\$IOSFT** задает таймаут на пропадание связи (авария) с мастером. Какое именно время устанавливать, зависит от длительности цикла опроса в системе и требований к ней. Например, если полный цикл опроса составляет 4 секунды, а таймаут задан равным 5 секундам, то пропадание *даже одного запроса* приведет к аварии. Если нежелательно, чтобы однократное не прохождение запроса считалось пропаданием связи, то можно поставить таймаут равным, например, 16 секунд – тогда пропаданием связи будет считаться только случай пропадания трех последовательных запросов.

С помощью регистра **\$IOSFV** нужно указать, для каких выходов модуля IO\_Slave нужно использовать безопасное состояние и задать собственно значение безопасного состояния (0 или 1) для выбранного выхода.

Установка запрограммированных выходов в безопасное состояние будет иметь разную логику в зависимости от регистра \$IOALR[3:0]:

- = 0** При наступлении безопасного состояния, на **BCEX** разрешенных выходах в регистре IOSFV установится безопасное значение.  
Например, при значении \$IOSFV = --10 в случае пропадания связи с IO\_Master (на время более чем IOSFT), реле на выходе OUT1 разомкнется (логический 0), реле на выходе OUT2 замкнется (логическая 1), а выходы OUT3 и OUT4 останутся без изменений (значение '-').
- = 1** При наступлении безопасного состояния, разрешенные выходы в регистре IOSFV установятся только те, которые не обновлялись в течении времени IOSFT.  
Например, при значении \$IOSFV = --10 выходы OUT1 или OUT2 установятся в безопасное состояние только в том случае, когда какой либо из них не будет обновляться модулем IO\_Master в течении времени больше, чем IOSFT

### 12.11.3 БЕЗОПАСНОЕ СОСТОЯНИЕ В РЕЖИМЕ IO\_MASTER

За конфигурацию безопасного состояния в режиме IO\_Master отвечают 2 параметра - \$IOSFT и \$IOSFV. Смысл этих параметров аналогичен в режиме IO\_Slave.

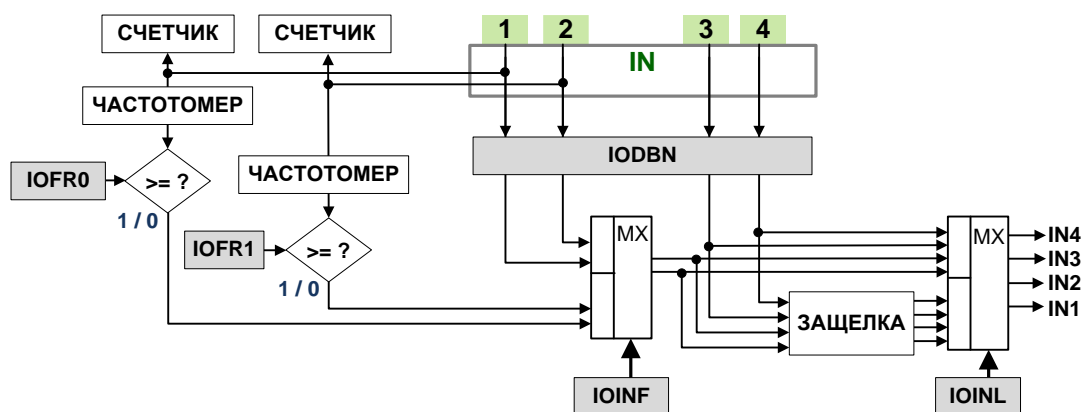
Разрешенные выходы в регистре \$IOSFV устанавливаются в безопасное состояние согласно индексу модуля IO\_Slave, с которым пропала связь. Например, если пропала связь с модулем n1, то в безопасное состояние перейдет только выход OUT2 (если он разрешен).



В случае применения функции безопасного состояния на выходах OUT4...OUT1 модуля IO\_Master, следует назначать удаленным модулям IO\_Slave адреса таким образом, чтобы они имели индекс от 0 до 3, т.к. у модуля отсутствуют выходы OUT8...OUT5.

## 12.12 ФУНКЦИИ ВХОДОВ

Структурная схема входов IN1-IN4 представлена на следующем рисунке:



Входы IN1 и IN2 имеют функции счета. Все входы проходят через функцию устранения дребезга контактов \$IODBN, поступают на мультиплексор детектора частоты \$IOINF (только IN1 и IN2) и далее на мультиплексор защелки \$IOINL (Latch). После мультиплексора защелки вход считается равным '1' или '0'.

По умолчанию, все мультиплексоры **отключены** и **подавление дребезга контактов не осуществляется** - входы "считываются" непосредственно с клемм.

### 12.12.1 "ДРЕБЕЗГ" КОНТАКТА.

Устранение дребезга контакта означает, что значение входа при опросе в течении заданного времени не изменилось. Счетчик времени опроса задается в 4-х байтном регистре \$IODBN. Старший байт задает счетчик для входа IN4, следующий для IN3, следующий для IN2 и младший байт для IN1. Счетчик может принимать значения от 0x00 до 0x7F. Значению 0x00 соответствует чтению контакта 1 раз (нет подавление дребезга). Опрос входных контактов осуществляется примерно раз в 25мс. Соответственно, максимальный счетчик времени может принимать значение равным ~3.2 с.

Например, значение \$IODBN = xx50xxxx означает, что в течении  $0x50=80 \cdot 25\text{мс} = 2\text{с}$  при опросе входа IN3 80 раз, его состояние не должно измениться.

### 12.12.2 ДЕТЕКТОР ЧАСТОТЫ.

Входы IN1 и IN2 имеют функцию счета импульсов входного сигнала. Счет осуществляется по возрастающему фронту. В не зависимости от конфигурации, подсчет импульсов и частоты происходит **всегда**. Частота считается как число импульсов в секунду и обновляется раз в секунду.

Счетчик импульсов на данный момент не участвует в логике работы модуля IO. Доступ к нему может быть осуществлен внешней Modbus командой чтения регистров (как и к значению частотомеров).

Частота сравнивается с компараторами IOFR0 и IOFR1 и, если её значение превышает величину записанную в регистрах \$IOFRx, на выходе компаратора устанавливается '1', иначе '0'. Далее, в зависимости от конфигурации мультиплексора \$IOINF значением соответствующего входа считается либо детектор частоты, либо прямое. Регистр \$IOINF содержит маску из 2-х бит. Установка бита маски в '1' назначает соответствующий вход детектором частоты.



Сумма значений частот на входах IN1 и IN2 не должна превышать 13 Кгц. В противном случае, функция счета останавливается на 10 секунд.

### 12.12.3 ЗАЩЕЛКА.

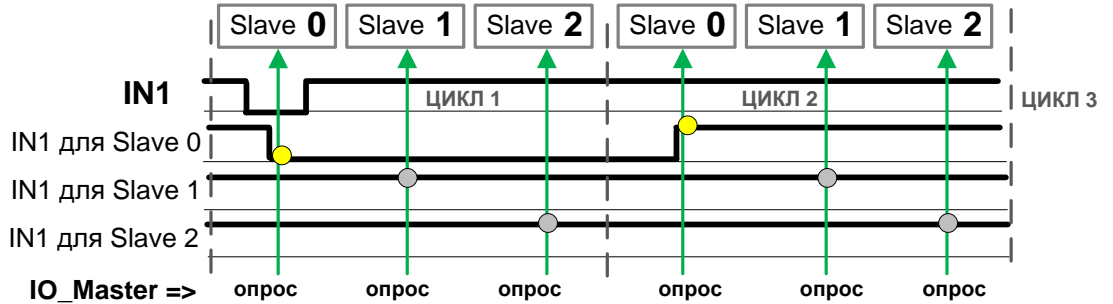
Под функцией "защелка" понимается способность запоминание перехода уровня на входе из "1" в "0" или из "0" в "1". Последующие изменения уровней не меняют значения входа до тех пор, пока текущее состояние не будет передано в режиме IO\_Master или не будет считано в режиме IO\_Slave.

В режиме IO\_Master модуль защелкивает состояние входа и запоминает это значение для каждого модуля IO\_Slave участвующего в опросе отдельно. Как только происходит опрос текущего модуля, значение защелки для него обновляется (если был факт смены). В режиме IO\_Slave, значение защелки входов обновляется после каждого опроса.

Режим защелки удобен в том случае, когда возможен кратковременный импульс на входе, который необходимо обработать, но времени он меньше полного цикла опроса всех устройств в режиме IO\_Master или возникает не в момент опроса в режиме IO\_Slave.

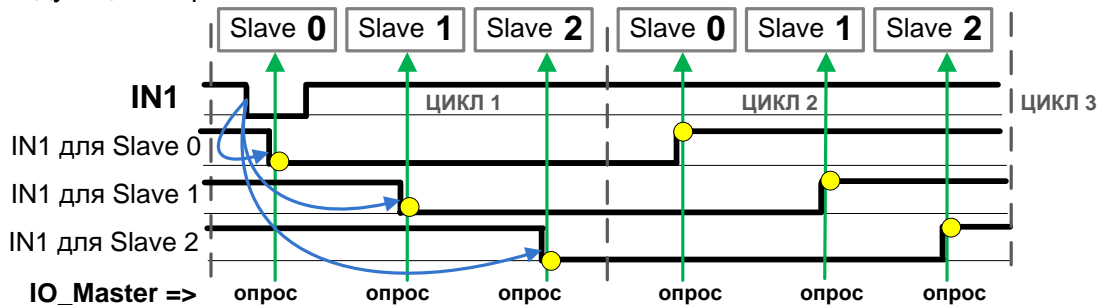
Проиллюстрируем работу IO\_Master в режиме защелки.

На следующей диаграмме режима защелки нет:



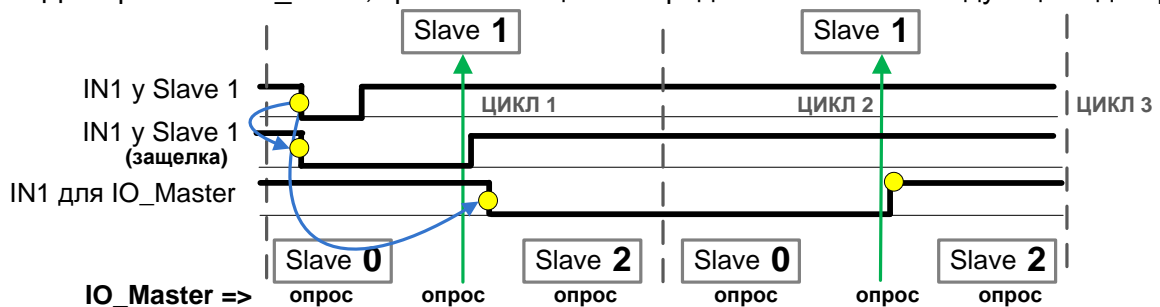
Короткий импульс '1' -> '0' возник в момент опроса IO\_Slave n0 и значение вернулось в '1' сразу же после опроса. IO\_Master "успел" передать текущее мгновенное состояние IN1 только одному IO\_Slave. Для остальных устройств, значение входа IN1 не изменилось, соответственно соответствующий OUTx у них также останется без изменений.

Теперь, активизируем на входе IN1 IO\_Master режим защелки в '0'. Диаграмма будет выглядеть следующим образом:



В данном случае, для каждого модуля IO\_Slave будет "доставлен" импульс со входа IN1. Как видно из рисунка, временная форма импульса не повторяется, передается лишь только факт изменения состояния.

Для режима IO\_Slave, работа защелки представлена на следующей диаграмме:



Принцип идентичен режиму IO\_Master. Защелка у модуля IO\_Slave сбрасывается после опроса модулем IO\_Master.

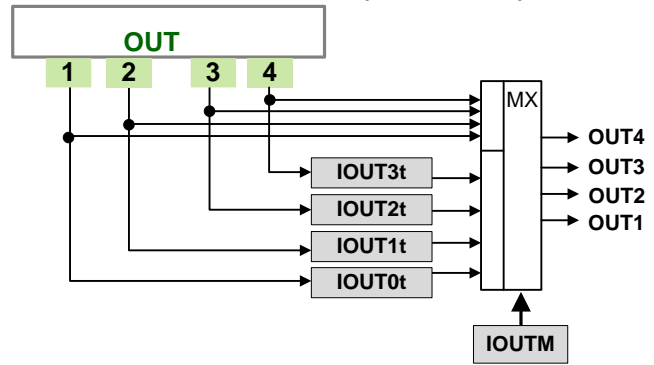


Может возникнуть ситуация, когда IO\_Slave получил запрос от IO\_Master, сформировал ответ и сбросил защелку, но ответ до IO\_Master не прошел. В данной ситуации изменения значения IN1 IO\_Slave для модуля IO\_Master **НЕ БУДЕТ** (как будто режим защелки у IO\_Slave отсутствует)

## 12.13 ФУНКЦИИ ВЫХОДОВ

Структурная схема выходов OUT1- OUT4 представлена на следующем рисунке:

Выходы могут устанавливаться как прямым постоянным значением при получении команд установки, так и проходить через процедуру генерации выходного импульса с заданным временем.



### 12.13.1 ВЫХОДНОЙ ИМПУЛЬС.

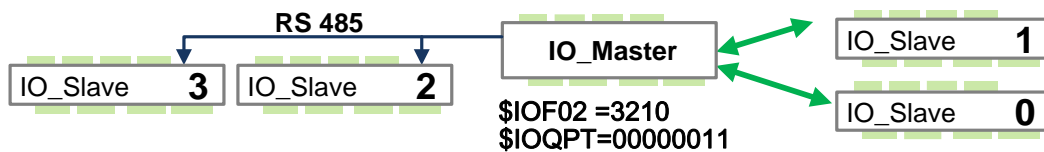
Помимо установки на выходах постоянного уровня, возможно формирование импульса заданной длительности после получения команды на установку выходов. В зависимости от конфигурации, импульс может быть положительным (переход из '0' в '1'), так и отрицательным (переход из '1' в '0'). Активация режима импульса на выходе и его полярность задается командой \$IOUTM. Длительность - \$IOUTxT, где x - номер выхода начиная с 0. Длительность кратна 100мс.

Допустим, выход OUT1 сконфигурирован на генерацию отрицательного импульса (\$IOUTM=---0) длительностью 10с (\$IOUT0T=0064). При получении команды на установку выхода OUT1 в значение '0' в режиме IO\_Slave или при транслировании на выход значения '0' в режиме IO\_Master, модуль выставляет на выходе '0' и запускает таймер IOUT0T. По истечении 10с, выход устанавливается в значение '1'. В течении всего времени работы таймера IOUT0T значение выхода OUT1 не меняется в не зависимости от полученной очередной команды. Также, если таймер IOUT0T уже запущен, то при получении команды на установку выхода в '0' он не перезапускается.

