

РАДИОМОДУЛЬ ВВОДА-ВЫВОДА
СПЕКТР 433 iO_{v2}

Руководство по эксплуатации

БАКП.464426.004 РЭ

Версия Руководства: **01.00**
Последнее изменение: 17.04.2024



ООО «РАТЕОС» Москва, Зеленоград
+7(499) 990-9716, www.rateos.ru, 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.

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

| | | |
|----------|--|-----------|
| 1 | ОБЩАЯ ИНФОРМАЦИЯ | 7 |
| 1.1 | Назначение и общие сведения | 7 |
| 1.2 | Отличие от предыдущих модулей Спектр 433 IO | 8 |
| 1.3 | Совместимость с предыдущими модулями Спектр 433 IO | 9 |
| 1.4 | Сопутствующее оборудование и материалы | 9 |
| 2 | РАЗЪЁМЫ, ОРГАНЫ УПРАВЛЕНИЯ И ИНДИКАЦИИ | 10 |
| 2.1 | Индикаторы | 10 |
| 2.1.1 | Индикатор <i>MODE</i> | 10 |
| 2.1.2 | Индикатор <i>RX/TX</i> | 11 |
| 2.2 | Разъёмы, клеммы, кнопка | 11 |
| 2.2.1 | Сброс модуля. Индикация версии ПО | 12 |
| 3 | УСТАНОВКА И ПОДКЛЮЧЕНИЕ ВНЕШНЕГО ОБОРУДОВАНИЯ | 13 |
| 3.1 | Установка модуля | 13 |
| 3.2 | Подключение антенн | 13 |
| 3.2.1 | Герметизация разъёмных соединений | 14 |
| 3.3 | Питание модуля | 14 |
| 3.4 | Интерфейсы RS-485, USB | 15 |
| 3.4.1 | Интерфейс RS-485 | 15 |
| 3.4.2 | Интерфейс USB. Установка драйвера шины USB на ПК | 15 |
| 3.4.3 | Индикация ошибок при работе по последовательным портам | 16 |
| 4 | НАСТРОЙКА МОДУЛЕЙ | 17 |
| 4.1 | Конфигурация параметров | 17 |
| 4.1.1 | Подключение к ПК для конфигурации | 17 |
| 4.2 | Работа модулей в качестве прозрачного радиоудлинителя RS-485/USB | 17 |
| 4.2.1 | Основные принципы | 17 |
| 4.2.2 | Параметры активного порта (\$COM) | 17 |
| 4.2.3 | Рабочий канал (\$CH) | 18 |
| 4.2.4 | Скорость обмена данными между модулями в эфире (\$AR) | 18 |
| 4.3 | Проверка модулей и стыковка с оборудованием перед установкой на объект | 19 |
| 5 | РЕЖИМЫ РАБОТЫ МОДУЛЯ | 20 |
| 5.1 | Режим загрузчика | 20 |
| 5.1.1 | Вход в командный режим по умолчанию (<i>BOOT_DEF</i>) | 20 |
| 5.1.2 | Режим Тест (<i>BOOT_TEST</i>) | 20 |
| 5.1.3 | Режим смены программного обеспечения (<i>BOOT_UPD_RS485/ BOOT_UPD_USB</i>) | 21 |
| 5.1.4 | Альтернативная конфигурация (<i>BOOT_CFG</i>) | 21 |
| 5.1.5 | Установка активного порта (<i>BOOT_SET_RS485/ BOOT_SET_USB</i>) | 23 |
| 5.1.6 | Включение режима "Маяк" (<i>BOOT_BEACON_OFF/ BOOT_BEACON_ON</i>) | 23 |
| 5.2 | Режим "КОМАНДНЫЙ" | 24 |
| 5.3 | Режимы передачи данных | 25 |
| 5.3.1 | Режим передачи данных «Прозрачный» EOT | 26 |
| 5.3.2 | Режим передачи данных «По символу» EOC | 27 |
| 5.3.3 | Режим передачи данных «Modbus» RTU | 27 |
| 5.3.4 | Режим передачи данных «DALLAS» DL8 | 27 |
| 5.3.5 | Конкатенация (склеивание) пакетов при передаче | 28 |
| 5.3.6 | Особенности режимов передачи данных | 28 |
| 5.3.7 | Буферизация принятых из эфира данных. Конкатенация данных при приеме | 29 |
| 5.4 | Режимы работы по эфиру | 30 |
| 5.4.1 | Адресация | 30 |
| 5.4.2 | Широковещательный режим | 31 |
| 5.4.3 | Адресный режим без подтверждения | 31 |
| 5.4.4 | Адресный режим с подтверждением | 31 |
| 5.4.5 | Особенности применения адресного режима с подтверждением | 32 |
| 5.5 | Ретрансляция пакетов | 33 |
| 5.6 | Алгоритм выхода в эфир (LBT) | 33 |
| 6 | ФОРМАТ ПАКЕТА В ЭФИРЕ | 35 |
| 6.1 | Помехоустойчивое кодирование | 35 |
| 6.2 | Перемежение | 35 |
| 6.3 | Мультипликация данных | 36 |

| | | |
|-----------|--|-----------|
| 6.4 | Рандомизация (скремблирование) | 37 |
| 6.5 | Оценка времени передачи | 37 |
| 7 | РАСШИРЕННЫЕ ВОЗМОЖНОСТИ ПРИЁМА И РЕТРАНСЛЯЦИИ | 40 |
| 7.1 | Ретрансляция с подменой адреса получателя | 42 |
| 7.2 | Приём ретранслированных пакетов..... | 42 |
| 7.3 | Игнорирование пакетов | 43 |
| 8 | ФИЛЬТРАЦИЯ ДАННЫХ | 45 |
| 8.1 | Фильтрация исходящих данных | 46 |
| 8.2 | Фильтрация входящих данных | 46 |
| 8.3 | Фильтрация данных при ретрансляции | 46 |
| 9 | ПРИМЕРЫ ОРГАНИЗАЦИИ СЕТЕЙ..... | 47 |
| 9.1 | адресный режим | 47 |
| 9.1.1 | адресный режим с подтверждением. | 47 |
| 9.1.2 | Адресный режим без подтверждения..... | 47 |
| 9.1.3 | Адресный режим без подтверждения с повторителем. | 47 |
| 9.2 | Режим «ШИРОКОВЕЩАТЕЛЬНЫЙ» | 48 |
| 9.2.1 | Широковещательная сеть без гарантии доставки сообщений | 48 |
| 9.2.2 | Широковещательная сеть с гарантией доставки сообщений | 48 |
| 9.2.3 | Широковещательная сеть с повторителем..... | 48 |
| 10 | ШИФРОВАНИЕ ДАННЫХ..... | 49 |
| 10.1 | Шифрование AES..... | 49 |
| 10.2 | Шифрование XOR..... | 49 |
| 11 | МОДУЛЬ ВВОДА ВЫВОДА (IO)..... | 50 |
| 11.1 | Общие сведения | 50 |
| 11.2 | Входы IN1...IN4..... | 50 |
| 11.3 | Выходы OUT1...OUT4 | 51 |
| 11.4 | Тест входов/выходов | 51 |
| 11.5 | Клемма 1WIRE | 51 |
| 11.6 | Адресация модулей | 52 |
| 11.7 | Режим IO_Slave..... | 52 |
| 11.8 | Режим IO_Master..... | 53 |
| 11.9 | Режим IO_Master. Мультиплексирование входов на выходы | 54 |
| 11.10 | Режим IO_Trans..... | 55 |
| 11.11 | Безопасное состояние выходов | 56 |
| 11.11.1 | Индикаторы и реле ALARM. | 56 |
| 11.11.2 | Безопасное состояние в режиме IO_Slave | 58 |
| 11.11.3 | Безопасное состояние в режиме IO_Master | 58 |
| 11.12 | Функции входов | 58 |
| 11.12.1 | "Дребезг" контакта..... | 59 |
| 11.12.2 | Детектор частоты | 59 |
| 11.12.3 | Защелка..... | 59 |
| 11.13 | Функции выходов | 61 |
| 11.13.1 | Выходной импульс..... | 61 |
| 11.14 | Опрос по RS485 | 61 |
| 11.15 | Фильтр для ретрансляции \$IORPT | 62 |
| 11.16 | Типовые схемы включения..... | 63 |
| 11.16.1 | Трансляция 4-х сухих контактов в обе стороны (точка-точка)..... | 63 |
| 11.16.2 | Трансляция 4-х входов на 4 удаленных выхода. Трансляция 4-х удаленных входов на выходы. | 63 |
| 11.16.3 | Трансляция 4-х входов на выходы произвольного количества модулей IO_SLAVE | 64 |
| 11.16.4 | Трансляция на выходы одного модуля входов нескольких удаленных модулей | 65 |
| 11.16.5 | Мультиплексирование выходов | 65 |
| 11.16.6 | Управление модулями IO_SLAVE внешним "мастером" | 66 |
| 12 | УДАЛЕННОЕ КОНФИГУРИРОВАНИЕ | 67 |
| 12.1 | Удалённое конфигурирование в командном режиме..... | 67 |
| 12.2 | Удалённое конфигурирование по modbus | 68 |
| 13 | ТЕСТИРОВАНИЕ СВЯЗИ..... | 69 |
| 13.1 | Тестирования связи в командном режиме | 69 |
| 13.2 | Тестирования связи в прозрачном режиме | 69 |
| 13.3 | Режим Маяк..... | 70 |

| | | |
|-----------|--|-----------|
| 14 | ТЕХНОЛОГИЧЕСКИЕ ПАРАМЕТРЫ. РЕГИСТРЫ RG | 71 |
| 15 | КОМАНДЫ УПРАВЛЕНИЯ МОДУЛЕМ | 72 |
| 15.1 | \$DMP(R) — вывод профиля (удалённого) модуля | 72 |
| 15.2 | \$IEE— сброс параметров по умолчанию | 72 |
| 15.3 | \$CH — изменение рабочего частотного канала | 72 |
| 15.4 | \$PWR — установка мощности передатчика | 72 |
| 15.5 | \$AR — скорость передачи данных по эфиру | 73 |
| 15.6 | \$MN —число бит, передаваемых без инверсии последнего бита | 73 |
| 15.7 | \$LBT —алгоритм выхода в эфир | 73 |
| 15.8 | \$P2/P1 — выбор протокола при передаче | 73 |
| 15.9 | \$MYID — изменение собственного адреса модуля | 73 |
| 15.10 | \$TXID — изменение адреса вызываемого модуля | 73 |
| 15.11 | \$WA —ожидание подтверждения в адресном режиме | 73 |
| 15.12 | \$TRY — число ретрансляций пакетов, при отсутствия подтверждения | 74 |
| 15.13 | \$COPY— максимальное число одинаковых широкоэвещательных пакетов | 74 |
| 15.14 | \$PAUSE — время между передачами копий широкоэвещательных пакетов | 74 |
| 15.15 | \$FEC — тип FEC для информационных пакетов в эфире | 74 |
| 15.16 | \$DM — мультипликация данных | 74 |
| 15.17 | \$COM — параметры последовательного порта | 74 |
| 15.18 | \$DAT — тип протокола входящих данных | 74 |
| 15.19 | \$PL — максимальное число пакетов в буфере | 75 |
| 15.20 | \$PACT — тайм-аут приема пакета в буфер на передачу | 75 |
| 15.21 | \$PSL — нарезка входящих данных | 75 |
| 15.22 | \$EOC — символ передачи данных | 75 |
| 15.23 | \$RCONC — конкатенация на прием | 75 |
| 15.24 | \$TCONC — конкатенации на передачу | 75 |
| 15.25 | \$RT — номер повторителя модуля | 75 |
| 15.26 | \$RID — адрес пакета разрешенного для ретрансляции | 75 |
| 15.27 | \$PID — адрес пакета для расширенного приема | 76 |
| 15.28 | \$LID / \$XID — вывод / удаление адресов для ретрансляции и расширенного приема .. | 76 |
| 15.29 | \$QLID — проверка списка RID/PID | 77 |
| 15.30 | \$xFL — способ фильтрации для данных и ретрансляции | 77 |
| 15.31 | \$xFLDMP — вывод значений фильтров | 77 |
| 15.32 | \$xFL+ — включение диапазона значений в фильтр | 77 |
| 15.33 | \$xFL- — исключение диапазона значений из фильтра | 77 |
| 15.34 | \$FLS — запись значений фильтров в энозу | 78 |
| 15.35 | \$RPRF — чтение профиля удалённого модуля | 78 |
| 15.36 | \$WPRF — запись профиля удалённого модуля | 78 |
| 15.37 | \$EPS — начать редактирование удалённого профиля | 78 |
| 15.38 | \$EPE — остановить редактирование профиля удалённого модуля | 78 |
| 15.39 | \$UPD - смена программного обеспечения | 78 |
| 15.40 | \$RCHK — опрос удалённого модуля | 78 |
| 15.41 | \$RFIND — поиск устройств в эфире | 78 |
| 15.42 | \$TEST — запрос тестовых пакетов | 78 |
| 15.43 | \$DRG — вывод всех технологических параметров | 79 |
| 15.44 | \$RG — запись технологических параметров модуля | 79 |
| 15.45 | \$IRG — инициализация технологических параметров по умолчанию | 79 |
| 15.46 | \$EXCHP — смена порта | 79 |
| 15.47 | \$R — сброс локального модуля | 79 |
| 15.48 | \$RST— сброс удалённого модуля | 79 |
| 15.49 | \$S — запись внутренних переменных модуля в ЭНОЗУ | 79 |
| 15.50 | \$E — выход из командного режима работы | 80 |
| 15.51 | \$LOG — вывод журнала событий в эфире модуля | 80 |
| 15.52 | \$SCAN — сканирование эфира | 80 |
| 15.53 | \$KEA/KEH — установка пароля AES шифрования | 80 |
| 15.54 | \$SK — установка пароля XOR шифрования | 80 |
| 15.55 | \$NAM — задание имени модулю | 80 |
| 15.56 | \$CFL— установка/снятие пароля на командный режим | 81 |