## 12.14 ОПРОС ПО RS485

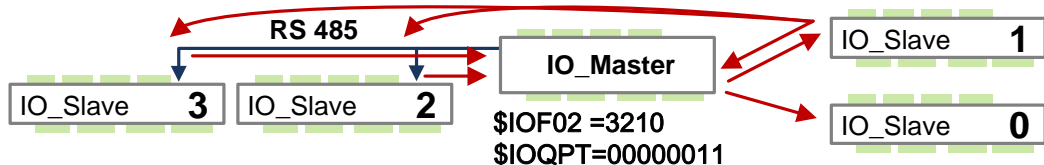
Модуль может опрашивать удаленные модули как по радиоэфиру, так и по интерфейсу RS485. Способ опроса задается в регистре \$IOQPT. Чтобы модуль в режиме IO\_Master опрашивал модуль IO\_Slave по RS485, необходимо установить соответствующий бит в 0.

Например, при конфигурации регистра \$IOQPT на рисунке ниже, модули n0 и n1 опрашиваются по эфиру, а модули n2 и n3 по RS485.



При опросе по RS485 необходимо учитывать тот факт, что модуль получив modbus запрос с чужим адресом, выдает его в эфир. А принимая запросы из эфира выдает на свой последовательный порт т.к. это требование принципа прозрачной передачи данных.

Рассмотрим опрос модуля 1 по эфиру из примера выше:



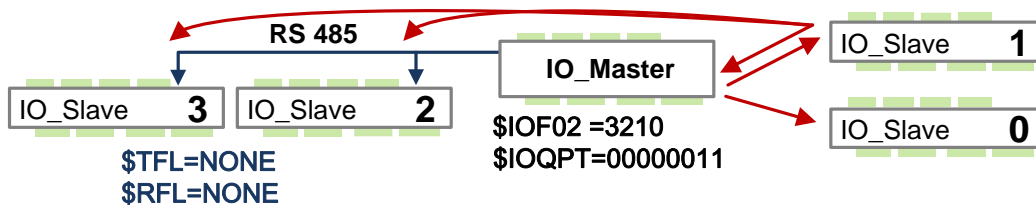
Красными стрелками показано "движение" запроса от IO\_Master к модулю n1 и его ответ. Модуль n1 получив по эфиру "свой" запрос, отвечает также по эфиру. Ответ принимают все модули в сети, включая модули n3 и n2. Если получение и выдача ответа модулем n0 на свой порт RS485 не критично, т.к. его порт никуда не подключен, то модули n3 и n2 получив ответ от модуля n1 выдают его на свой порт RS485. В итоге, модуль IO\_Master получает 3 ответа. Более того, данные с портов RS485 модулей n2 и n3



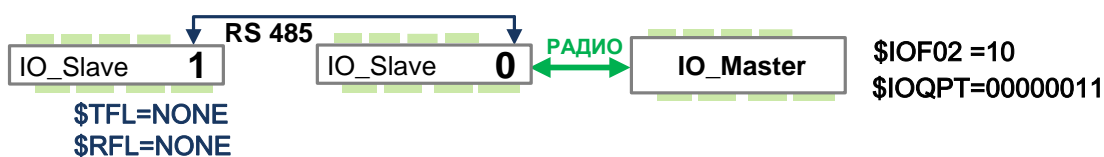
перекрестно поступают на порты друг друга, что приводит к выдаче их в эфир. Это приведет к закликиванию и не работоспособности системы.

Для предотвращения подобной ситуации в приведенной схеме включения, необходимо отключить модулям n3 и n2 передачу и прием прозрачных данных, либо переводить их на параметры, не совпадающие с параметрами передачи данных по эфиру всей системы.

Для отключения приема и передачи данных у модулей n2 и n3 необходимо задать запрещающие фильтры на передачу в эфир и прием из эфира. В этом случае, модули n2 и n3 получив по эфиру не "свой" запрос не будут выдавать его в порт RS485, а получив не "свой" запрос по RS485 не будут выдавать его в эфир:



Приведем еще одну конфигурацию:

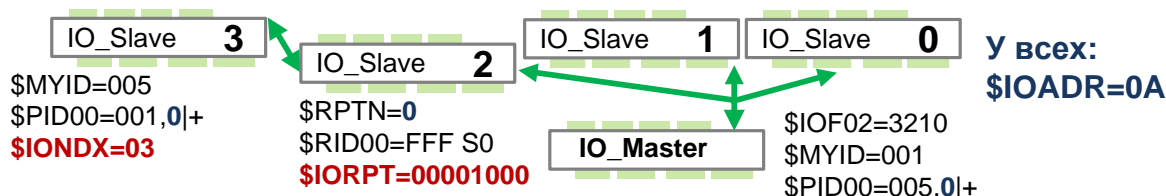


Модули с n0 и n1 опрашиваются по радио которые соединены друг с другом по RS485. Для модуля n1 также нужно отключить прием/передачу из эфира.

## 12.15 ФИЛЬТР ДЛЯ РЕТРАНСЛЯЦИИ \$IORPT

В модуле реализован дополнительный фильтр на уровне индексов модулей для ретрансляции в эфире modbus пакетов. Фильтр активируется если текущий модуль является ретранслятором. Фильтр программируется в регистре \$IORPT и содержит битовую маску индексов модулей, пакеты которых разрешены для ретрансляции. Старший бит отвечает за индекс 7, младший - за индекс 0. Если адрес в modbus пакете попадает в диапазон адресов от \$IOADR+0 до \$IOADR+7, то он проходит анализ на фильтр \$IORPT - если бит соответствующий номеру индекса установлен в 1, то этот пакет разрешен для ретрансляции, иначе нет.

Анализ фильтра \$IORPT осуществляется после анализа масок \$RID - если на уровне регистров \$RID пакет не разрешен для ретрансляции, а в фильтре \$IORPT разрешен, то пакет в итоге не будет ретранслирован. По умолчанию, фильтр на все индексы **включен** (ретрансляция разрешена). Работа фильтра \$IORPT показана на следующем рисунке:



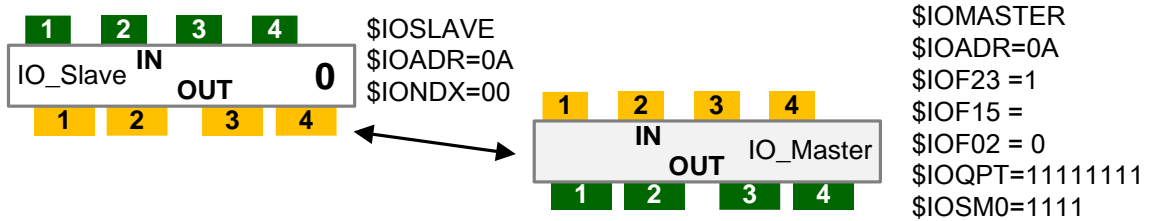
Допустим, модуль IO\_Master опрашивает 4 модуля IO\_Slave. Модули n0-n2 находятся в прямой радиовидимости, а с модулем n3 связь не стабильна. Модуль n2 назначен ретранслятором (\$RPT=00). Если фильтр разрешен полностью (по умолчанию), то модуль n2 будет ретранслировать абсолютно все пакеты, даже для модулей n1 и n0. Это приведет к тому, что в эфире будет много лишних и ненужных пакетов. Конечно, на уровне масок \$PID можно запрограммировать модули чтобы они воспринимали пакеты только от ретранслятора, но это усложняет конфигурацию каждого устройства.

Заданием параметра \$IORPT=00001000 у ретранслятора задача решается минимальной конфигурацией - будут ретранслироваться modbus пакеты только с адресом 0A+03, т.е. запросы и ответы модуля n3 и не будут к модулям n2-n0.

## 12.16 ТИПОВЫЕ СХЕМЫ ВКЛЮЧЕНИЯ

### 12.16.1 ТРАНСЛЯЦИЯ 4-Х СУХИХ КОНТАКТОВ В ОБЕ СТОРОНЫ (ТОЧКА-ТОЧКА)

В схеме работают два модуля: один в режиме IO\_Master, другой – в режиме IO\_Slave. Замыкание/размыкание любого из четырех входов одного модуля приведет к замыканию/размыканию контактов реле соответствующего выхода второго модуля и наоборот. Например, при замыкании входа IN1 модуля IO\_Slave сработает реле OUT1 модуля IO\_Master, а при замыкании входа IN3 модуля IO\_Master сработает реле OUT3 модуля IO\_Slave.



В цикл опроса мастера включен единственный (нет смысла тратить время на опрос не включенных в систему модулей) удаленный модуль IO\_Slave n0 (IOF02=0), для этого модуля заданы команды считывания входов (IOF02=0) и подмена ее на функцию F23 (IOF23=1). Таким образом, в каждом цикле в эфире будет 2 пакета - запрос на чтение/установку и ответ.

Чтобы разрешить установку всех своих выходов в соответствии с состояниями всех входов удаленного модуля n0, для мастера установлена маска IOSM10=1111.

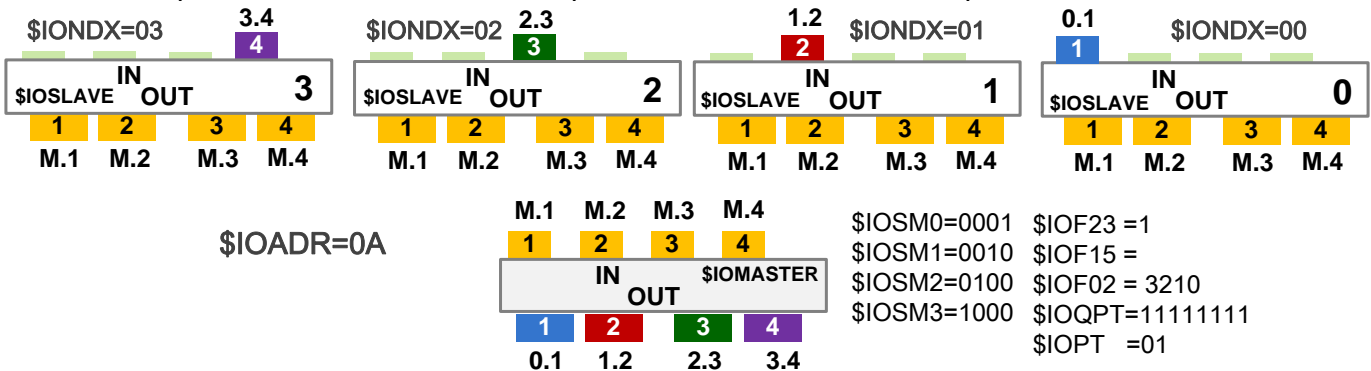
При такой схеме модули IO\_Slave и IO\_Master работают полностью симметрично.

### 12.16.2 ТРАНСЛЯЦИЯ 4-Х ВХОДОВ НА 4 УДАЛЕННЫХ ВЫХОДА. ТРАНСЛЯЦИЯ 4-Х УДАЛЕННЫХ ВХОДОВ НА ВЫХОДЫ.

В этой схеме работают пять модулей – один в режиме IO\_Master, четыре (на удаленных объектах) – в режиме IO\_Slave.

Замыкание/размыкание любого из четырех входов мастера приведет к замыканию/размыканию контактов реле на соответствующих выходах каждого из четырех удаленных модулей IO\_Slave. Например, при замыкании входа IN1 модуля IO\_Master сработают реле на выходах OUT1 всех четырех модулей IO\_Slave, таким образом, входы мастера транслируются на выходы всех IO\_Slave.

Замыкание входа IN1 первого удаленного модуля транслируется на выход OUT1 мастера, замыкание входа IN2 второго – на выход OUT2 мастера и так далее.



Четыре модуля на удаленных объектах запрограммированы на работу в режиме IO\_Slave, им присвоены последовательные индексы от 0 (IONDX=00) до 3 (IONDX=03).

Пятый модуль работает как IO Master, он формирует запросы и команды по радиоэфире каждую секунду (IOPT=01). В цикл опроса мастера включены все четыре удаленных модуля и для каждого из этих модулей заданы команды считывания входов (IOF02=3210) с подменой функции 02 на 23 (IOF23=1). Таким образом, цикл опроса состоит из четырех этапов по два пакета.

Замыкание/размыкание любого из четырех входов мастера приведет к замыканию/размыканию контактов реле на соответствующих выходах каждого из четырех удаленных модулей IO\_Slave. Например, при замыкании входа IN1 модуля IO\_Master

сработают реле на выходах OUT1 всех четырех модулей IO\_Slave. Таким образом, входы мастера транслируются на выходы всех IO\_Slave.

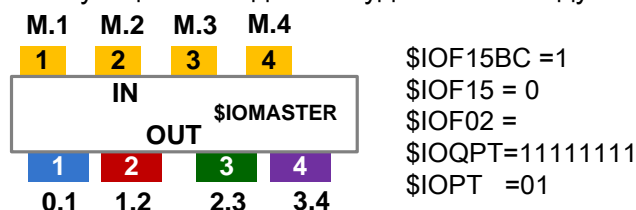
Поскольку у модуля имеются только четыре выхода, суммарное количество транслируемых на них удаленных входов не может превышать четырех (иначе входы одного удаленного модуля начнут "перебивать" входы другого). Поэтому трансляцию состояния входов от четырех модулей IO\_Slave на один IO\_Master можно обеспечить только при использовании одного входа на каждом из модулей IO\_Slave. Например, вход IN1 первого удаленного модуля транслируется на выход OUT1 мастера, вход IN2 второго – на выход OUT2 мастера и так далее. Чтобы неиспользуемые входы модулей IO\_Slave не влияли на выходы IO\_Master, их маскируют регистры IOSM0...IOSM3 модуля IO\_Master. В результате на выход OUT1 мастера будет влиять только вход IN1 удаленного модуля n0 (IOSM0=0001), а состояние остальных входов этого модуля мастер будет игнорировать. Аналогичным образом маскированы остальные три выхода мастера.

Если в приведенном примере нет необходимости в трансляции удаленных входов на выходы мастера (остается только управление удаленными выходами), "освободившиеся" от этой функции выходы мастера можно использовать для индивидуальной (в отличие от групповой с помощью выхода АВАРИЯ) индикации наличия/отсутствия связи с каждым из удаленных модулей IO\_Slave с использованием механизма безопасного состояния.

Для этого в приведенной для этого примера схеме необходимо входы IN4, IN3, IN2 и IN1 всех четырех удаленных модулей оставить разомкнутыми (логическое состояние 0), а в модуле IO\_Master разрешить включение безопасного состояния всех выходов, а в качестве собственно безопасных состояний для каждого выхода задать логическую 1 (IOSFV=1111). Теперь, при наличии связи с удаленными модулями выходы мастера будут разомкнуты (поскольку на них будут транслироваться разомкнутые входы удаленных модулей), а при пропадании связи с тем или иным модулем соответствующий выход будет замыкаться (сработает безопасное состояние). При этом при пропадании связи с удаленным модулем n0 замкнется выход OUT1 мастера, при пропадании связи с n1 замкнется выход OUT2 и так далее. При восстановлении связи соответствующий выход разомкнется. Таким образом, обеспечится индивидуальная индикация связи с каждым из удаленных модулей.