| | | |
|-----------|--|-----------|
| 15.57 | \$CFU— ввод пароля на командный режим | 81 |
| 15.58 | \$LOCK — регистр блокировок | 81 |
| 15.59 | \$DMPLOCK — вывод регистра блокировок | 81 |
| 16 | КОМАНДЫ УПРАВЛЕНИЯ МОДУЛЕМ ВВОДА ВЫВОДА (IO) | 82 |
| 16.1 | \$IOIEE – инициализация модуля IO в начальное состояние | 82 |
| 16.2 | \$IORIN – сброс значений входов | 82 |
| 16.3 | \$IOROUT – сброс значений выходов | 82 |
| 16.4 | \$IOMASTER – ввод модуля IO в режим IO_MASTER | 82 |
| 16.5 | \$IOSLAVE – ввод модуля IO в режим IO_SLAVE | 83 |
| 16.6 | \$IOTRANS– ввод модуля IO в режим IO_TRANS | 83 |
| 16.7 | \$IOADR – базовый адрес модуля | 83 |
| 16.8 | \$IONDX – индекс модуля | 83 |
| 16.9 | \$IORPT – фильтр для ретрансляции | 83 |
| 16.10 | \$IOF02 – список модулей IO_Slave, опрашиваемых функцией F02 | 83 |
| 16.11 | \$IOF23 – подмена команды F02 на F23 | 83 |
| 16.12 | \$IOF15 – список IO_Slave модулей, опрашиваемых функцией F15 | 83 |
| 16.13 | \$IOF15BC – широкопередаточный режим функции F15 | 83 |
| 16.14 | \$IOQPT – способ опроса IO_Slave устройств | 83 |
| 16.15 | \$IOPT – пауза между опросами | 84 |
| 16.16 | \$IOWA – время ожидания ответа от IO_Slave устройств | 84 |
| 16.17 | \$IOSM – маскирование выходов | 84 |
| 16.18 | \$IOSFT – таймаут пропадания связи | 84 |
| 16.19 | \$IOSFV – конфигурация выходов в безопасном состоянии | 84 |
| 16.20 | \$IOALR – режим безопасного состояния выходов | 85 |
| 16.21 | \$IOMXO – мультиплексор входов в режиме IO_Master | 85 |
| 16.22 | \$IODBN – счетчик дребезга входов | 85 |
| 16.23 | \$IOINF – конфигурация детектора частоты | 86 |
| 16.24 | \$IOFR0/1 – значение компаратора детектора частоты | 86 |
| 16.25 | \$IOINL – защелка входов | 86 |
| 16.26 | \$IOUTM – генерация выходного импульса | 86 |
| 16.27 | \$IOUTxT – длительность выходного импульса | 86 |
| 16.28 | \$IOINT – передача в эфир состояний входов при изменении состояния одного из | 86 |
| 16.29 | \$IOMOD – режим работы модуля ввода/вывода | 86 |
| 17 | КАРТА РЕГИСТРОВ MODBUS | 87 |
| 17.1 | Чтение входов | 87 |
| 17.2 | Установка выходов | 87 |
| 17.3 | Чтение профиля | 88 |
| 17.4 | Запись профиля | 88 |
| 17.5 | Чтение состояния модуля | 88 |
| 17.6 | Запись состояния модуля | 89 |
| 18 | ОТЛАДОЧНЫЙ ПОРТ | 90 |
| 19 | ВОПРОСЫ И ОТВЕТЫ | 92 |
| 20 | ГАБАРИТНЫЕ РАЗМЕРЫ | 93 |
| 21 | ОСНОВНЫЕ ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ | 94 |

1 ОБЩАЯ ИНФОРМАЦИЯ

1.1 НАЗНАЧЕНИЕ И ОБЩИЕ СВЕДЕНИЯ

Радиомодуль «СПЕКТР 433 IO V2» БАП.464426.004 (далее по тексту — модуль) представляет собой устройство для приёма/передачи данных по радиоканалу со скоростью 4800,9600,19200,38400,57600 и 76800 бод в диапазоне частот 433 МГц при выходной мощности 10 мВт, что позволяет использовать его без получения разрешений органов ГосСвязьНадзора.

Модуль предназначен для использования в системах удаленного управления и/или сбора данных в качестве окончательного 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.

Модуль осуществляет все необходимые функции для обмена данными по радиоканалу:

- управление встроенным приёмопередатчиком диапазона 433 МГц;
- 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.

1.2 ОТЛИЧИЕ ОТ ПРЕДЫДУЩИХ МОДУЛЕЙ СПЕКТР 433 IO

Радиомодуль «СПЕКТР 433 IO V2» представляет собой усовершенствованную версию модуля "Спектр 433 IO". Далее перечислены основные нововведения и отличия.

Аппаратный уровень

- применение мощного процессора семейства ARM, что позволило реализовать новые функциональные возможности, а также, возможность в будущем адаптировать алгоритмы под конкретные задачи за минимальное время;
- введение 2-х дополнительных скоростей в эфире (57600 и 76800) без каких либо ограничений в конфигурации модуля;
- применение нового приемопередатчика с лучшей чувствительностью и выходной мощностью до 550 мВт;
- встроенный гальванически изолированный интерфейс RS485 для передачи пользовательских данных *по умолчанию*.

Протокол V2 в эфире

- проверочная сумма заголовка пакета (CRC) увеличена с 8 до 16 бит, что позволяет уменьшить вероятность выделения из "шума" радиоэфира пакетов, которые такими не являются;
- добавлен адресный (точка-точка) режим без подтверждения;
- реализовано шифрование данных с помощью алгоритма AES128;
- размер пакета в эфире увеличен с 256 до 512 байт;
- мультипликация (дублирование) данных внутри пакета;
- способность передавать информационные блоки в пакете в обратном порядке, в случае передачи копий одного и того же пакета (для увеличения вероятности приема).

Модуль передачи данных

- буферизация пользовательских данных в модуле передачи данных именно в том виде, в котором они поступают по последовательному порту. Например, если в буфер модема пока он "занят" передачей данных в эфир поступают пакеты 7, 20 и 15 байт, то после окончания текущей передачи, в эфир будут переданы именно пакеты размером 7, 20 и 15 байт. В предыдущей реализации, в данном случае в эфир мог передаться 1 пакет размером 42 байт ("склейка из 3 пакетов");
- Гибкое управление буферизацией пользовательских данных на уровне пакетов;
- Фильтры на передачу, ретрансляцию и прием.

Модуль ввода/вывода (I/O)

- Способность модуля в режиме IO_Master опрашивать (устанавливать выходы и считывать входы) модуль IO_Slave всего за 2 пакета в эфире (ранее, это занимало 4 пакета);
- мультиплексирование (ретрансляция) входов одного или нескольких модулей IO_Slave на выходы одного или нескольких *других* модулей IO_Slave без использования входов/выходов модуля IO_Master (ранее, это было невозможно);
- программирование входов на дребезг контактов, режима "защелки" и счетчика или детектора частоты;
- опрос модулей IO_Slave по интерфейсу RS485;
- добавлен интерфейс 1-WIRE.

Разное

- адаптивный вывод параметров в командном режиме - отображение только тех параметров, которые участвуют в текущей активной конфигурации работы;
- наличие отладочного порта.

1.3 СОВМЕСТИМОСТЬ С ПРЕДЫДУЩИМИ МОДУЛЯМИ СПЕКТР 433 IO

В версии модуля V2 присутствует поддержка совместимости по протоколу в эфире с предыдущими модулями версии V1. Модуль V2 на прием по умолчанию "распознает" пакеты из эфира как протокола V1 так и V2. Выбор протокола на передачу выбирается командой \$P1/\$P2. По умолчанию, модуль «СПЕКТР 433 IO V2» настроен на работу в протоколе V2 (\$P2).

Для совместной работы модулей «СПЕКТР 433 IO» и «СПЕКТР 433 IO V2» необходимо выбирать скорости в эфире от 4800 до 38400. Если в системе используются только модули версии V2, то доступен весь ряд скоростей в эфире.

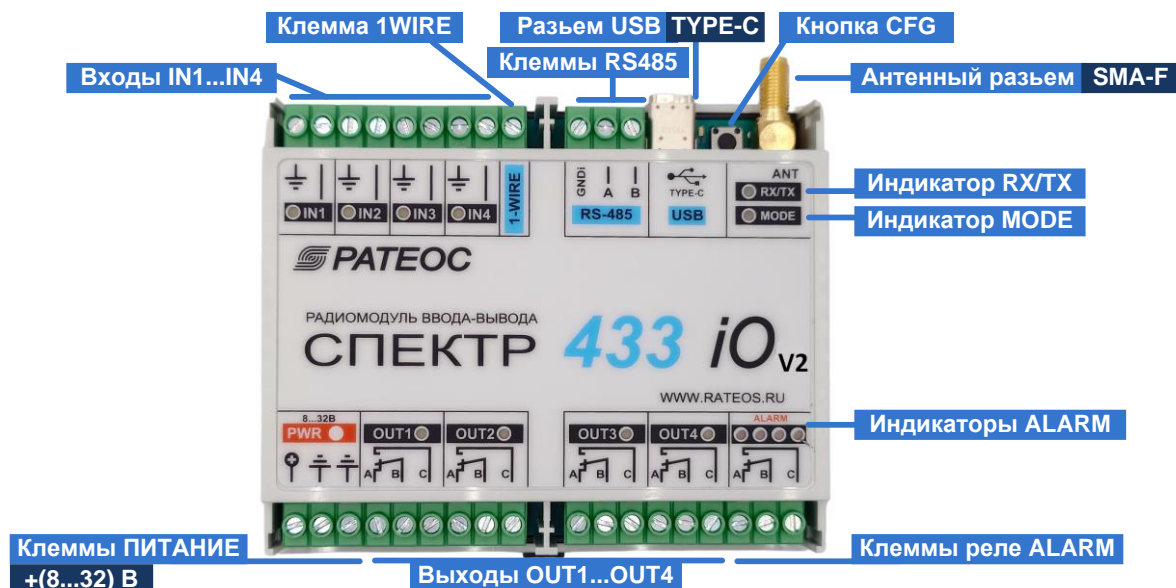
Если предполагается использовать модуль в качестве передачи сухих контактов (I/O) то следует иметь ввиду, что модуль «СПЕКТР 433 IO V2» в режиме Master не будет работать с модулями «СПЕКТР 433 IO» настроенные на режим Slave. Таким образом, если нужна передача только сухих контактов "смешивать" в одной системе модули разных версий **не рекомендуется**.

1.4 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);
- UTP 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;

2 РАЗЪЁМЫ, ОРГАНЫ УПРАВЛЕНИЯ И ИНДИКАЦИИ



2.1 ИНДИКАТОРЫ

2.1.1 ИНДИКАТОР MODE

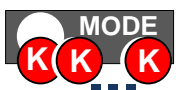
В нормальном режиме (режим передачи данных), двухцветный индикатор MODE показывает следующие состояния:



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 [Red bar] ...

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 [Red bar] ...

Входной буфер модуля переполнен - пропускная способность модуля по эфиру ниже, чем плотность данных, подаваемых в модуль. Новые данные потеряны.

Мигает красным [Red bar] ...

Ошибка режима входных данных. Формат данных не соответствует выбранному режиму (параметр \$DAT).



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



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"

2.1.2 Индикатор RX/TX

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



Передача данных в эфир (**К**расный)



Прием данных из эфира (**З**еленый)



Нет активности в эфире (не горит)

В остальных случаях используется как вспомогательный индикатор режимов.

2.2 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

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.

2.2.1 СБРОС МОДУЛЯ. ИНДИКАЦИЯ ВЕРСИИ ПО

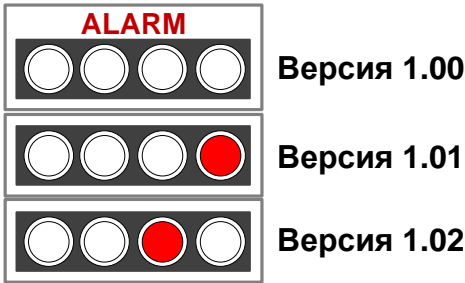
С помощью кнопки CFG можно осуществить сброс модуля не переключая питания. Это может быть удобно, например, для перехода в режим загрузчика в уже смонтированном модуле.

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:



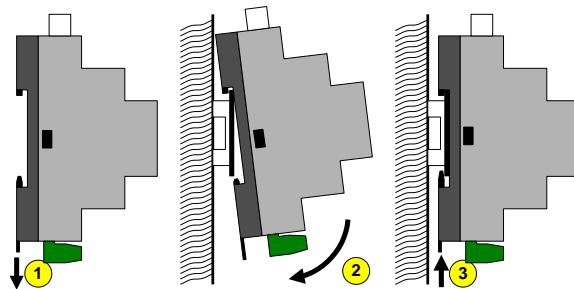
3 УСТАНОВКА И ПОДКЛЮЧЕНИЕ ВНЕШНЕГО ОБОРУДОВАНИЯ

3.1 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.

3.2 ПОДКЛЮЧЕНИЕ АНТЕНН

Совместно с модулем применяются различные антенны диапазона 433 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.

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 7...10 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).

3.2.1 ГЕРМЕТИЗАЦИЯ РАЗЪЁМНЫХ СОЕДИНЕНИЙ

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 (термофит) без клеевого слоя — она не обеспечит требуемой герметичности. Ее можно применить только вместо ПВХ изоленты поверх слоя мастики или самовулканизирующейся ленты.

Обычную изоленту — со временем под действием окружающей среды клеевой слой теряет свои свойства и слои изоленты начинают отслаиваться.

Ацетатные герметики — они «разъедают» цветные металлы и разъём не прослужит и несколько месяцев.

3.3 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 | 10 мВт | 1 Вт | 70 мА | 40 мА |
| \$PWR=1 | 150 мВт | 1,3 Вт | 110 мА | 55 мА |
| \$PWR=2 | 350 мВт | 1,9 Вт | 160 мА | 80 мА |
| \$PWR=3 | 550 мВт | 2,4 Вт | 200 мА | 100 мА |

3.4 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.



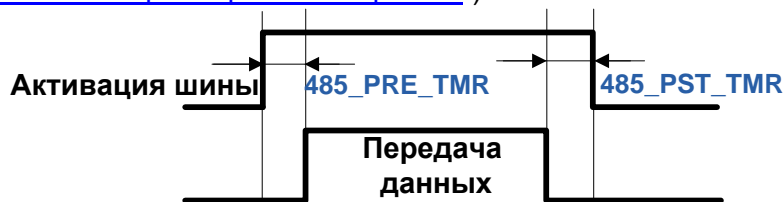
- Выбор интерфейса для передачи данных производится с помощью команд загрузчика BOOT_SET_485/ BOOT_SET_USB.
- Конфигурация возможна по любому порту.

3.4.1 ИНТЕРФЕЙС RS-485

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

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**.