### 12.16.3 ТРАНСЛЯЦИЯ 4-Х ВХОДОВ НА ВЫХОДЫ ПРОИЗВОЛЬНОГО КОЛИЧЕСТВА МОДУЛЕЙ IO\_SLAVE.

Рассмотрим задачу трансляции состояния четырех входов одного модуля на четыре выхода произвольного количества удаленных модулей. Замыкание/размыкание любого из входов IN1...IN4 на модуле в "центре" приведет к замыканию/размыканию реле на соответствующем выходе всех удаленных модулей.



Для решения задачи запрограммируем все удаленные модули на работу в режиме IO\_Slave. Поскольку мастер способен адресно управлять не более чем восемью удаленными модулями, для управления произвольным количеством удаленных модулей нужно использовать режим широковещательной адресации. Для этого включим в нем широковещательный режим \$IOF15BC=1.

В такой системе Modbus адреса каждого из удаленных модулей IO\_Slave могут быть любыми и даже все одинаковыми, поскольку мастер будет посылать команду F15 (Set Coils) в широковещательном режиме и ее выполнят все получившие ее удаленные модули.

Для того чтобы мастер формировал широковещательные команды F15 в радиозфире, необходимо, кроме всего прочего, обязательно включить хотя бы один из удаленных модулей в цикл своего опроса регистре IOF15. В примере также запрещено чтение входов удаленных модулей (IOF02=), поскольку это не требуется в данной задаче.

В данной схеме, в отличие от схемы с индивидуальной адресацией, модуль IO\_Master не сможет определять пропадание связи с удаленными модулями IO\_Slave, поэтому реле на выходе АВАРИЯ и индикаторы АВАРИЯ модуля IO\_Master не будут работать. На удаленных

же модулях IO\_Slave данный выход и индикаторы будут функционировать в обычном режиме.

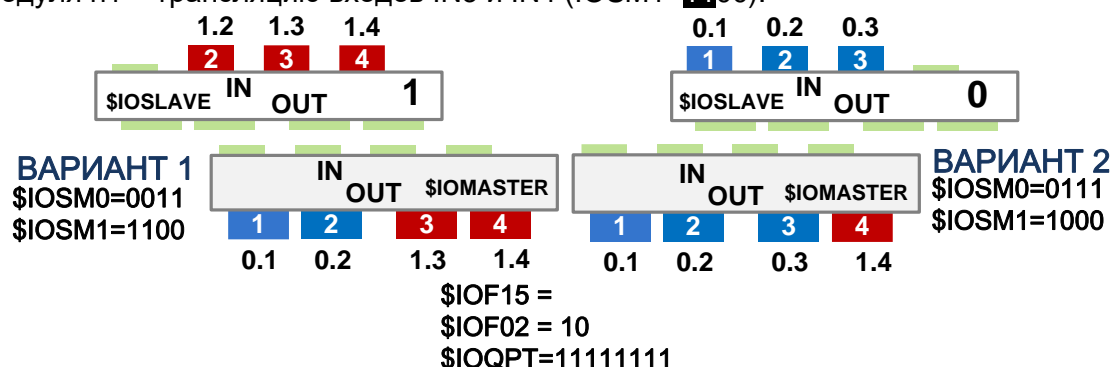
#### 12.16.4 ТРАНСЛЯЦИЯ НА ВЫХОДЫ ОДНОГО МОДУЛЯ ВХОДОВ НЕСКОЛЬКИХ УДАЛЕННЫХ МОДУЛЕЙ

Рассмотрим ситуацию, при которой требуется управлять выходами одного модуля в соответствии с состояниями входов на нескольких (до 4-х) удаленных модулях. Предположим, что имеются два удаленных модуля и два входа каждого из них нужно транслировать на выходы третьего модуля.

Схема и настройки для решения данной задачи показаны на рисунке ниже.

Два удаленных модуля переводим в режим IO\_Slave и присваиваем им адреса. Третий модуль делаем мастером (\$IOMASTER) и включаем в опрос модули с индексами 0 и 1 задав формирование команды чтения входов F02 (IOF2=10).

При такой конфигурации все 4 входа каждого из двух IO\_Slave будут транслироваться на 4 выхода мастера "перебивая" друг друга. Чтобы исключить это, нужно использовать маски IOSMx – для каждого IO\_Slave разрешить трансляцию на выход мастера только двух входов. Например, для модуля n0 разрешить трансляцию входов IN1 и IN2 (IOSM0=0011), а для модуля n1 – трансляцию входов IN3 и IN4 (IOSM1=1100).



Теперь входы IN1 и IN2 модуля n0 будут управлять выходами OUT1 и OUT2 мастера, а входы IN3 и IN4 модуля n1 – выходами OUT3 и OUT4 мастера (ВАРИАНТ 1).

Очевидно, данную схему легко модифицировать для трансляции других комбинаций входов удаленных модулей IO\_Slave – нужно просто изменить маски IOSM0 и IOSM1. Например, можно транслировать три входа с первого модуля и один – со второго (IOSM0=0111, IOSM1=1000) (ВАРИАНТ 2).

#### 12.16.5 УПРАВЛЕНИЕ МОДУЛЯМИ IO\_SLAVE ВНЕШНИМ "МАСТЕРОМ"

Пожалуй, одна из распространенных задача при использовании модулей "Спектр 868 IO Lora" – встраивание их в систему сбора/управления под управлением того или иного OPC сервера, ПЛК, пульта управления и т.д.

Как правило, в таких системах используются проводные модули ввода/вывода, а управление осуществляется по шине RS-485 с использованием протокола Modbus. В случаях, когда использование проводов невозможно или экономически невыгодно, применяют "прозрачные" радиоудлинители RS-485 (радиомодем "Спектр 868 Lora").

Таким образом, для встраивания модулей в систему сбора/управления достаточно установить на стороне мастера (ПЛК, OPC сервер и т.д.) радиомодем "Спектр 868 Lora". Управление модулями "Спектр 868 IO Lora" будет осуществляться в протоколе Modbus, при этом мастер системы будет работать с этими модулями точно так же, как он работал бы с проводными модулями ввода/вывода, подключенными к нему по шине RS-485.

Для такого использования модули "Спектр 868 IO Lora" нужно запрограммировать в режим IO\_Slave и присвоить им уникальные Modbus адреса. Мастер системы будет работать с модулями по этим адресам через радиомодем "Спектр 868 Lora".

Параметры передачи данных в радиоэфире (рабочий канал, скорость) всех модулей должны совпадать. Кроме этого, модули при такой схеме следует использовать в широкополосном режиме.

## 13 УДАЛЕННОЕ КОНФИГУРИРОВАНИЕ

В модуле имеется возможность удалённого конфигурирования по радио. Для понимания процесса удалённого конфигурирования вводится понятие *профиля*. Под профилем понимается набор всех параметров модуля, необходимых для его работы. Профиль хранится в ЭНОЗУ модуля. Профиль разделен на 2 части - общий (профиль модуля для передачи данных, ретрансляции и т.д.) и IO (профиль модуля ввода-вывода).

В целях безопасности, удалённое конфигурирование может быть запрещено. Для запрета необходимо установить бит \$LOCK.bRemoteCfgLock. Бит может быть сброшен как удалённо, так и по последовательному порту, а установлен только по последовательному порту.

В случае запрета удалённого конфигурирования все команды, относящиеся к данному процессу, исполняются как обычно, однако реальных изменений в удалённом модуле не происходит.

**Параметры задаваемые командами \$RG в профиль не входят** и могут изменяться только **локально**. Также, удаленно нельзя задать параметры шифрования.

### 13.1 УДАЛЁННОЕ КОНФИГУРИРОВАНИЕ В КОМАНДНОМ РЕЖИМЕ

Удалённое конфигурирование в командном режиме осуществляется с помощью ввода соответствующих текстовых команд:

- \$RPRF - чтение общего профиля модуля;
- \$RPIO - чтение профиля модуля ввода-вывода (IO);
- \$WPRF - запись общего профиля модуля;
- \$WPIO - запись профиля IO;
- \$EPS - начать редактировать удаленный профиль;
- \$EPE - остановить редактирование удаленного профиля;
- \$DMPR - вывод профиля удаленного модуля;

Команды чтения/записи профилей передаются в адресном режиме. До начала редактирования профиля его **необходимо загрузить** из удаленного модуля командами \$RPRF и/или \$RPIO. Далее, с помощью команды \$DMPR узнать текущую конфигурацию.



Команда \$DMPR выводит результат всегда - в не зависимости считан профиль удаленного модуля или нет. Если часть профиля не считана, команда будет выводить "мусор" вместо его реальных значений.

После прочтения профиля его можно редактировать также, как и профиль локального модуля. Для "старта" редактирования именно профиля удаленного модуля (который предварительно должен быть прочитан), необходимо ввести команду \$EPS (**E**dit **P**rofile **S**tart), иначе все вводимые команды будут изменять значения локального профиля. После окончания редактирования необходимо ввести команду \$EPE (**E**dit **P**rofile **E**nd).

После редактирования профиля его можно записать в удаленный модуль с помощью команд \$WPRF и/или \$WPIO. После получения данных команд, удаленный модуль автоматически перезагрузится через 5 сек.

При вводе команд \$W(R)PRF и \$W(R)PIO модуль в автоматическом режиме посылает удаленному модулю соответствующие запросы до тех пор, пока ответ не будет получен. Если удаленный модуль недоступен, запросы будут слаться бесконечно. Для остановки процесса необходимо нажать клавишу 's'.

Пример удалённого конфигурирования (сообщения подсказки во время выполнения команд, могут быть другие):

```
OK> $RPRF 002          чтение профиля удаленного модема с адресом 002
Press `s` to stop...
OK>
** profile read done.
OK> $COM=57600,8N1     команда изменяет ВСЕ ЕЩЕ локальный профиль
OK> $EPS              начать редактирование считанного профиля
*** Edit remote profile START***
OK> $COM=57600,8N1     команда изменяет считанной профиль
OK> $EPE              остановить редактирование считанного профиля
*** Edit remote profile END***
OK> $WPRF 002         запись профиля обратно в модем 002
Press `s` to stop...
OK>
** profile write done.  Модем 002 перезагрузится через 5 с
OK>
```



Если удаленный модуль на этапе выполнения команды \$WPRF "пропал" из эфира и у него были до этого изменены параметры связи по эфиру (скорость, частота) то возможно, что новые параметры у него уже вступили в силу и связь пропала в момент получения от него подтверждения. Для проверки необходимо перенастроить локальный модуль на такие же параметры и попробовать прочитать профиль удаленного модуля.

## 13.2 УДАЛЁННОЕ КОНФИГУРИРОВАНИЕ ПО MODBUS

Чтение и запись профилей как локального, так и удаленного модулей доступно по протоколу Modbus. Для этого модулю должен быть присвоен корректный адрес в системе Modbus командами \$IOADR и \$IONDX. Форматы сообщений описаны в разделе "[Карта регистров modbus](#)".

# 14 ТЕСТИРОВАНИЕ СВЯЗИ.

Тестирование связи между модулями возможно следующими способами:

- В командном режиме;
- В прозрачном режиме передачи данных;
- Режим "Маяк".

## 14.1 ТЕСТИРОВАНИЯ СВЯЗИ В КОМАНДНОМ РЕЖИМЕ

В командном режиме для тестирования связи доступны следующие команды: \$RFIND, \$RCHK и \$TEST. При выполнении данных команд, модули должны быть настроены на одну частоту (\$CH) и иметь одинаковую скорость в эфире (\$AR). Команды осуществляют опрос в бесконечном цикле. Для остановки цикла необходимо нажать в терминале символ 's'.

**\$RFIND** - поиск устройств в эфире. Команду удобно применять, когда собственные адреса (\$MYID) модулей не известны. Найденные модули выводятся списком.

```

OK> $RFIND
press 's' to stop...
OK>
** 1: ID=002 (0A:01), v1.00 (0), RSSI=-87,-82
** tx
** 2: ID=005 (0A:02), v1.00 (0), RSSI=-109,-96
** tx
** tx
OK>

```

Версия ПО  
ID устройства: 0 - СПЕКТР 868 IO Lora  
Уровень сигнала запроса  
Уровень сигнала ответа  
RPTN=5  
IOADB : IOADR  
MYID удаленного модема  
Номер повторителя (если активен)

**\$RCHK** - опрос удаленного модуля с заданным параметром \$MYID.

```

OK> $RCHK 002
press 's' to stop...
OK>
** v1.00 (0) ;Rt=525 ms,RSSI=-42,-40
** v1.00 (0) ;Rt=550 ms,RSSI=-42,-40
** v1.00 (0) ;Rt=550 ms,RSSI=-41,-41s
OK>

```

Уровень сигнала запроса  
Уровень сигнала ответа  
Версия ПО  
ID устройства  
Время отклика (Response Time)

**\$TEST** - запрос BER пакета у удаленного модуля с заданным параметром \$MYID. При запросе необходимо задать тип помехоустойчивого кодирования. Код задается аналогично команде \$FEC.

```

OK> $TEST 002 3I
press 's' to stop...
OK>
** (0,138): byte_err=0 bit_err=0 RSSI=-40,FEC=R:15,9I
** (1,138): byte_err=0 bit_err=0 RSSI=-41,FEC=R:15,9I
** (2,138): byte_err=0 bit_err=0 RSSI=-41,FEC=R:15,9Is
OK>

```

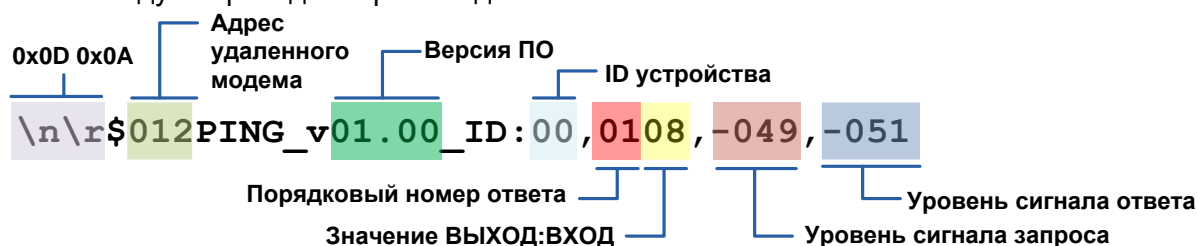
Байтовые ошибки  
Битовые ошибки  
Порядковый номер пакета + размер (байт)  
Тип FEC  
Уровень сигнала ответа

## 14.2 ТЕСТИРОВАНИЯ СВЯЗИ В ПРОЗРАЧНОМ РЕЖИМЕ

В прозрачном режиме передачи данных необходимо послать в активный порт локального модуля ASCII строку без пауз между символами вида \$hhhPING\_RATEOSp, где hhh - адрес удаленного модуля, p - мощность (0...3), на которой удаленный модуль будет передавать ответ. Если нет необходимости задавать мощность у вызываемого модуля, вместо значений 0...3 можно послать любой другой символ, например, символ 'x'. Адрес может быть как групповой, так и индивидуальный.

При получении строки \$hhhPING\_RATEOSp удаленный модуль с адресом hhh переключает текущую мощность на значение 'p' (если задано). Новое значение мощности не

сохраняется в профиле модуля, однако действует на весь оставшийся сеанс работы (до перезагрузки или выключения питания). После этого, удаленный модуль отвечает и на локальный модуль приходит строка вида:



Примеры:

**Запрос:** \$FFFFPING\_RATEOS0 (при получении, передавать ответ на PWR=0)

Ответ: \$002PING\_v01.00\_ID:00,0208,-049,-045

В сети обнаружен модуль 002. Версия ПО 1.00. ID устройства 0 (СПЕКТР 868IO Lora). Порядковый номер ответа 02. Значение выходов 0 (0000), значение входов 8 (1000). Уровень сигнала при приеме запроса удаленным модулем -49 дБ. Уровень сигнала при приеме ответа -45 дБ.

**Запрос:** \$FFFFPING\_RATEOS1 (при получении, передавать ответ на PWR=1)

Ответ: \$002PING\_v01.00\_ID:00,0208,-049,-045

\$003PING\_v01.00\_ID:00,0108,-049,-045

В сети присутствуют два модуля с адресами 002 и 003.

**При групповом запросе обнаружение всех модулей не гарантируется.**

**Запрос:** \$123PING\_RATEOSx (при получении, не изменять мощность)

Ответ: \$123PING\_v01.00\_ID:00,0108,-000,-000

Вызываемый модуль является локальным, о чем свидетельствует значение уровней сигнала равное 0.

Также, для проверки связи в лабораторных условиях можно использовать другой способ. Для этого нужны будут два компьютера (ПК) или один ПК с двумя COM-портами (с двумя переходниками USB-RS-232/485), на котором с каждым из двух портов работает отдельная терминальная программа (либо в одной программе открыты два последовательных порта). Естественно, параметры портов программы должны соответствовать параметрам RS-485 модулей.

К ПК следует подключить два проверяемых модуля — каждый к «своему» переходнику USB-RS-485/232 (в командный режим переводить модули не нужно!). В качестве антенн при проверке в лабораторных условиях (в пределах комнаты) можно использовать как штатные антенны, так и просто отрезок провода (5...10 см) в антенном разъёме.

Для проверки связи отправляйте произвольные символы в окне одной терминальной программы — они должны передаться через модули в окно второй программы, и наоборот.

Если символы проходят в обе стороны — модули по радио настроены правильно.

### 14.3 РЕЖИМ МАЯК

В режиме маяка, модуль автоматически, выдает в эфир строку \$hhhPING\_RATEOSp. Где hhh - адрес удаленного модуля, который задается регистрами \$RG04-05, а 'p' - текущая мощность модуля в профиле. Удаленный модуль получив её, выдает ответ. В данном режиме можно проверять связь только между 2-мя модулями. В противном случае, если в сети более 2х удаленных модулей, то не будет понятно от какого именно модуля приходит ответ. Режим маяка, если включен, активизируется каждый раз при включении питания. Чтобы его отключить, необходимо выполнить процедуру отключения через команду загрузчика BOOT\_BEACON\_OFF.

В режиме маяка функции IO отключаются, а индикатор MODE индицируется получение ответа от удаленного модуля - в случае получения ответа, индикатор MODE кратковременно загорается зеленым, если нет - красным.



## 15 ТЕХНОЛОГИЧЕСКИЕ ПАРАМЕТРЫ. РЕГИСТРЫ RG

Переменные изменяемые командой \$RG не входят в общий профиль модуля и не могут быть изменены удаленно. Вывод всех значений осуществляется командой \$DRG, сброс всех регистров по умолчанию - командой \$IRG. Далее приведена таблица значений регистров RG (все адреса десятичные, под именем регистра указано значение по умолчанию):

Адрес	Имя	Значение
00	485_PRE_TMR (0xFF)	Время паузы перед передачей первого байта после активизации шины RS485. Задается в битовых интервалах текущей скорости порта (\$COM). Значение 0x00 и 0xFF соответствуют 0x01.
01	485_PST_TMR (0xFF)	Время удержания шины RS485 в активном состоянии после передачи стоп-бита последнего байта. Задается в битовых интервалах текущей скорости порта (\$COM). Значение 0x00 и 0xFF соответствуют 0x01.
02-03	C485/C232 (0xC4 0x85)	Активный порт для передачи данных: 0xC4 0x85 - активный порт RS485; 0xC2 0x32 - активный порт USB или RS232. Значение задается при выборе порта в режиме загрузчика. Переключить порты также можно командой <a href="#">\$EXCHP</a> .
04-05	BEAC_EN (0xFF 0xFF)	Активизация режима "Маяк" 0xBE 0xAC - режим "Маяк" активен. Все остальные значения - не активен. Значение задается в режиме загрузчика.
06-07	BEAC_ADR (0xFF 0xFF)	Адрес вызова в режиме "Маяк". Адрес подставляется в строку \$hhhPING_RATEOSp вместо символов hhh.
08	MISC1 (0xFF)	Битовый регистр: 7...2 - резерв 1...0 - яркость индикатора (при его наличии). Регистр изменяется внутренним ПО.
09	CMD_on_DBG (0xFF)	Признак входа в командный режим по отладочному порту через кнопку CFG. Активное значение 0xCD. Остальные значения - командный режим по порту передачи данных. Регистр изменяется внутренним ПО.
10	RfOff (0xFF)	Отключение приемопередатчика. Активное значение 0x0F
14	Gfsk (0xFF)	Активное значение 0x73. В текущей реализации ПО не имеет смысла.
15	dbg_dump_cfg (0xFF)	Печать пользовательских данных на отладочном порту. см. раздел " <a href="#">Отладочный порт</a> ".

## 16 КОМАНДЫ УПРАВЛЕНИЯ МОДУЛЕМ

При описании команд приняты следующие сокращения:

**b** - двоичное число 0 или 1;

**d** - десятичное число от 0 до 9;

**h** - шестнадцатеричное число от 0 до 9 и от A до F.

**c** - символ. Определен по тексту описания команды.

Другие сокращения определены в тексте описания команды. При вводе значения, ведущие нули должны присутствовать.

### 16.1 \$DMP(R) — ВЫВОД ПРОФИЛЯ (УДАЛЁННОГО) МОДУЛЯ

Ввод: **\$DMP** или **\$DMPR**

Пример:

```
OK> $DMP
CH=00 TXID=FFF COM=9600,8N1 ОБЩИЙ ПРОФИЛЬ
PWR=0 MYID=002 DAT=EOT MAXP=0
AR=3 ACKT=016 EOT=0015 PLEN=000 EOC=0AT
BCN=1 RESPT=000 FEC=H:12,8I (MaxSz=140)
BCT=005 RETRY=000 MDA=00000001 CONCT=000
IOADR=0A+IONDX=01 (0Bh) TFL=---- RPT=-
IORPT=11111111 RFL=---- RTFL=----
MaxTt=715 ms,ACKT >=016,*SK,*KEY,*LID
-----
IOSLAVE IOALR=F1
IOSFT=010 IOSFV=---- ПРОФИЛЬ МОДУЛЯ
IOINL=---- IODBN=00000000h ВВОДА-ВЫВОДА
IOINF=00 IOFR[1:0] =0000,0000
IOUTM=---- IOUT[3:0]t=0000,0000,0000,0000
IOINT=00000000
IOMOD=00000000
OK>
```



Профиль разделен на 2 части. Всё что находится выше пунктирной линии на рисунке (саму линию команда \$DMP не выводит) является общим профилем. Всё что ниже - профилем модуля ввода-вывода. Это необходимо учитывать в случае возможного удаленного конфигурирования. Верхняя часть профиля считывается/записывается командами \$RPRF/WPRF, а нижняя - \$RPIO/\$WPIO.

В выводе команды отображаются следующие "подсказки":

- MaxSz=140** максимальный размер пакета данных в эфире в зависимости от помехоустойчивого кодирования (\$FEC) и/или шифрования AES;
- MaxTt=715 ms** максимальное время передачи пакета на выбранной скорости (\$AR);
- ACKT >= 016** минимально рекомендуемое значения параметра ожидания подтверждения в режиме "точка-точка", а также для внешнего оборудования (\$ACKT) в x100 мс;
- \*SK** наличие ключа шифрования XOR. При выводе удаленного профиля, подсказка соответствует все равно локальному профилю;
- \*KEY** наличие ключа шифрования AES. При выводе удаленного профиля, подсказка соответствует все равно локальному профилю;
- \*LID** наличие активных ячеек \$RID или \$PID.

### 16.2 \$IEE — СБРОС ПАРАМЕТРОВ ПО УМОЛЧАНИЮ

Ввод: **\$IEE hhh**, где hhh — требуемое значение MYID модуля.

После ввода команды происходит инициализация параметров, хранящихся в ЭНОЗУ значениями по умолчанию. Адрес MYID модуля становится равным hhh. После выполнения команды происходит автоматический сброс модуля.



Команда \$IEE затрагивает следующие параметры:

- пароли шифрования (сброс осуществляется командами **\$KEA=** или **\$SK=000000**);
- модуль ввода-вывода (сброс осуществляется командой **\$IOIEE**, **\$IORIN** или **\$IOROUT**);
- технологические параметры (сброс осуществляется командой **\$IRG**);

**16.3 \$CH — ИЗМЕНЕНИЕ РАБОЧЕГО ЧАСТОТНОГО КАНАЛА**Ввод: **\$CH=dd**, где **dd** - номер рабочего канала от 00 до 03

Канал	Частота, МГц
00	868,765 (по умолчанию)
01	868,890
02	869,015
03	869,140

После ввода команды,  
модуль сразу перенастраивается  
на новый канал и сохраняет его в ЭНОЗУ

**16.4 \$PWR — УСТАНОВКА МОЩНОСТИ ПЕРЕДАТЧИКА**Ввод: **\$PWR=d**, где **d**:

PWR	Значение мощности, мВт
0	25 (по умолчанию)
1	100
2	200
3	350

После ввода команды,  
модуль сразу перенастраивается  
на новую мощность и сохраняет её в ЭНОЗУ

**16.5 \$AR — СКОРОСТЬ ПЕРЕДАЧИ ДАННЫХ ПО ЭФИРУ**Ввод: **\$AR=d**, где **d**- значение 0-7.

AR	Скорость (бит/сек)	Ширина полосы (КГц)
0	980	125
1	1760	125
2	2700	62.5
3	3125 (по умолчанию)	125
4	4500	62.5
5	5470	125
6	9300	125
7	22000	125

После ввода команды,  
модуль сразу перенастраивается  
на новую скорость и сохраняет её в ЭНОЗУ

**16.6 \$MYID — ИЗМЕНЕНИЕ СОБСТВЕННОГО АДРЕСА МОДУЛЯ**Ввод: **\$MYID=hhh**, где **hhh** — любое значение, кроме FFF, Fxx, xFх или ххF или 000.**16.7 \$TXID — ИЗМЕНЕНИЕ АДРЕСА ВЫЗЫВАЕМОГО МОДУЛЯ**Ввод: **\$TXID=hhh**, где **hhh** — любое значение, кроме 000

Команда задает адрес получателя пакетов, устанавливая тем самым режим работы модуля в эфире.

Ввод значений вида FFF, Fxx, xFх или ххF означает широкоэвещательный (групповой) режим передачи данных.

Ввод значений, отличающихся от FFF, Fxx, xFх или ххF, означает режим «точка-точка» с модулем, чей адрес MYID совпадает с введенным значением hhh.

**16.8 \$FEC — ТИП FEC ДЛЯ ИНФОРМАЦИОННЫХ ПАКЕТОВ В ЭФИРЕ**Ввод: **\$FEC=dc**, где**d** – тип помехоустойчивого кода (FEC):

- 0 - RS (7.5)
- 1 - RS (7.3)
- 2 - RS (15.11)
- 3 - RS (15.9)
- 4 - HAM (12.8)

5-7 - код выключен

**c** – признак перемежения:

- 'I' - перемежение включено
- 'N' - перемежение выключено

Пример: `$FEC=0I`  
`$FEC=5N`

Если помехоустойчивое кодирование не используется переключение не имеет значения, однако должно быть введено. Правильность ввода можно проверить командой \$DMP.

## 16.9 \$COM — ПАРАМЕТРЫ ПОСЛЕДОВАТЕЛЬНОГО ПОРТА

Ввод: `$COM=Rate Word Parity nStop`, где  
**Rate** - скорость обмена: 1200, 2400, 4800, 9600, 19200, 38400, 57600 или 115200;  
**Word** - размер слова, бит: 7 или 8;  
**Parity** - четность: N - none, O - odd (нечет), E - even (чет), M - mark, S - space;  
**nStop** - число стоповых бит: 1 или 2.

Пример: `$COM=9600,8N1`  
`$COM=19200,7E2`

## 16.10 \$EXCHP — СМЕНА ПОРТА

Ввод: `$EXCHP`  
С помощью данной команды происходит смена портов между активным (для передачи данных) и отладочным.

Если вход в командный режим был по отладочному порту, то после ввода команды этот порт будет активным, а другой отладочным и наоборот.

К примеру, если активным портом является RS485 (USB отладочным) и через него был вход в командный режим, то при вводе в командном режиме команды \$EXCHP порт RS485 будет отладочным, а USB - для передачи данных (активный).

После ввода команды \$EXCHP модуль автоматически перезагружается.

## 16.11 \$MDA — РЕЖИМ РАБОТЫ МОДУЛЯ

Ввод: `$MDA=bbbbbbbb`

Бит	Назначение	1	0
7	<b>PAC#1</b> Не устанавливать в 1	Да	Нет
6-1	Резерв		
0	<b>bDataMultEnb</b> Разрешение мультипликации данных внутри пакета передаваемого в эфир.	Да	Нет

## 16.12 \$ACKT — ВРЕМЯ ОЖИДАНИЯ ПОДТВЕРЖДЕНИЯ В РЕЖИМЕ «ТОЧКА-ТОЧКА»

Ввод: `$ACKT=ddd`, где ddd — десятичное число (000-255).

Шаг 100 мс. Если по истечении времени ACKT с момента окончания отправки пакета не получено подтверждения о доставке от адресуемого модуля, отправка пакета повторяется (см. раздел «Индивидуальный режим («точка-точка»»).



Минимальное время ACKT с учетом текущей скорости в эфире отображается в результате вывода команды \$DMP в строке "ACKT>=". Параметр \$ACKT не следует устанавливать ниже этого значения (если используется режим передачи данных "точка-точка").

Несмотря на то, что параметр имеет смысл только в режиме "точка-точка", данное время также нужно устанавливать у внешнего оборудования.