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

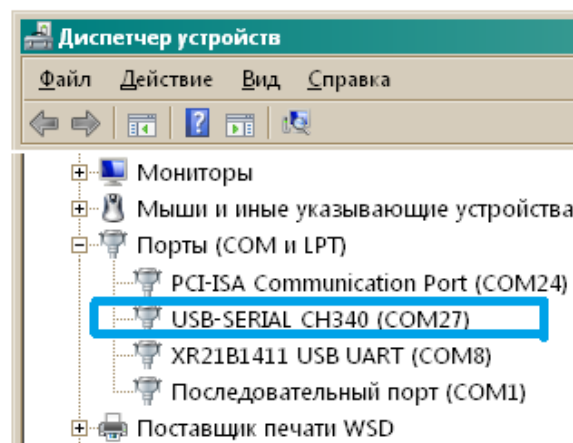
3.4.2 ИНТЕРФЕЙС USB. УСТАНОВКА ДРАЙВЕРА ШИНЫ USB НА ПК

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.



3.4.3 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](#)").

4 НАСТРОЙКА МОДУЛЕЙ

4.1 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 некоторых параметров. Параметры хранятся в энергонезависимой памяти (ЭНОЗУ/EEPROM) модуля.

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



- широковещательная адресация (\$TXID=FFFF \$COPY=1)
- активный порт RS485. Скорость 9600,8N1
- Частотный канал 433,920 КГц (\$CH=92)
- Скорость в эфире 9600 бит/сек (\$AR=3). Время ожидания ответа для внешнего оборудования при данной скорости должно быть ~2 с.
- Мощность в эфире 10 мВт (\$PWR=0)
- конфигурация по USB (настройки порта 57600,8N1)

4.1.1 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. Установка параметров осуществляется подачей текстовых команд. Подробная информация описана в разделе "[Режим "КОМАНДНЫЙ"](#)".

4.2 РАБОТА МОДУЛЕЙ В КАЧЕСТВЕ ПРОЗРАЧНОГО РАДИОУДЛИНИТЕЛЯ RS-485/USB

4.2.1 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-485/232, 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.



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.

4.2.2 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.

Параметры активного порта устанавливаются командой \$COM.

4.2.3 РАБОЧИЙ КАНАЛ (\$CH)

Все модемы в одной системе должны работать на одной частоте. В диапазоне частот 433 МГц используется стандартная сетка частот из 69 каналов с шагом 25 кГц :

| | | | | | | | | | | | | | |
|----|---------|----|---------|----|---------|----|---------|----|---------|----|---------|----|---------|
| 01 | 433.075 | 11 | 433.325 | 21 | 433.575 | 31 | 433.825 | 41 | 434.075 | 51 | 434.325 | 61 | 434.575 |
| 02 | 433.100 | 12 | 433.350 | 22 | 433.600 | 32 | 433.850 | 42 | 434.100 | 52 | 434.350 | 62 | 434.600 |
| 03 | 433.125 | 13 | 433.375 | 23 | 433.625 | 33 | 433.875 | 43 | 434.125 | 53 | 434.375 | 63 | 434.625 |
| 04 | 433.150 | 14 | 433.400 | 24 | 433.650 | 34 | 433.900 | 44 | 434.150 | 54 | 434.400 | 64 | 434.650 |
| 05 | 433.175 | 15 | 433.425 | 25 | 433.675 | 35 | 433.925 | 45 | 434.175 | 55 | 434.425 | 65 | 434.675 |
| 06 | 433.200 | 16 | 433.450 | 26 | 433.700 | 36 | 433.950 | 46 | 434.200 | 56 | 434.450 | 66 | 434.700 |
| 07 | 433.225 | 17 | 433.475 | 27 | 433.725 | 37 | 433.975 | 47 | 434.225 | 57 | 434.475 | 67 | 434.725 |
| 08 | 433.250 | 18 | 433.500 | 28 | 433.750 | 38 | 434.000 | 48 | 434.250 | 58 | 434.500 | 68 | 434.750 |
| 09 | 433.275 | 19 | 433.525 | 29 | 433.775 | 39 | 434.025 | 49 | 434.275 | 59 | 434.525 | 69 | 434.775 |
| 10 | 433.300 | 20 | 433.550 | 30 | 433.800 | 40 | 434.050 | 50 | 434.300 | 60 | 434.550 | 92 | 433.920 |

Рабочий канал выбирают так, чтобы уйти от помех, вызванных работой другого оборудования в радиоэфире (работать на свободном канале). Как правило, оборудование диапазона LPD работает в его центре, поэтому для уменьшения вероятности помех от него в качестве установки по умолчанию в модемах задан рабочим канал 92 (частота 433,920 МГц) в центре диапазона.

Возможность выбора рабочего канала позволяет развернуть на одной территории несколько независимых систем (групп модемов): для этого нужно просто «разнести» эти группы на разные каналы, чтобы они не мешали друг другу. Разнос каналов при этом должен быть достаточным для обеспечения отсутствия влияния групп друг на друга.

Необходимый разнос каналов зависит от установленной в модемах скорости работы в радиоэфире (параметр Ar, см. далее): чем выше скорость, тем дальше друг от друга следует разносить каналы:

| Скорость в эфире | Минимальный разнос частот | Минимальный разнос каналов |
|---------------------|---------------------------|----------------------------|
| Ar 1 (4 800 бит/с) | 50 кГц | 2 |
| Ar 3 (9 600 бит/с) | 90 кГц | 4 |
| Ar 5 (19 200 бит/с) | 150 кГц | 6 |
| Ar 7 (38 400 бит/с) | 240 кГц | 10 |
| Ar 8 (57 600 бит/с) | 260 кГц | 11 |
| Ar 9 (76 800 бит/с) | 280 кГц | 12 |

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

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

- Если данные через модулями 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 | Девияция, КГц | Ширина полосы, КГц |
|----|---------------------------|---------------|--------------------|
| 0 | 4800 (низкая девиация) | ± 5 | 10 |
| 1 | 4800 (высокая девиация) | ± 20 | 40 |
| 2 | 9600 (низкая девиация) | ± 10 | 20 |
| 3 | 9600 (высокая девиация) | ± 20 | 40 |
| 4 | 19200 (низкая девиация) | ± 20 | 40 |
| 5 | 19200 (высокая девиация) | ± 40 | 80 |
| 6 | 38400 (низкая девиация) | ± 40 | 80 |
| 7 | 38400 (высокая девиация) | ± 100 | 200 |
| 8 | 57600 | ± 100 | 200 |
| 9 | 76800 | ± 100 | 200 |
| 10 | 4800 | $\pm 2,4$ | 4,8 |
| 11 | 9600 | $\pm 4,8$ | 9,6 |
| 12 | 14400 | $\pm 7,2$ | 14,4 |
| 13 | 19200 | $\pm 9,6$ | 19,2 |



Из ряда скоростей Ar0-7 рекомендуем использовать только нечетные значения параметра Ar (с высокой девиацией): Ar1, Ar3, Ar5 и Ar7.

Скорости Ar10-13 предназначены для работы с радиомодемами Спектр 9600 GM.

4.3 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:

["Режимы передачи данных"](#);

["Вопросы и ответы"](#).