## 16.13 \$DAT — ТИП ПРОТОКОЛА ВХОДЯЩИХ ДАННЫХ

Ввод: `$DAT=EOT` (по умолчанию)  
Ввод: `$DAT=EOC`  
Ввод: `$DAT=RTU`  
Ввод: `$DAT=DL8`

## 16.14 \$MAXP — МАКСИМАЛЬНОЕ ЧИСЛО ПАКЕТОВ В БУФЕРЕ

Ввод: `$MAXP=d`, где d — десятичное число (0-7).

Задаёт максимальное число пакетов которые могут находиться одновременно в буфере на передачу в эфир. Значение 0 (по умолчанию) соответствует максимально возможному числу пакетов.

### 16.15 \$EOT — ТАЙМ-АУТ ПРИЕМА ПАКЕТА В БУФЕР НА ПЕРЕДАЧУ

Ввод: **\$EOT=dddd**, где dddd — число от 0000 до 9999. Значение 0000 соответствует 0001

Время задается следующим образом:

**0000...4999** время в мс.

**5000...9999** время в 0.5мс x (dddd-5000)

Пример: **\$EOT=1000** - 1000мс

**\$EOT=5001** - (5001-5000)\*0.5= 0.5 мс (500 мкс)

### 16.16 \$PLEN — НАРЕЗКА ВХОДЯЩИХ ДАННЫХ

Ввод: **\$PLEN=ddd**, где ddd — десятичное число (000-255). Ведущие нули должны присутствовать.

Параметр задает размер "нарезки" пакетов в режиме EOT в байтах x 8. Значению 000 соответствует запрещение "нарезки".

### 16.17 \$EOC — СИМВОЛ ПЕРЕДАЧИ ДАННЫХ

Ввод: **\$EOC=hhc**, где

**hh** - символ передачи данных

**c** - признак передачи самого символа:

'T' - передавать символ.

'N' - не передавать символ

Пример: | **\$EOC=31T** - передача по символу 0x31 ('1'). Сам символ передается.

Параметр имеет смысл только в режиме EOC.

### 16.18 \$RESPT — ВРЕМЯ ЗАДЕРЖКИ ОТПРАВКИ ПОДТВЕРЖДЕНИЯ В РЕЖИМЕ «ТОЧКА-ТОЧКА»

Ввод: **\$RESPT=ddd**, где ddd — число от 000 до 255.

Шаг 10 мс. Значение 000 означает отсутствие задержки. *Параметр временно не задействован.*

### 16.19 \$RETRY — ЧИСЛО РЕТРАНСЛЯЦИЙ ПАКЕТОВ, ТРЕБУЮЩИХ ПОДТВЕРЖДЕНИЯ

Ввод: **\$RETRY=ddd**, где ddd — число от 000 до 255. *Параметр временно не задействован.*

### 16.20 \$RPT — НОМЕР ПОВТОРИТЕЛЯ МОДУЛЯ

Ввод: **\$RPT=dh**, где

**d** - номер повторителя 0...9. Значения больше 7 означает отключение функции повторителя;

**h** - задержка ретрансляции значения от 0...F (x100мс).

Всего в радиосети может быть до восьми повторителей (номера 0...7). Каждый повторитель должен иметь уникальный номер. В выводе команды \$DMP значение команды выводится как RPT=d,dly:h

### 16.21 \$RID — АДРЕС ПАКЕТА РАЗРЕШЕННОГО ДЛЯ РЕТРАНСЛЯЦИИ

**\$RIDdd=hhh Tb [dddddddS<sub>1</sub>S<sub>2</sub>]**, где

**dd** Номер ячейки от 00 до 07;

**hhh** адрес, или маска адреса;

**T** Тип адреса: 'S' - hhh является адресом отправителя (**S**ource), 'D' - hhh является адресом получателя (**D**estination)

**b** разрешение на повторение пакета с нулевым РП (ретрансляционным полем):

0 - запрещено

1 - разрешено

**Не обязательные поля:**

**ddddddd** Значение маски. Вводится путем перечислением номеров повторителей от 0 до 7 в произвольном порядке.

**S<sub>1</sub>** символ логической операции между РП пакета и маской ddddddd:

«&» операция «И»;

«|» операция «ИЛИ».

**S<sub>2</sub>** действие над пакетом в случае совпадения РП пакета с маской ddddddd:

«+» пакет может быть ретранслирован (после анализа RIDxx)

«-» пакет не будет ретранслирован (после анализа RIDxx)

Если анализ маски разрешен, решение о ретрансляции/не ретрансляции пакета производится **только после** анализа RIDxx. При наличии активных ячеек \$RID в выводе команды \$DMP отображается подсказка "**\*LID**".

**Пример:** \$RID00=123 S1 014&+

Адрес записывается в ячейку 00. Ретранслируются пакеты, в заголовке которых адрес отправителя равен 123. Пакеты от модуля 123 с нулевыми значениями РП разрешены для повторения. Пакет от модуля 123 повторяется только в том случае, если он прошел через ретрансляторы 0 **И** 1 **И** 4.

\$RID07=03F D0 401|+

Адрес записывается в ячейку 07. Ретранслируются пакеты, предназначенные для группы 03. Пакеты с нулевыми значениями РП не ретранслируются; это означает, что пакет уже должен был быть ретранслирован каким-либо другим ретранслятором. Пакет группе 03 повторяется только в том случае, если он прошел через ретрансляторы 4 **ИЛИ** 0 **ИЛИ** 1.

\$RID06=012 S0

Адрес записывается в ячейку 06. Ретранслируются пакеты, в заголовке которых адрес отправителя равен 012. Пакеты от модуля 012 с нулевыми значениями РП не разрешены для повторения; это означает, что пакет уже должен был быть повторен каким либо другим ретранслятором.

## 16.22 \$PID — АДРЕС ПАКЕТА ДЛЯ РАСШИРЕННОГО ПРИЁМА

\$PIDdd=hhh [dddddddS<sub>1</sub>S<sub>2</sub>] ,где

dd номер ячейки от 00 до 07;

hhh адрес, или маска адреса;

**Не обязательные поля:**

ddddddd Значение маски. Вводится путем перечислением номеров повторителей от 0 до 7 в произвольном порядке.

S<sub>1</sub> символ логической операции между РП пакета и маской ddddddd:

«&» операция «**И**»;

«|» операция «**ИЛИ**».

S<sub>2</sub> действие над пакетом в случае совпадения РП пакета с маской ddddddd:

«+» пакет принимается (иначе не принимается);

«-» пакет не принимается (иначе принимается).

Если анализ маски разрешен и на основе анализа операции маски над РП пакет может быть принят, окончательное решение о приеме/не приеме пакета принимается на следующем уровне приема пакетов (как при обычном приеме).

В случае отсутствия маски адрес PID автоматически становится адресом базовой станции.

Если в модуле есть хотя бы один активный адрес базовой станции, при приеме пакета модуль сравнивает адрес отправителя пакета с адресом базовой станции. Если адреса не совпадают, пакет игнорируется (но может ретранслироваться, если адрес получателя/отправителя совпадает с одним из RIDxx). При наличии активных ячеек \$PID в выводе команды \$DMP отображается подсказка "**\*LID**".

## 16.23 \$LID / \$XID — ВЫВОД / УДАЛЕНИЕ АДРЕСОВ ДЛЯ РЕТРАНСЛЯЦИИ И РАСШИРЕННОГО ПРИЁМА

Ввод: \$LID

Вывод списка адресов для расширенной ретрансляции и приема.

Ввод: \$XIDdd, где dd — номер ячейки адреса от 00 до 07. Команда удаляет ячейку с адресом RID или PID. При вводе номера ячейки больше 07, удаляется весь список сразу.

Пример:

Вывод списка:

```

OK> $LID
#  xID Adr Rst RptPath
-----
00 RID 001 S1 4567 &+
01 RID 002 D0 01 |-
02 ---
03 RID 003 S0
04 PID 004
05 ---
06 ---
07 PID 007 05 &-
OK>

```

Удаление 2-х начальных ячеек:

```

OK> $XID00
OK> $XID01
OK>

```

Вывод списка после удаления:

```

OK> $LID
#  xID Adr Rst RptPath
-----
00 ---
01 ---
02 ---
03 RID 003 S0
04 PID 004
05 ---
06 ---
07 PID 007 05 &-
OK>

```

## 16.24 \$xFL — СПОСОБ ФИЛЬТРАЦИИ ДЛЯ ДАННЫХ И РЕТРАНСЛЯЦИИ

Ввод: **\$TFL=cccc** или **\$RFL= cccc** или **\$RTFL= cccc**, где  
**cccc** способ фильтрации ----, RTUA, DL8A, 1BYT или NONE

Пример: **\$TFL=----**  
**\$RFL=RTUA**

Способы фильтрации описаны в разделе "[Фильтрация данных](#)".

## 16.25 \$xFLDMP — ВЫВОД ЗНАЧЕНИЙ ФИЛЬТРОВ

Ввод: **\$TFLDMP** или **\$RFLDMP** или **\$RTFLDMP**

## 16.26 \$xFL+ — ВКЛЮЧЕНИЕ ДИАПАЗОНА ЗНАЧЕНИЙ В ФИЛЬТР

Ввод: **\$TFL+aabb** или **\$RFL+aabb** или **\$RTFL+aabb**, где  
**aa** начальное значение в шестнадцатеричном формате;  
**bb** конечное значение в шестнадцатеричном формате.

Для ввода 1-го значения достаточно указать равные значения aa и bb.

Пример: **\$TFL+0313**. Включить в фильтр для передаваемых данных значения от 0x03 до 0x13  
**\$RTFL+5A5A**. Включить в фильтр для ретрансляции значение от 0x5A

## 16.27 \$xFL- — ИСКЛЮЧЕНИЕ ДИАПАЗОНА ЗНАЧЕНИЙ ИЗ ФИЛЬТРА

Ввод: **\$TFL-aabb** или **\$RFL-aabb** или **\$RTFL-aabb**.

Аргументы аналогичны команде \$xFL+.

## 16.28 \$FLS — ЗАПИСЬ ЗНАЧЕНИЙ ФИЛЬТРОВ В ЭНОЗУ

Ввод: **\$FLS**

По данной команде все значения фильтров (для исходящих данных и ретрансляции) записываются в ЭНОЗУ (сами способы фильтрации сохраняются в ЭНОЗУ по команде \$S)

## 16.29 \$BCN — МАКСИМАЛЬНОЕ ЧИСЛО ОДИНАКОВЫХ ШИРОКОВЕЩАТЕЛЬНЫХ ПАКЕТОВ

Ввод: **\$BCN=d**, где d число от 0 до 9.

Значение больше 7 соответствует 7. Значение 0 соответствует 1.

Для уменьшения вероятности потери данных в ширококвещательном режиме (см. раздел «Режимы работы по эфиру (гарантии доставки данных адресату)»), когда подтверждения о доставке отсутствуют, можно последовательно передавать несколько копий ширококвещательного пакета.

При получении адресатом дублируемые ширококвещательные пакеты игнорируются.

## 16.30 \$BCT — ВРЕМЯ МЕЖДУ ПЕРЕДАЧАМИ ШИРОКОВЕЩАТЕЛЬНЫХ ПАКЕТОВ

Ввод: **\$BCT=ddd**, где ddd значение от 000 до 255.

Шаг 100 мс. Значение 000 соответствует отсутствию задержки.

Параметр активизируется только в широкопередаточном режиме и задает время между последовательной передачей копий (параметр \$BCN > 1) широкопередаточного пакета.

### 16.31 \$RPRF — ЧТЕНИЕ ОСНОВНОГО ПРОФИЛЯ УДАЛЁННОГО МОДУЛЯ

Ввод: \$RPRF hhh, где hhh — индивидуальный адрес удалённого модуля.

Команда для удаленного конфигурирования (см. раздел «[Удалённое конфигурирование](#)»).

### 16.32 \$RPIO — ЧТЕНИЕ ПРОФИЛЯ IO УДАЛЁННОГО МОДУЛЯ

Ввод: \$RPIO hhh, , где hhh — индивидуальный адрес удалённого модуля.

Команда для удаленного конфигурирования (см. раздел «[Удалённое конфигурирование](#)»).

### 16.33 \$WPRF — ЗАПИСЬ ОСНОВНОГО ПРОФИЛЯ УДАЛЁННОГО МОДУЛЯ

Ввод: \$WPRF hhh, где hhh — индивидуальный адрес удалённого модуля.

Команда для удаленного конфигурирования (см. раздел «[Удалённое конфигурирование](#)»).

### 16.34 \$WPIO — ЗАПИСЬ ПРОФИЛЯ IO УДАЛЁННОГО МОДУЛЯ

Ввод: \$WPIO hhh, где hhh — индивидуальный адрес удалённого модуля.

Команда для удаленного конфигурирования (см. раздел «[Удалённое конфигурирование](#)»).

### 16.35 \$EPS — НАЧАТЬ РЕДАКТИРОВАНИЕ УДАЛЁННОГО ПРОФИЛЯ

Ввод: \$EPS

Команда для удаленного конфигурирования (см. раздел «[Удалённое конфигурирование](#)»).



После ввода этой команды все команды, редактирующие профиль, относятся к профилю **удалённого** модуля. Для возврата к редактированию профиля локального модуля, необходимо выполнить команду \$EPE или выйти из командного режима.

### 16.36 \$EPE — ОСТАНОВИТЬ РЕДАКТИРОВАНИЕ ПРОФИЛЯ УДАЛЁННОГО МОДУЛЯ

Ввод: \$EPE

После ввода этой команды все команды, редактирующие профиль, относятся к профилю локального модуля. Команда для удаленного конфигурирования (см. раздел «[Удалённое конфигурирование](#)»).

### 16.37 \$UPD - СМЕНА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Ввод: \$UPD

После ввода этой команды модуль перейдет в режим смены ПО по текущему порту. Если команда введена ошибочно, достаточно переключить питание модулю.

### 16.38 \$RCHK — ОПРОС УДАЛЁННОГО МОДУЛЯ

Ввод: \$RCHK hhh, где hhh – адрес удалённого модуля.

Результат работы команды описан в разделе "[Тестирование связи в командном режиме](#)".

### 16.39 \$RFIND — ПОИСК УСТРОЙСТВ В ЭФИРЕ

Ввод: \$RFIND

Принцип работы команды описан в разделе "[Тестирование связи в командном режиме](#)".

**Пример:**

```
OK> $RFIND
  press 's' to stop...
OK>
** 1: ID=002(0A:01),v1.00(0),RSSI=-43,-41
** 2: ID=005(0A:02),v1.00(0),RSSI=-108,-94 RPTN=5
** txs
OK>
```



## 16.40 \$TEST — ЗАПРОС ТЕСТОВЫХ ПАКЕТОВ

Ввод: **\$TEST hhh,dc**, где  
**hhh** - адрес удаленного модуля;  
**d** – тип помехоустойчивого кодирования;  
**c** – признак перемежения;

По этой команде локальный модуль начинает запрашивать у удаленного модуля с адресом hhh тестовые пакеты. Тип помехоустойчивого кодирования при вводе аналогичен команде \$FEC. Результат работы команды описан в разделе "[Тестирование связи в командном режиме](#)".