5 РЕЖИМЫ РАБОТЫ МОДУЛЯ

Модуль может находиться в одном из следующих режимов:

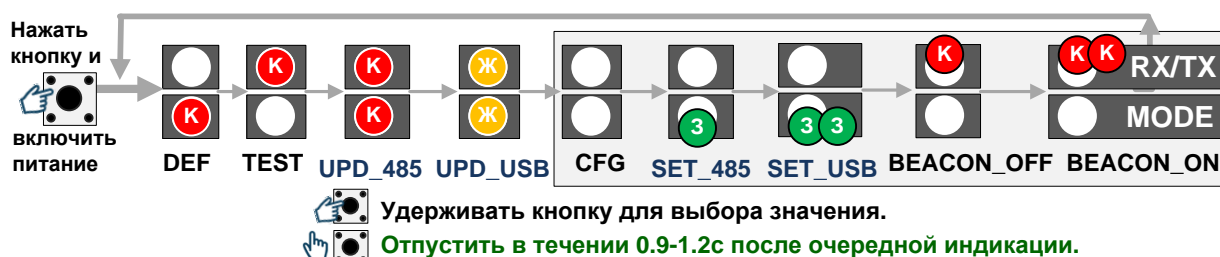
- **Режим загрузчика** – дополнительные режимы (команды) конфигурации при включении питания;
- **Командный** – режим конфигурации с помощью компьютера;
- **Рабочий** – "штатный", режим работы;

5.1 РЕЖИМ ЗАГРУЗЧИКА

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

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

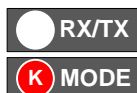
Вход в режим осуществляется при зажатии кнопки "CFG" и включении питания. Текущие значения отображаются на индикаторах "RX/TX" и "MODE". Выбор осуществляется отпускание кнопки в нужный момент в соответствии со схемой:



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

5.1.1 ВХОД В КОМАНДНЫЙ РЕЖИМ ПО УМОЛЧАНИЮ (**BOOT_DEF**)

Индикация для отпущения



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

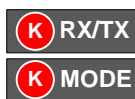
Также, модуль ввода-вывода переходит в тестовый режим (см. раздел [Тест ВХОДОВ/ВЫХОДОВ](#)).

5.1.2 РЕЖИМ ТЕСТ (**BOOT_TEST**)

На данный момент функциональность не определена.

5.1.3 РЕЖИМ СМЕНЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (BOOT_UPD_RS485/ BOOT_UPD_USB)

Индикация для отпущания



BOOT_UPD_RS485



BOOT_UPD_USB

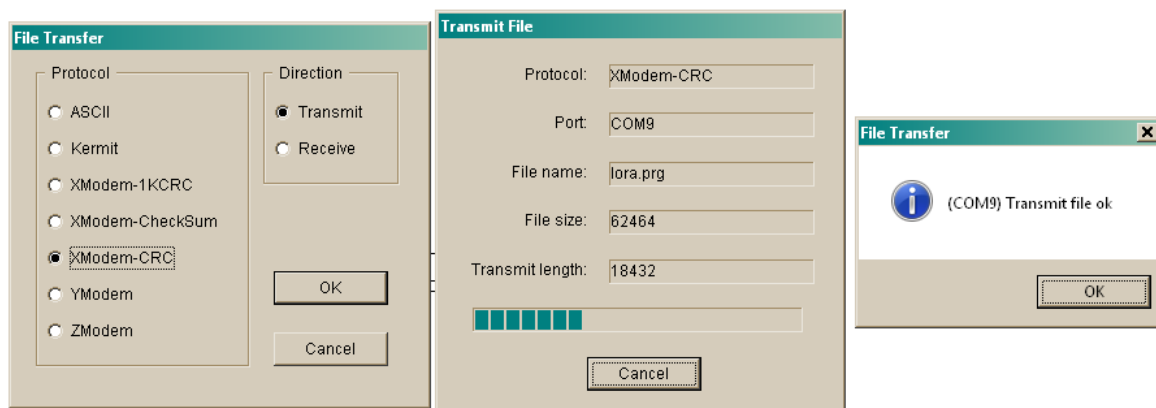
Для смены ПО следует:

- 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 433IO_v2 (c)Rateos 1998-2024 bv01;
Ready rcv by XModem-CRC...CCCCC
```

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

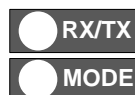
С помощью инструмента «Передача файлов» отправьте в модуль файл с нужной версией встроенного ПО в протоколе Xmodem-CRC и дождитесь окончания передачи.



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

5.1.4 АЛЬТЕРНАТИВНАЯ КОНФИГУРАЦИЯ (BOOT_CFG)

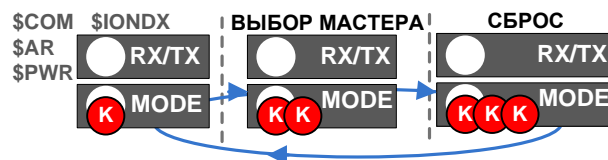
Индикация для отпущания



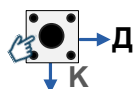
В данном режиме доступно:

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

После отпущания 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**.

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

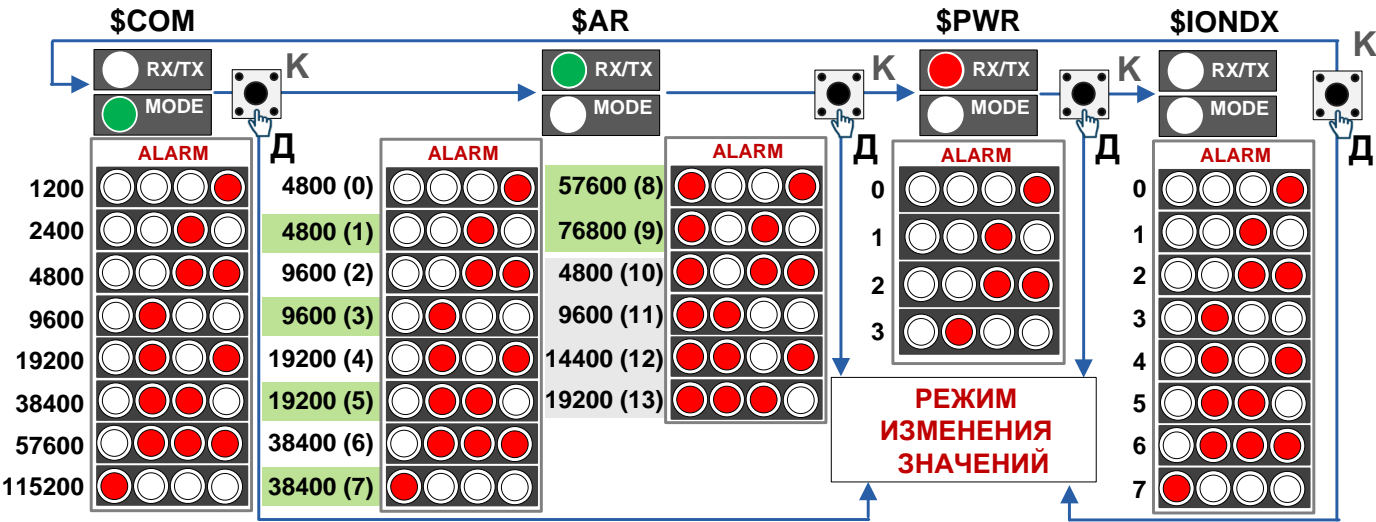
ИЗМЕНЕНИЕ РЕГИСТРОВ \$COM, \$AR, \$PWR и \$IONDX.

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".

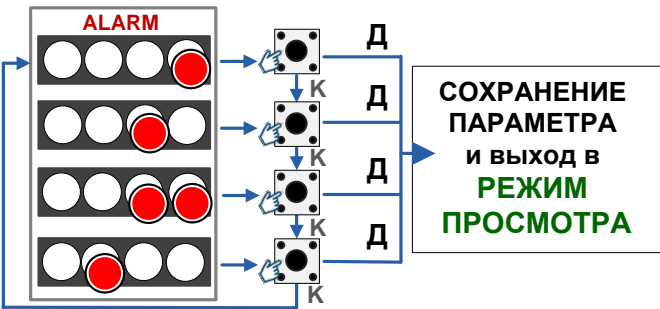
Если текущее значение \$IONDX больше 7, все светодиоды ALARM зажгутся. При сохранении параметра \$IONDX устанавливается режим \$IOSLAVE, базовый адрес \$IOADR устанавливается в значение 0x0A.

Всего доступны два режима - режим просмотра и изменения.

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

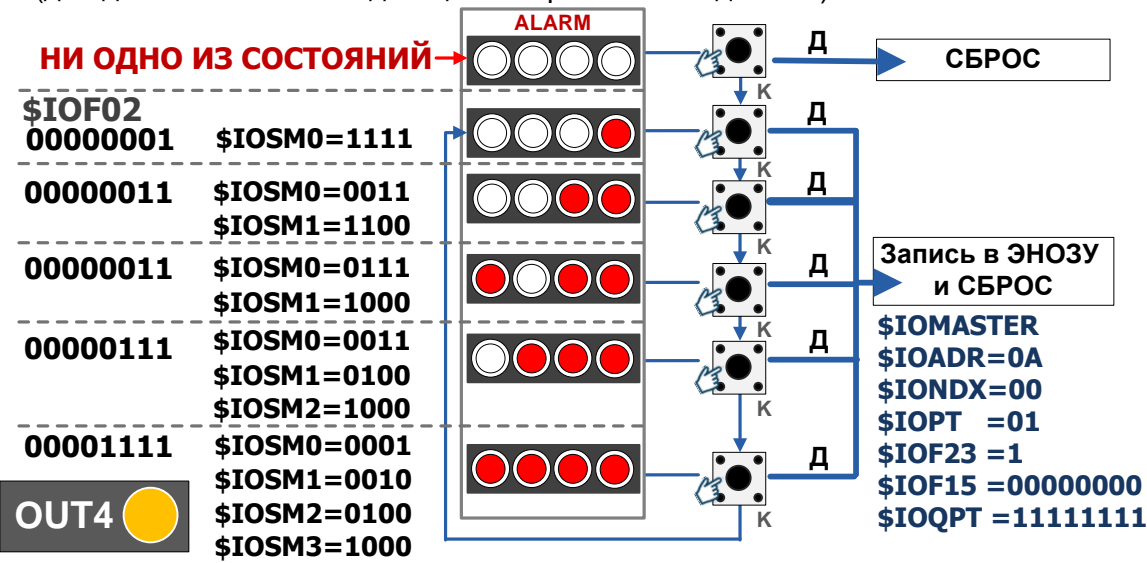


Режим изменения значений.
На светодиодах ALARM показываются варианты **миганием красным**.
На рисунке показан пример изменения параметра \$PWR.



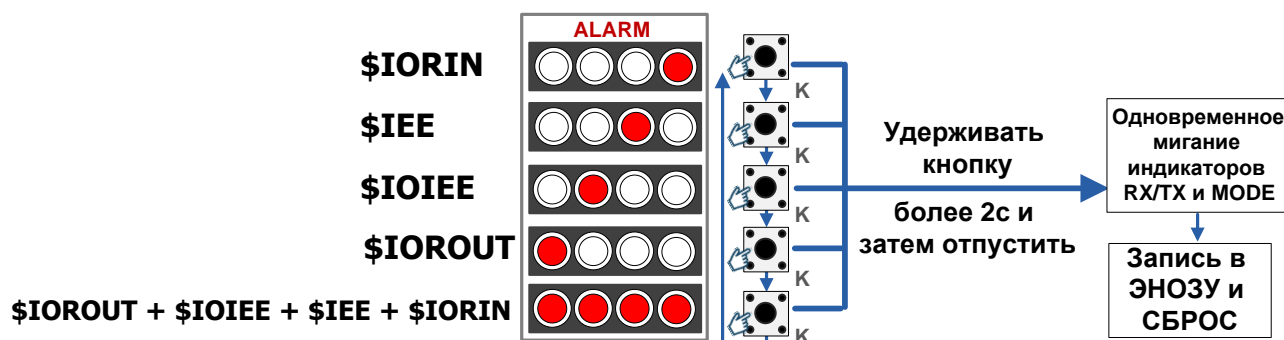
ВЫБОР МАСТЕРА.

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.

5.1.5 УСТАНОВКА АКТИВНОГО ПОРТА (**BOOT_SET_RS485/BOOT_SET_USB**)

Индикация для отпущения (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 перехода 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 ЭНЗУ модуля.

5.1.6 ВКЛЮЧЕНИЕ РЕЖИМА "Маяк" (**BOOT_BEACON_OFF/BOOT_BEACON_ON**)

Индикация для отпущения (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 |

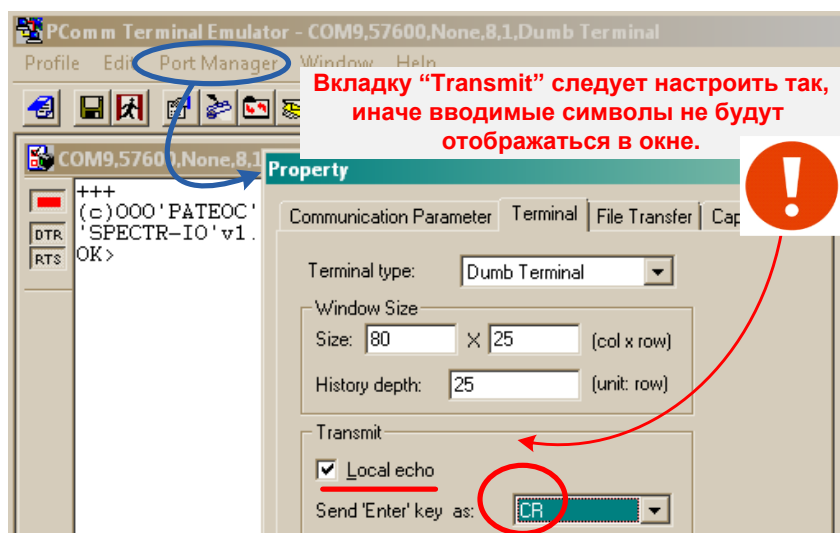
В случае успешного действия, индикаторы RX/TX и MODE начнут одновременно мигать желтым цветом, операция будет выполнена и модуль пересбросится. Если режим загрузчика заблокирован (в регистре \$LOCK), то при входе в данные режимы индикаторы RX/TX и MODE начнут мигать красным цветом и модуль просто пересбросится.

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

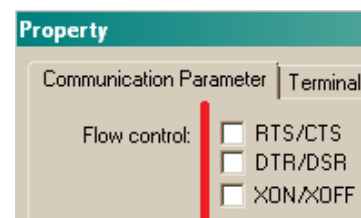
5.2 РЕЖИМ "КОМАНДНЫЙ"

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 Moxa 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) 000'PATEOC' 23/02/2024 (17:36)
'SPECTR 433IO_V2'v1.01 #sn:RM1000001 #name:
OK>
```

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;