**Пример:**

```

ОК> $TEST 002 3I
press 's' to stop...
ОК>
** (0,138): byte_err=0 bit_err=0 RSSI=-40,FEC=R:15,9I
** (1,138): byte_err=0 bit_err=0 RSSI=-41,FEC=R:15,9I
** (2,138): byte_err=0 bit_err=0 RSSI=-41,FEC=R:15,9Is
ОК>

```

## 16.41 \$DRG — ВЫВОД ВСЕХ ТЕХНОЛОГИЧЕСКИХ ПАРАМЕТРОВ

Ввод: **\$DRG**

## 16.42 \$RG — ЗАПИСЬ ТЕХНОЛОГИЧЕСКИХ ПАРАМЕТРОВ МОДУЛЯ

Команда существует в 3-х вариантах.

Ввод: **\$RGdd=hh** - запись в шестнадцатеричном формате  
**\$RGDdd=ddd** - запись в десятичном формате  
**\$RGBdd=bbbbbbbb** - запись в двоичном формате

где, **dd** - адрес регистра;

**hh**, **ddd** и **bbbbbbbb** - значение регистра в соответствующем формате.

После ввода, значение сразу записывается в ЭНОЗУ (выполнять команду \$S нет необходимости). Во избежание неправильной работы модуля без особой необходимости не следует изменять технологические параметры.

Технологические параметрах описаны в разделе «[Технологические параметры. Регистры RG](#)».

**Пример:**

```

$RG15=FE
$RGD15=254
$RGB15=11111110

```

Во всех 3-х случаях, в регистр 15 будет занесено число 254.

## 16.43 \$IRG — ИНИЦИАЛИЗАЦИЯ ТЕХНОЛОГИЧЕСКИХ ПАРАМЕТРОВ ПО УМОЛЧАНИЮ

Ввод: **\$IRG**

Значения по умолчанию указаны в разделе «[Технологические параметры. Регистры RG](#)».

Команду \$S после этого выполнять не нужно.

## 16.44 \$R — СБРОС ЛОКАЛЬНОГО МОДУЛЯ

Ввод: **\$R**

Аппаратный сброс (перезагрузка) локального модуля.

## 16.45 \$RST — СБРОС УДАЛЁННОГО МОДУЛЯ

Ввод: **\$RST hhh**, где **hhh** – адрес удалённого модуля.

При получении команды на сброс, удаленный модуль перезагрузится через 5 с. Команда выполняется в бесконечном цикле до момента получения ответа от удаленного модуля. Чтобы остановить цикл, необходимо напечатать символ 's'.

**Пример:**

```

$RST 002
press 's' to stop...
ОК>
** tx
** reset start
ОК>

```

## 16.46 \$S — ЗАПИСЬ ВНУТРЕННИХ ПЕРЕМЕННЫХ МОДУЛЯ В ЭНОЗУ

Ввод: **\$S**

По этой команде все ранее измененные параметры записываются в ЭНОЗУ и вступают в силу после сброса модуля (команда **\$R**).

## 16.47 \$E — ВЫХОД ИЗ КОМАНДНОГО РЕЖИМА РАБОТЫ

Ввод: **\$E**



Выполнение этой команды не активизирует изменений, проведенных в командном режиме. Для активации изменений необходимо выполнить команды **\$S** и **\$R**. Выйти из командного режима можно также по кнопке CFG.

## 16.48 \$LOG — ВЫВОД ЖУРНАЛА СОБЫТИЙ В ЭФИРЕ МОДУЛЯ

Ввод: **\$LOG**

Используется для диагностики работы модуля. По этой команде на последовательный порт выводится таблица с историей обмена пакетами в эфире. Модуль сохраняет последние 16 активные записи. Записи не сохраняются в ЭНОЗУ.

## 16.49 \$SCAN — СКАНИРОВАНИЕ ЭФИРА

Ввод: **\$SCAN**

По команде **\$SCAN** модуль в режиме реального времени выводит на порт все пакеты принимаемые из эфира, а также передаваемые в эфир (если командный режим в настоящий момент работает по отладочному порту). Для выхода из режима SCAN необходимо ввести символ 's'.

## 16.50 \$KEA/КЕН — УСТАНОВКА ПАРОЛЯ AES ШИФРОВАНИЯ

Команда существует в 2-х вариантах.

Ввод: **\$KEA=c1c2c3...c16** - установка пароля в символьном формате  
**\$KEN=hh1hh2hh3...hh16** - установка пароля в 16-ричном формате

где,

**c1...c16** - ASCII символы кроме пробела и запятой;

**hh1...hh16** - шестнадцатеричные числа от 00 до FF;

Пароль может содержать до 16 символов (**\$KEA**) или до 32 чисел (**\$KEN**). Оставшиеся не введенные значения (до 16 или 32) заполняются символами '-' (0x2D). Например, пароли "123" и "123---" идентичны. После задания пароля, в выводе команды **\$DMP** отображается подсказка \*KEY. С помощью обеих команд вводится один и тот же пароль - различается только лишь способ ввода.

Для удаления пароля необходимо ввести любую из двух команд без аргументов.

Пример:

```
OK> $KEA=hello
*** KEA:hello-----
OK> $KEN=1234ab
*** KEN:0x1234AB2D2D2D2D2D2D2D2D2D2D2D2D
OK> $KEA=
*** key deleted
OK>
```

## 16.51 \$SK — УСТАНОВКА ПАРОЛЯ XOR ШИФРОВАНИЯ

Ввод: **\$SK=hhhhhh** где,

**h** - шестнадцатеричное число от 0 до F;

Все 6 цифр должны быть введены. После задания пароля, в выводе команды **\$DMP** отображается подсказка \*SK. Для удаления пароля необходимо ввести пароль 000000.

Пример:

```
OK> $SK=1234AB
OK> $SK=000000
*** key deleted
OK>
```

**16.52 \$NAM — ЗАДАНИЕ ИМЕНИ МОДУЛЮ**

Ввод: **\$NAM=c1c2c3....c12** где,  
**c1....c12** - ASCII символы кроме пробела и запятой;

Имя модуля может содержать до 12 символов. Имя отображается при выводе результатов команд \$RFIND и в приветствии при входе в командный режим. Для удаления имени необходимо ввести команду \$NAM=.

**Пример:**

```
$NAM=Spektr868IO_Lora
*** NAM:Spektr868IO_
OK> $NAM=
*** NAM:
OK>
```

**16.53 \$CFL — УСТАНОВКА/СНЯТИЕ ПАРОЛЯ НА КОМАНДНЫЙ РЕЖИМ**

Ввод: **\$CFL=c1c2c3....c16** где,  
**c1....c16** - ASCII символы кроме пробела и запятой;

Пароль может содержать до 16 символов. Оставшиеся не введенные значения (до 16) заполняются символами '-' (0x2D). Например, пароли "123" и "123---" идентичны.

Для удаления пароля необходимо ввести команду без аргументов.

**Пример:**

```
OK> $CFL=123
*** CFL:123-----
OK> $CFL=
*** key deleted
OK>
```

**16.54 \$CFU — ВВОД ПАРОЛЯ НА КОМАНДНЫЙ РЕЖИМ**

Ввод: **\$CFU=c1c2c3....c16** где,  
**c1....c16** - ASCII символы кроме пробела и запятой;

**16.55 \$LOCK — РЕГИСТР БЛОКИРОВОК**

Ввод: **\$LOCK=bbbbbbbb**

Бит	Назначение	1	0
7-3	Резерв		
2	<b>bBootCmdLock</b> Блокировка конфигурирования модуля через режим загрузчика.	Да	Нет
1	<b>bEkranCfgLock</b> Блокировка конфигурирования модема через символьный индикатор. В модуле Спектр 868 IO Lora не имеет значение.	Да	Нет
0	<b>bRemoteCfgLock</b> Блокировка удалённого конфигурирования по эфиру	Да	Нет

**16.56 \$DMPLOCK — ВЫВОД РЕГИСТРА БЛОКИРОВОК**

Ввод: **\$DMPLOCK**

# 17 КОМАНДЫ УПРАВЛЕНИЯ МОДУЛЕМ ВВОДА ВЫВОДА (IO)

Команды для модуля IO начинаются с префикса "IO" и расположены в нижней части (секции) вывода команды \$DMP.

При описании команд приняты следующие сокращения:

**b** - двоичное число 0 или 1;

**d** - десятичное число от 0 до 9;

**h** - шестнадцатеричное число от 0 до 9 и от A до F.

**c** - символ. Определен в тексте описания команды.

Другие сокращения определены в тексте описания команды. При вводе значения, ведущие нули должны присутствовать.

После конфигурирования модуля, необходимо ввести команду \$S (сохранение параметров в ЭНЗУ) и выполнить перезагрузку.

CH=00 TXID=FFF COM=9600,8N1	CH=00 TXID=FFF COM=9600,8N1
PWR=0 MYID=001 DAT=EOT MAXP=0	PWR=0 MYID=002 DAT=EOT MAXP=0
AR=3 ACKT=016 EOT=0025 PLEN=000 EOC=0AT	AR=3 ACKT=016 EOT=0025 PLEN=000 EOC=0AT
BCN=1 RESPT=000 FEC=H:12,8I (MaxSz=158)	BCN=1 RESPT=000 FEC=H:12,8I (MaxSz=158)
BCT=005 RETRY=000 MDA=00000001 CONCT=000	BCT=005 RETRY=000 MDA=00000001 CONCT=000
IOADR=0A+IONDX=00 (0Ah) TFL=----- RPT=-	IOADR=0A+IONDX=00 (0Ah) TFL=----- RPT=-
IORPT=11111111 RFL=----- RTFL=-----	IORPT=11111111 RFL=----- RTFL=-----
MaxTt=715 ms,ACKT >=016	MaxTt=715 ms,ACKT >=016
IOSLAVE IOALR=F0	IO_MASTER
IOSFT=010 IOSFV=-----	IOSM0=1111 IOF15=-----
IOINL=----- IODBN=00000000h	IOSM1=1111 IOF02=-----210
IOINF=00 IOFR[1:0] =0000,0000	IOSM2=0000 IOQPT=11111111
IOUTM=----- IOUT[3:0]t=0000,0000,0000,0000	IOSM3=1111 IOSFT=010 IOSFV=----- IOALR=F0
IOINT=00000000	IOSM4=1111 IOINL=----- IODBN=00000000h
IOMOD=00000000	IOSM5=1111 IOINF=00 IOFR[1:0] =0000,0000
	IOSM6=1111 IOUTM=----- IOUT[3:0]t=0000,0000,0000,0000
	IOSM7=1111 IOINT=00000000
	IOMOD=00000000
	IO_MXO= M1_2_3_4 M1_2_3_4 M1_2_3_4 M1_2_3_4
	M1_2_3_4 M1_2_3_4 M1_2_3_4 M1_2_3_4



Вывод команды \$DMP отображает только те регистры в секции модуля I/O, которые влияют на его работу только в текущем режиме - IO\_Master, IO\_Slave или IO\_Trans.

## 17.1 \$IOIEE – инициализация модуля IO в начальное состояние

Ввод: \$IOIEE

После ввода все параметры модуля ввода/вывода устанавливаются в значения по умолчанию и вступают в силу после перезагрузки модуля (\$R)

## 17.2 \$IORIN – сброс значений входов

Ввод: \$IORIN

По этой команде удаляются запомненные значения входов удаленных модулей IO\_Slave у текущего модуля (если он когда то работал в режиме IO\_Master).

## 17.3 \$IOROUT – сброс значений выходов

Ввод: \$IOROUT

Сброс выходов.

## 17.4 \$IOMASTER – ввод модуля IO в режим IO\_MASTER

Ввод: \$IOMASTER

После ввода команды изменится нижняя часть вывода команды \$DMP.

## 17.5 \$IOSLAVE – ВВОД МОДУЛЯ IO В РЕЖИМ IO\_SLAVE

Ввод: \$IOSLAVE

После ввода команды изменится нижняя часть вывода команды \$DMP.

## 17.6 \$IOTRANS – ВВОД МОДУЛЯ IO В РЕЖИМ IO\_TRANS

Ввод: \$IOTRANS

После ввода команды изменится нижняя часть вывода команды \$DMP.

## 17.7 \$IOADR – БАЗОВЫЙ АДРЕС МОДУЛЯ

Ввод: \$IOADR=hh

## 17.8 \$IONDX – ИНДЕКС МОДУЛЯ

Ввод: \$IONDX=hh

После ввода команд \$IOADR и/или \$IONDX, адрес модуля равен IOADR+IONDX по модулю 256.

## 17.9 \$IORPT – ФИЛЬТР ДЛЯ РЕТРАНСЛЯЦИИ

Ввод: \$IORPT=bbbbbbbb

Регистр IORPT содержит битовую маску индексов модулей пакеты которых разрешены для ретрансляции в случае, если у текущего модуля активирован режим ретрансляции. Номер бита равен индексу. Старший бит (7) слева, младший бит (0) справа.

Пример: \$IORPT=00000011

Если у текущего модуля активирован режим ретрансляции, то modbus пакеты с адресами \$IOADR+0 и \$IOADR+1 подлежат ретрансляции, а modbus пакеты с адресами в диапазоне \$IOADR+2 ... \$IOADR+7 ретранслироваться не будут.

## 17.10 \$IOF02 – СПИСОК МОДУЛЕЙ IO\_SLAVE, ОПРАШИВАЕМЫХ ФУНКЦИЕЙ F02

Ввод: \$IOF02=dddddddd, где d - номера индексов от 0 до 7.

Регистр IOF02 содержит список индексов тех модулей IO\_Slave, для которых будет формироваться команда на чтение состояния входов (F02, Read Input Status). Индексы можно вводить в любом порядке. Отсутствие ввода списка обнуляет весь регистр сразу.

Пример: \$IOF02=107 IO\_Master будет опрашивать функцией F02 удаленные модули n7, n1 и n0.  
\$IOF02= Обнулить регистр

## 17.11 \$IOF23 – ПОДМЕНА КОМАНДЫ F02 НА F23

Ввод: \$IOF23=0 или \$IOF23=1

При вводе значения 0, по маске параметра \$IOF02 будет формироваться команда F02. При вводе значения 1, по маске параметра \$IOF02 будет формироваться команда F23.

По умолчанию, параметр IOF23=1

## 17.12 \$IOF15 – СПИСОК IO\_SLAVE МОДУЛЕЙ, ОПРАШИВАЕМЫХ ФУНКЦИЕЙ F15

Ввод: \$IOF15=bbbbbbbb, где d - номера индексов от 0 до 7.

Регистр IOF15 содержит список индексов тех модулей IO\_Slave, для которых будет формироваться команда на управление выходами (F15, Force Multiple Coils). Индексы можно вводить в любом порядке. Отсутствие ввода списка обнуляет весь регистр сразу.

Пример: \$IOF15=50 IO\_Master будет опрашивать функцией F15 удаленные модули с n5 и n0.  
\$IOF15= Обнулить регистр

## 17.13 \$IOF15BC – ШИРОКОВЕЩАТЕЛЬНЫЙ РЕЖИМ ФУНКЦИИ F15

Ввод: \$IOF15BC=0 или \$IOF15BC=1

При вводе значения 0, команда F15 передается адресно. При вводе значения 1, команда F15 передается с широковещательным адресом (сразу всем).

## 17.14 \$IOQPT – СПОСОБ ОПРОСА IO\_SLAVE УСТРОЙСТВ

Ввод: \$IOQPT=bbbbbbbb

Регистр **IOQPT** содержит битовую маску индексов тех модулей IO\_Slave, для которых будет формироваться запросы по радиоэффиру. Отсутствие значения 1 в маске соответствует опросу модуля IO\_Slave по последовательному порту. Номер бита равен индексу. Старший бит (7) слева, младший бит (0) справа. По умолчанию, параметр **IOQPT=11111111**

Пример: **\$IOQPT=11111010**

Модуль в режиме IO\_Master будет опрашивать модули IO\_Slave n2 и n0 по последовательному порту. Остальные по радиоэффиру.

## 17.15 \$IOPT – ПАУЗА МЕЖДУ ОПРОСАМИ

Ввод: **\$IOPT=dd**

Задаёт паузу в секундах после опроса модуля IO\_Slave. Значение от 0 до 15. Значения больше 15 соответствуют 15.

## 17.16 \$IOSM – МАСКИРОВАНИЕ ВЫХОДОВ

Ввод: **\$IOSMd=bbbb**, где

**d** – индекс модуля IO\_Slave (0...7,8); **bbbb** – маска выхода.

Конфигурация регистра **IOSM**:

Бит	Назначение
3	bOut4_Enable
2	bOut3_Enable
1	bOut2_Enable
0	bOut1_Enable

В режиме IO\_Master восемь регистров **IOSM0...IOSM7** позволяют маскировать выходы мастера от влияния входов удаленных модулей IO\_Slave при опросе удаленных модулей Modbus командой F02/F23 (чтение входов).

При вводе индекса 8, значение маски **bbbb** устанавливается сразу для всех индексов.

Пример: **\$IOSM0=0001** - при опросе мастером удаленного модуля IO\_Slave n0 состояние его входа IN1 будет транслироваться на выход OUT1 мастера, а остальные три входа (IN2...IN4) на выходы мастера транслироваться не будут.

**\$IOSM5=0110** - при опросе удаленного модуля IO\_Slave n5 состояние его входов IN3 и IN2 будет транслироваться на выходы OUT3 и OUT2 мастера, а входы IN4 и IN1 на выходы мастера транслироваться не будут.

## 17.17 \$IOSFT – ТАЙМАУТ ПРОПАДАНИЯ СВЯЗИ

Ввод: **\$IOSFT=ddd**, где **ddd** – число от 000 до 999.

Задаёт допустимое время отсутствия связи в секундах, после которого запускается механизм безопасного состояния выходов (если разрешено), а также срабатывает выход АВАРИЯ и загораются соответствующие индикаторы.

Пример: **\$IOSFT=010**

Отсутствием связи будет считаться неполучение запросов от мастера (для модуля в режиме IO\_Slave) или отсутствие ответов от удаленных модулей (для IO\_Master) в течение 10 секунд.

## 17.18 \$IOSFV – КОНФИГУРАЦИЯ ВЫХОДОВ В БЕЗОПАСНОМ СОСТОЯНИИ

Ввод: **\$IOSFV=cccc**

Каждый символ 'с' отвечает за соответствующий выход (самый левый символ отвечает за выход OUT4, самый правый - за OUT1) и может принимать следующие значения:

- безопасное состояние для данного выхода запрещено;

**0** при наступлении случая "безопасное состояние" реле на соответствующем выходе замкнется;

**1** при наступлении случая "безопасное состояние" реле разомкнется.

Пример: **\$IOSFV= - - 01**

Разрешено использование безопасного состояния для выходов OUT2 и OUT1. Безопасным состоянием для OUT2 является замыкание реле, а для OUT1 - размыкание.

## 17.19 \$IOALR – РЕЖИМ БЕЗОПАСНОГО СОСТОЯНИЯ ВЫХОДОВ

Ввод: \$IOALR=hh

Конфигурация регистра IOALR:

Бит	Назначение
7...4	ALARM_RELAY_MASK
3...0	ALARM_SLAVE_MODE

**ALARM\_RELAY\_MASK** - маска включенных индикаторов ALARM, по которой "включается" (размыкается) реле ALARM. Маска обрабатывается по операции логическое "ИЛИ". Действительна как в режиме IO\_Slave, так и в режиме IO\_Master.

**ALARM\_SLAVE\_MODE** - режим индикации состояния авария модуля в режиме IO\_Slave:

**0** - состояние "авария" означает отсутствие опроса со стороны IO\_Master. На индикации ALARM выставляется двоичный индекс модуля IO\_Slave плюс 1 (младший бит слева)

**1** - состояние "авария" означает отсутствие обновления выхода со стороны IO\_Master. На индикации ALARM отображается номер выхода, который не обновляется.

**Пример:** Режим IO\_Master:

**\$IOALR=50** (двоичный код : 0101\_0000)

В режиме авария реле ALARM **разомкнется** только в том случае, если авария наступит по причине отсутствия связи с модулями IO\_Slave с индексами **0 ИЛИ 2**. Если авария наступит по причине отсутствия связи с модулями 1 или 3, реле ALARM не изменит своего состояния.

Режим IO\_Slave:

**\$IOALR=31** (двоичный код : 0011\_0001)

В режиме авария реле ALARM **разомкнется** только в том случае, если авария наступит по причине не обновления выходов OUT1 или OUT2.

**\$IOALR=F0** (двоичный код : 1111\_0000)

В режиме авария реле ALARM **разомкнется** в случае отсутствия опроса от IO\_Master.

Индекс текущего модуля IO\_Slave не важен, т.к. маска установлена на все разряды.

## 17.20 \$IOMXO – МУЛЬТИПЛЕКСОР ВХОДОВ В РЕЖИМЕ IO\_MASTER

Ввод: \$IOMXO=n,S1InA S2InB S3InC S4InD, где

**n** - индекс модуля IO\_Slave от 0 до 7 для которого задано мультиплексирование;

**S1,S2,S3,S4** - индексы модулей IO\_Slave (от 0 до 7) или модуль IO\_Master (символ 'M') от которых задано мультиплексирование на выходы OUT1, OUT2, OUT3, OUT4 модуля n;

**InA, InB, InC, InD** - номера входов (от 1 до 4) соответствующих модулей, от которых задано мультиплексирование.

При вводе индекса n=8 мультиплекс задается сразу для всех модулей с индексами от 0 до 7.

**Пример:** **\$IOMXO=0,13 21 M2 62**

Для модуля IO\_Slave с индексом 0 задано следующее мультиплексирование:

На выход OUT1 транслируется вход IN3 модуля IO\_Slave с индексом 1

На выход OUT2 транслируется вход IN1 модуля IO\_Slave с индексом 2

На выход OUT3 транслируется вход IN2 модуля IO\_Master

На выход OUT4 транслируется вход IN2 модуля IO\_Slave с индексом 6.

По умолчанию, для каждого индекса IO\_Slave задано значение равное:

M1 M2 M3 M4 (отображение **M1\_2\_3\_4** в выводе команды \$DMP)

## 17.21 \$IODBN – СЧЕТЧИК ДРЕБЕЗГА ВХОДОВ

Ввод: \$IODBN=hhhhhhh

Первые 2 цифры (после знака '=') задают счетчик для входа IN4, следующие 2 - для входа IN3 и т.д. Максимальное значение для каждого счетчика 0x7F. Значение 0x00 означает отсутствие подавления дребезга для данного входа. Счетчик задает число 25 мс интервалов в течении которых, вход после изменения состояния не должен изменять свое значение.

**Пример:** **\$IODBN=7F501300**

Для входа IN4 задано 0x7F=127\*25=3.1с в течении которых он не должен менять свое значение после смены состояния;

Для входа IN3 задано 0x50=80\*25=2с;

Для входа IN2 задано 0x13=19\*25=0.48с;

Для входа IN1 функция подавления дребезга не задана.

## 17.22 \$IOINF – КОНФИГУРАЦИЯ ДЕТЕКТОРА ЧАСТОТЫ

Ввод: \$IOINF=bb

Первое значение (после знака '=') соответствует входу IN2, второе - IN1.

- 0 - мультиплексор детектора частоты для данного входа запрещен;
- 1 - мультиплексор детектора частоты для данного входа разрешен.

## 17.23 \$IOFR0/1 – ЗНАЧЕНИЕ КОМПАРАТОРА ДЕТЕКТОРА ЧАСТОТЫ

Ввод: \$IOFR0/1=hhhh

Пример: | \$IOFR0=0123 Для входа IN1 компаратор детектора частоты равен 291 Гц;  
| \$IOFR1=000A Для входа IN2 компаратор детектора частоты равен 10 Гц;

## 17.24 \$IOINL – ЗАЩЕЛКА ВХОДОВ

Ввод: \$IOINL=cccc

Каждый символ 'с' отвечает за соответствующий вход (самый левый за вход IN4, самый правый - за IN1) и может принимать следующие значения:

- - функция защелки для данного входа запрещена;
- 0 - режим защелки по уровню '0' (переход из '1' в '0');
- 1 - режим защелки по уровню '1' (переход из '0' в '1').

## 17.25 \$IOUTM – ГЕНЕРАЦИЯ ВЫХОДНОГО ИМПУЛЬСА

Ввод: \$IOUTM=cccc

Каждый символ 'с' отвечает за соответствующий выход (самый левый за выход OUT4, самый правый - за OUT1) и может принимать следующие значения:

- генерация импульса для данного выхода запрещена;
- 0 при получении команды на установку выхода в '0', будет сгенерирован **отрицательный** импульс;
- 1 при получении команды на установку выхода в '1', будет сгенерирован **положительный** импульс.

Пример: | \$IOUTM= - - 1 0

Для выхода OUT1 задан отрицательный импульс, для выхода OUT2 - положительный. Для выходов OUT4 и OUT3 генерация импульсов запрещена.

## 17.26 \$IOUTxT – ДЛИТЕЛЬНОСТЬ ВЫХОДНОГО ИМПУЛЬСА

Ввод: \$IOUTxT=hhhh, где x – номер выхода от 0 до 3.

Длительность задается кратно 100мс.

Пример: | \$IOUT3T= 0064 -для выхода OUT4 длительность импульса задана равной  
| 100 x 100мс = 10 с.

## 17.27 \$IOINT – ПЕРЕДАЧА В ЭФИР СОСТОЯНИЙ ВХОДОВ ПРИ ИЗМЕНЕНИИ СОСТОЯНИЯ ОДНОГО ИЗ

Ввод: \$IOINT=bbbbbbbb (в текущей реализации состояние регистра игнорируется.)

## 17.28 \$IOMOD – РЕЖИМ РАБОТЫ МОДУЛЯ ВВОДА/ВЫВОДА

Ввод: \$IOMOD=bbbbbbbb (в текущей реализации состояние регистра игнорируется.)



## 18 КАРТА РЕГИСТРОВ MODBUS

В следующих разделах приведены сообщения поддерживаемых функций. Под параметром "Slave Address" всегда подразумевается значение \$IOADR+\$IONDX. В описании значений приняты следующие обозначения:

- [hi] Старший байт 16 битного регистра (биты 15...8).
- [lo] Младший байт 16 битного регистра (биты 7...0).
- [A:B] Битовое поле значения в байте. Например, [3:0] - значение занимающее биты 3-0.

При записи/чтении регистров адреса которых лежат в не диапазона, выдается ответ "Exception" (функция 8Xh) с Exception Code = 0x01.

### 18.1 ЧТЕНИЕ ВХОДОВ

#### Функция 01 (Read Coil Status)

#### Функция 02 (Read Input Status)

Function	0x01/0x02
Starting Address Hi	0x00
Starting Address Lo	0x00...0x07. Входу IN1 соответствует адрес 0x00, Входу IN2 - адрес 0x01 и т.д. По адресам >= 0x04 считываются нули.
No. of Points Hi	0x00
No. of Points Lo	0x01...0x08

**Примечание:** Сумма Starting Address Lo + No. of Points Lo должна быть меньше 8

#### Функция 23 (Read/Write 4X Registers)

Function	0x17
Read Ref. Address Hi	0x00
Read Ref. Address Lo	0x01
Quantity to Read Hi	0x00
Quantity to Read Lo	0x01
Write Ref. Address Hi	0x00
Write Ref. Address Lo	0x02
Quantity to Write Hi	0x00
Quantity to Write Lo	0x01
Byte Count	0x02
Write Data 1 Hi	не важно
Write Data 1 Lo	[7:4] маска разрешения на установку выходов, [3:0] значение выходов в формате OUT4:OUT3:OUT2:OUT1.

**Ответ:**

Byte Count	0x02
Read Data 1 Hi	0x00
Read Data 1 Lo	[3:0] значение входов в формате IN4:IN3:IN2:IN1.

### 18.2 УСТАНОВКА ВЫХОДОВ

#### Функция 15 (Force Multiple Coils)

Function	0x0F
Coil Address Hi	0x00
Coil Address Lo	0x00...0x07. Выходу OUT1 соответствует адрес 0x00, выходу OUT2 - адрес 0x01 и т.д. По адресам >= 0x04 установка выходов не происходит.
Quantity of Coils Hi	0x00
Quantity of Coils Lo	0x01...0x08
Byte Count	0x01
Force Data	[3:0] значение выходов в формате OUT4:OUT3:OUT2:OUT1.

**Примечание:** Сумма Coil Address Lo + Quantity of Coils Lo должна быть меньше 8

#### Функция 23 (Read/Write 4X Registers)

См. значения в разделе "Чтение входов".

### 18.3 ЧТЕНИЕ ПРОФИЛЯ

#### Функция 03 (Read Holding Registers)

#### Функция 04 (Read Input Registers)

Function	0x03/0x04
Starting Address Hi	0x01 - база регистров основного профиля 0x02 - база регистров профиля модуля ввода/вывода
Starting Address Lo	0x00...0x1B - регистры основного профиля 0x00...0x1D - регистры профиля модуля ввода/вывода
No. of Points Hi	не важно
No. of Points Lo	число 16 битных регистров для чтения

#### Примечание:

По адресам 0x0100 и 0x0200 находится регистр версии модуля:

DEVICE\_ID[hi] + VERSION[lo] где,

DEVICE\_ID - тип устройства:

0x00 - радиомодуль СПЕКТР 868 IO Lora;

0x01 - радиомодем СПЕКТР 868 Lora.

VERSION - версия встроенного ПО.

[7:4] - основная цифра версии;

[3:0] - вспомогательная цифра версии;

*Значение регистров профиля выходят за рамки данного описания и доступны по запросу.*

### 18.4 ЗАПИСЬ ПРОФИЛЯ

#### Функция 16 (Preset Multiple Registers)

Function	0x10
Starting Address Hi	0x01 - база регистров основного профиля 0x02 - база регистров профиля модуля ввода/вывода
Starting Address Lo	0x01...0x1B - регистры основного профиля 0x01...0x1D - регистры профиля модуля ввода/вывода
No. of Registers Hi	не важно
No. of Registers Lo	число 16 битных регистров для записи
Byte Count	число байт (число 16 битных регистров x 2)
Data Hi (i)	Значение 16 битного регистра/регистров
Data Lo (i)	

#### Примечание:

По адресам 0x0100 и 0x0200 находится регистр версии модуля, поэтому запись по этому адресу не доступна.

*Значение регистров профиля выходят за рамки данного описания и доступны по запросу.*

#### Функция 06 (Preset Single Register)

Function	0x06
Reg. Address Hi	Аналогично функции 16
Reg. Address Lo	Аналогично функции 16
Preset Data Hi	Значение 16 битного регистра
Preset Data Lo	

### 18.5 ЧТЕНИЕ СОСТОЯНИЯ МОДУЛЯ

#### Функция 03 (Read Holding Registers)

#### Функция 04 (Read Input Registers)

Function	0x03/0x04
Starting Address Hi	0x00 - база регистров состояния
Starting Address Lo	0x00...0x0C - регистры состояния
No. of Points Hi	не важно
No. of Points Lo	число 16 битных регистров для чтения

**Регистры состояния:**

Адр.	Значение
0x00	Регистр версии. Аналогичен регистру 0x0100 и 0x0200 (см. раздел " <a href="#">Чтение профиля</a> ")
0x01	ALARMS[hi] + INPUT[lo] ALARMS - состояние индикаторов ALARM текущего модуля. Биты [3:0]. Бит 3 - самый правый индикатор, бит 0 - самый левый, INPUT - значение входов текущего модуля. Биты [3:0] - IN4:IN3:IN2:IN1.
0x02	SLAVE_OK[hi] + OUT[lo] SLAVE_OK - состояние удаленных модулей. 1 - доступен, 0 - нет связи или не опрашивается. Имеет смысл, если текущий модуль IO_Master. Бит 7 отвечает за модуль с индексом 7, бит 6 за модуль с индексом 6 и т.д. OUT - значение выходов текущего модуля. Биты [3:0] - OUT4:OUT3:OUT2:OUT1.
0x03	SLAVE_IN0[hi] + SLAVE_IN1[lo] SLAVE_IN0 - состояние входов удаленного модуля n0. Биты [3:0] - IN4:IN3:IN2:IN1 SLAVE_IN1 - состояние входов удаленного модуля n1. Биты [3:0] - IN4:IN3:IN2:IN1
0x04	SLAVE_IN2[hi] + SLAVE_IN3[lo] - состояние входов удаленных модулей n2 и n3
0x05	SLAVE_IN4[hi] + SLAVE_IN5[lo] - состояние входов удаленных модулей n4 и n5
0x06	SLAVE_IN6[hi] + SLAVE_IN7[lo] - состояние входов удаленных модулей n6 и n7
0x07	CNT_IN1_HI - старший байт счетчика на счетном входе IN1 текущего модуля.
0x08	CNT_IN1_LO - младший байт счетчика на счетном входе IN1 текущего модуля.
0x09	CNT_IN2_HI - старший байт счетчика на счетном входе IN2 текущего модуля.
0x0A	CNT_IN2_LO - младший байт счетчика на счетном входе IN2 текущего модуля.
0x0B	FREQ_IN1 - значение частоты на счетном входе IN1 текущего модуля.
0x0C	FREQ_IN2 - значение частоты на счетном входе IN2 текущего модуля.

## 18.6 ЗАПИСЬ СОСТОЯНИЯ МОДУЛЯ

**Функция 16 (Preset Multiple Registers)**

Function	0x10
Starting Address Hi	0x00 - база регистров состояния
Starting Address Lo	0x00...0x0C - регистры состояния
No. of Registers Hi	Не важно
No. of Registers Lo	число 16 битных регистров для записи
Byte Count	число байт (число 16 битных регистров x 2)
Data Hi (i)	Значение 16 битного регистра/регистров
Data Lo (i)	

**Функция 06 (Preset Single Register)**

Function	0x06
Reg. Address Hi	Аналогично функции 16
Reg. Address Lo	Аналогично функции 16
Preset Data Hi	Значение 16 битного регистра
Preset Data Lo	

**Примечание:**

Записывать можно любые регистры в диапазоне адресов 0x00...0x0C, однако реальное действие происходит только со следующими параметрами (см. раздел "[Чтение состояния модуля](#)"):

OUT[lo]	- изменение выходов, биты [3:0]
SLAVE_IN0... SLAVE_IN7	-
CNT_IN1_HI - CNT_IN1_LO	- изменение счетчика на счетном входе IN1
CNT_IN2_HI - CNT_IN2_LO	- изменение счетчика на счетном входе IN2

## 19 ОТЛАДОЧНЫЙ ПОРТ

В модуле присутствует отладочный порт (далее DEBUG), с помощью которого удобно отслеживать логику работы модуля в реальном времени. Отладочный порт становится противоположным активному (для передачи данных) порту - если активный порт RS485, отладочный USB и наоборот, если активный USB, отладочный - RS485. Выбор портов осуществляется по командам загрузчика "BOOT\_SET\_485/ BOOT\_SET\_USB" см. раздел "[Установка активного порта](#)".

На DEBUG порт выводится информация о текущей работе модуля на фиксированной скорости 57600 8N1 (терминальная программа должна быть настроена на эту скорость).


Также, как и по активному порту, через DEBUG порт можно войти в командный режим (передача 3-х символов "+"). Функциональность модуля в командном режиме на DEBUG порту не прерывается. Для анализа обмена пакетами по радиоэфиру можно использовать команды \$LOG (выводит последние 16 записи) или \$SCAN (вывода обмена в реальном времени).

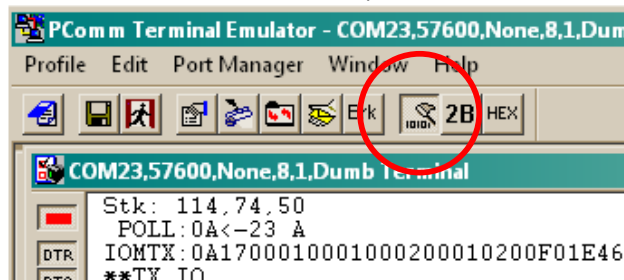
Для вывода и возможного анализа пользовательских данных (передаваемых в эфир или принятых из эфира) можно воспользоваться технологическим регистром dbg\_dump\_cfg (\$RG15) см. раздел "[Технологические параметры](#)". Конфигурация регистра выглядит следующим образом:

Биты	Значение
7-6	comTxDmp - печать данных передаваемых в эфир (полученных по активному порту): 00 - данные печатаются как есть (NAT). 01 - данные печатаются в шестнадцатеричном формате (HEX). 1x - данные не печатаются (DISable).
5-4	comRxDmp - печать данных принятых из эфира предназначенные для передачи в активный порт. Значение аналогично переменной comTxDmp;
3...0	Резерв

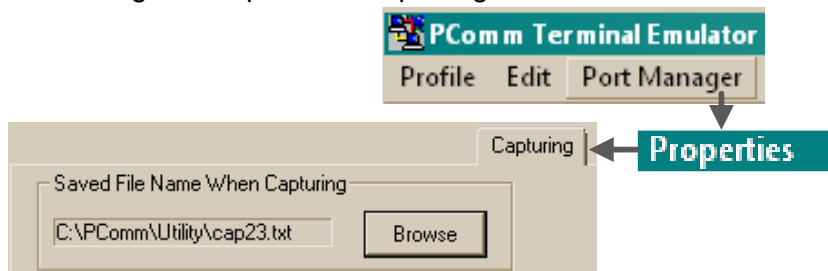
В случае возникновения ситуаций требующих анализа, через DEBUG порт можно записать лог работы модуля в реальном времени. Рассмотрим процесс с помощью терминальной программы Pcomm (<http://rateos.ru/files/PComm210.zip>).

Для начала необходимо подключиться к DEBUG порту. Если настроено все верно, в окне терминала должны непрерывно выдаваться *читаемые* сообщения (если ничего не выдается или отображается "мусор", проверить подключение и скорость).

В верхнем меню нажать на пиктограмму . При нажатии на пиктограмму вся информация записывается в файл CAPx.txt, где x - номер порта. При отжати - запись в файл прекращается.



Местоположение файла CAPx.txt можно узнать через меню PortManager->Properties->Capturing:



## 20 ВОПРОСЫ И ОТВЕТЫ

В данном разделе приведены часто задаваемые вопросы и пути их решения.

### ПЕРЕДАЧА ДАННЫХ ПО АКТИВНОМУ ПОРТУ

*При передаче данных мигает красным светодиод MODE.*

Несовпадение скоростей по последовательному порту у модуля и подключенного оборудования. Или же, если подключение осуществляется по RS485 - перепутаны линии А и В. Скорость активного порта модуля задается командой \$COM. Индикация светодиода MODE описана в разделе "Индикатор MODE".

*При передаче данных не загорается ни один светодиод.*

Оборудование подключено к модулю не по тому порту. При включении питания, если выбран активным порт USB, светодиод MODE кратковременно моргает 2 раза зеленым.

*При передаче данных обмен идет, оба модема моргают светодиодами RX/TX на прием и передачу, но подключенное оборудование не видит друг друга.*

Не настроен тайм-аут ожидания ответа у внешнего оборудования "мастер". По умолчанию, у всех систем работающих по проводу тайм-аут установлен в **минимальное** значение, т.к. временем распространения данных по проводной линии связи можно пренебречь - во внимание берется только скорость передачи и время реакции у оборудования "слэйв". Радиомодуль при передаче данных вносит существенные задержки, которые могут превышать "проводные" в десятки и иногда в сотни раз.

Необходимо корректировать тайм-аут ожидания ответа у внешнего оборудования "мастер". Минимальное значение не должно быть меньше, чем указано в подсказке "АСКТ=>" при выводе команды \$DMP.

*При передаче данных передающий модем моргает светодиодами RX/TX на прием и передачу, однако у принимающего модема ничего не моргает.*

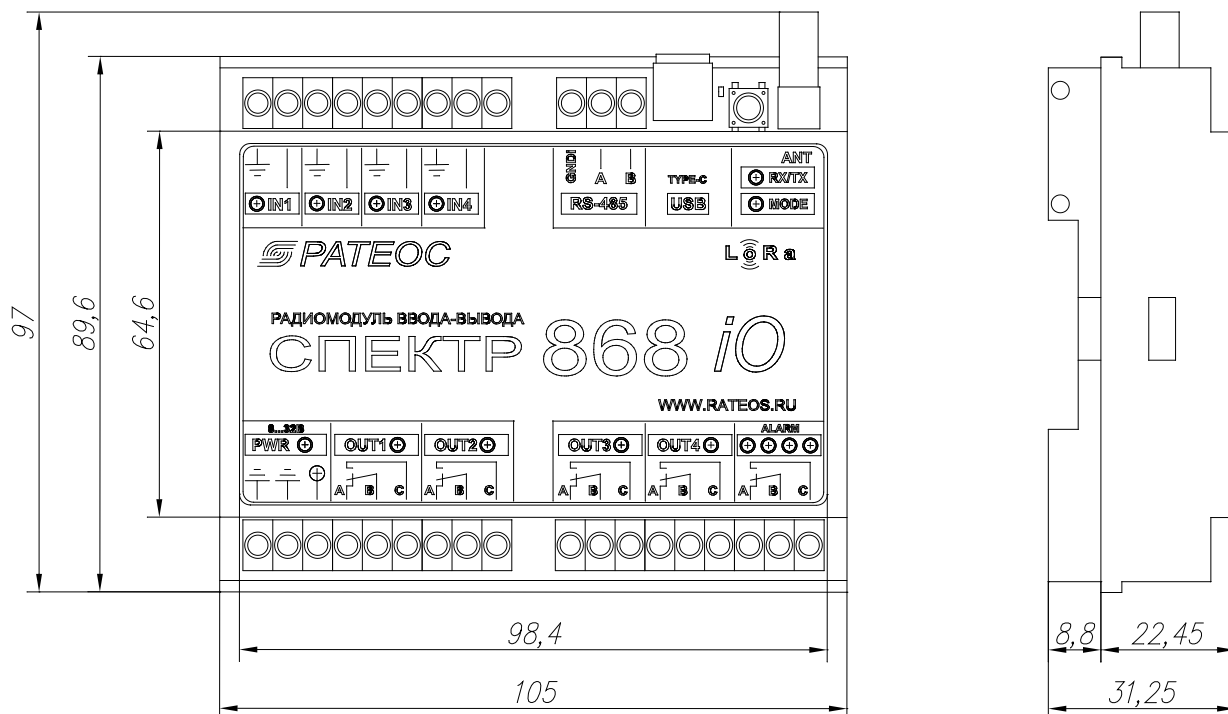
Скорее всего у принимающего модема перепутаны местами линии А и В т.к. подключенное оборудование не понимает запросы и, соответственно не отвечает.

### МОДУЛЬ ВВОДА/ВЫВОДА

*Вопрос*

Ответ

# 21 ГАБАРИТНЫЕ РАЗМЕРЫ



## 22 ОСНОВНЫЕ ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Диапазон рабочих температур:	от -40 до +50 °С
Напряжение питания:	+(8...32) В
Потребляемая мощность (не более):	0,5 Вт (средняя), 3 Вт (режим передачи)
Диапазон рабочих частот приемопередатчика:	868,765...869,140 МГц
Номинальная мощность передатчика:	25 мВт
Чувствительность приемника:	-118 дБм, при скорости передачи 4800 бит/с
Шаг установки рабочей частоты:	125 кГц
Разъем для внешней антенны:	SMA, 50 Ом
Телеметрические входы:	4 входа на замыкание с общей "землей"
Телеметрические выходы:	4 электромагнитных реле на переключение, ~240 В, 4 А
Выход "АВАРИЯ"	Электромагнитное реле на переключение, ~240 В, 4 А
Способ подключения внешних устройств	Винтовые клеммы (провод сечением 0,2...2,5 мм)
Протокол управления входами/выходами:	MODBUS RTU или MODBUS ASCII (автоопределение)
Интерфейс для прозрачной трансляции данных:	RS-485, С гальванической развязкой или USB Type-C, 1 200...115 200 бод
Интерфейс для конфигурации:	USB (разъем Type-C) или RS-485
Светодиодная индикация:	состояние входов (4 индикатора) состояние выходов (4 индикатора) состояние выхода "АВАРИЯ" (4 индикатора) наличие питания (1 индикатор) признак командного режима (1 индикатор) наличие данных в буфере RS-485 (1 индикатор) режим "ПЕРЕДАЧА" (1 индикатор)