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

```
OK> $DMP
P2 TXID=FFFF COPY=1 ,MYID=0004
CH=92 PWR=0 COM=9600,8N1
AR=3 LBT=0 DAT=EOT PACT=005 PSI=000 PL=7
MN=3 DM=0 FEC=R7,3 RCONC=0 TCONC=0
IOADR=0A+IONDX=00 (0Ah) TFL=---- RPT=--
IORPT=11111111 RFL=---- RTFL=----
MaxTt=586 ms
IOSLAVE IOALR=F0
IOSFT=030 IOSFV=----
IOINL=---- IOBN=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> $psl=333 ← 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.

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).

5.3 РЕЖИМЫ ПЕРЕДАЧИ ДАННЫХ



Данный раздел описывает режимы передачи данных по порту данных. Если используется только модуль ввода-вывода (I/O), то ознакомление с текущим разделом можно пропустить.

Модуль имеет внутренний 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 или ошибках - **мигает красным**.

В адресном режиме с подтверждением, 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 остальных режимах 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.

Режим передачи данных задается регистром \$DAT и имеет следующие значения:

EOT - "Прозрачный" режим

EOC - Режим передачи по символу.

RTU - Передача пакетов Modbus RTU

DL8 - Передача пакетов с контрольной суммой DALLAS CRC8

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 PACT (например, в режиме протокола EOT) из буфера будет удалена вся строка "11122243337".

Переполнение входящего буфера является признаком неправильно настроенной логики работы внешнего оборудования. Как правило, это неверно выставленный тайм-аут ожидания ответа в протоколе - 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 \$PL (Packet Limit), 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 \$PL не гарантирует работоспособность системы (где нет регулировки тайм-аута) по эфиру. По умолчанию, параметр \$PL отключен (равен 0) - в буфере может находиться максимально возможное число пакетов.

5.3.1 РЕЖИМ ПЕРЕДАЧИ ДАННЫХ «ПРОЗРАЧНЫЙ» EOT

Режим передачи данных "прозрачный" задается командой \$DAT=EOT. Связанные с этим режимом команды - \$PACT и \$PSL. В этом режиме пользовательские данные передаются в эфир без какого либо анализа со стороны модуля.

Признаком формирования 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. Тайм-аут задается числом символов на текущей скорости порта данных и программируется командой \$PACT (PACket Timeout). Например, если скорость по порту равна 9600, то 1 символ передается за ~1.04 мс. При значении \$PACT=015, таймаут будет равен $15 \cdot 1.04 = 15.6$ мс.

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 может служить параметр "нарезки" данных \$PSL (Packet SLice), который задает размер в байтах $\times 8$ при превышении которого, уже принятые ранее в буфер данные без пауз "защелкиваются" для передачи в эфир. Например, если значение \$PSL=025 ($25 \times 8 = 200$) и в буфер модуля непрерывным потоком поступает 800 байт, то в итоге в буфере сформируется 4 пакета. "Нарезка" происходит только при не нулевом значении параметра \$PSL. Если \$PSL=000 (по умолчанию) то в текущем примере по окончании приема 800 байт сработает только таймаут \$PACT и в буфере будет 1 пакет размером 800 байт.

Таким образом, условиями для формирования пакета для передачи в эфир в данном режиме являются тайм-аут \$PACT по приему последнего байта **ИЛИ** размер принятых данных превысил значение \$PSL.

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

5.3.2 РЕЖИМ ПЕРЕДАЧИ ДАННЫХ «ПО СИМВОЛУ» EOC

Режим передачи данных "по символу" задается командой \$DAT=EOC. Связанные с этим режимом команда - \$EOC.

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

5.3.3 РЕЖИМ ПЕРЕДАЧИ ДАННЫХ «MODBUS» RTU

В режиме RTU признаком конца данных является тайм-аут \$PACT (аналогично режиму EOT). Одновременно с приемом, осуществляется подсчет контрольной суммы (CRC) данных по стандарту протокола MODBUS RTU. По истечении \$PACT 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. 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 анализ данных на соответствие пакетов стандарту Modbus RTU по контрольной сумме. Также, возможна фильтрация по адресам с помощью фильтров на передачу.

5.3.4 РЕЖИМ ПЕРЕДАЧИ ДАННЫХ «DALLAS» DL8

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

5.3.5 КОНКАТЕНАЦИЯ (СКЛЕИВАНИЕ) ПАКЕТОВ ПРИ ПЕРЕДАЧЕ

В случае, когда необходимо часто передавать преимущественно в одну сторону много "мелких" пакетов, в модуле предусмотрена процедура конкатенации их в один пакет на стороне передатчика. Приемный модуль при получении такого пакета производит обратную процедуру де конкатенации и выдает каждый "мелкий" пакет в свой порт с таймаутом РАСТ. Данное свойство позволяет увеличить пропускную способность канала за счет уменьшения количества передач пакетов в эфир, т.к. каждый пакет кроме пользовательской информации содержит еще и служебную. Временное разделение пакетов при этом сохраняется - приемный модуль выдает в порт их как отдельные пакеты, а не единым потоком.

Например, на порт передающего модема постоянно поступают пакеты размером 15 байт. При отсутствии конкатенации на передачу, модуль постоянно будет выходить в эфир на передачу пакетов размером по 15 байт. При конкатенации, при очередной передаче модуль обнаруживает, что в буфере есть, например, уже больше 10 пакетов размером 15 байт. Вместо того, чтобы продолжать передавать в эфир каждый пакет по отдельности, модуль склеивает 10 пакетов в единый пакет размером 150 байт и передает его в эфир 1 раз. Приемный модем получая этот пакет, выдает в порт 10 раз по 15 байт выжидая после каждых 15 байт время РАСТ.

Признак конкатенации на передачу задается параметром TCONC (см. команду [\\$TCONC](#)), которая задает уровни максимального размера пакета при конкатенации. Уровень 0 соответствует запрещению конкатенации, а уровень 7 (максимальный) означает, что размер конкатенированного пакета может быть до 255 байт. К примеру, если уровень конкатенации равен 7 (размер пакета до 255 байт), то склеить можно 3 пакета по 80 байт. Или 4 пакета размерами 10, 50, 60, 80 байт соответственно. Пакеты размером превышающие уровень конкатенации передаются как обычно.

Необходимо иметь ввиду, что конкатенация применяется к пакетам, которые уже находятся в буфере - модуль не ждет прихода по порту пакетов для дополнения конкатенированного пакета до максимального размера, а склеивает то, что готово для передачи в текущий момент.

В системах "запрос-ответ" в конкатенации на передачу нет необходимости. По умолчанию, конкатенация отключена (\$TCONC=0) и доступна только в протоколе V2.

5.3.6 ОСОБЕННОСТИ РЕЖИМОВ ПЕРЕДАЧИ ДАННЫХ

При работе радиомодулей внешнее оборудование «не замечает» их и работает так же, как и по проводам.

При этом следует понимать особенности передачи данных через радиомодули. Основные причины, по которым система отказывается работать через модули (естественно, предполагается, что осуществлена проверка связи между самими модулями):

- задержка при передачи 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 проводу заключается в задержках при передаче данных. Если при работе по проводу задержки практически отсутствуют, то при работе через радиомодули они могут составлять сотни миллисекунд и даже секунды.

Такие задержки возникают из-за того, что путь прохождения данных при работе через модули «удлиняется»: 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), система предусматривает такое увеличение — мастер (пульт, ПЛК, OPC-сервер) должен иметь возможность конфигурации временных параметров протокола.

Для установки времени ожидания ответа для мастера необходимо ознакомиться с разделом "[Оценка времени передачи](#)".

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. Рассмотрим на примере режим ЕОТ ("прозрачный").

Условием формирования пакета данных для передачи в эфир является пауза между приходящими данными (параметр \$PACT) - если пауза превышает заданное значение, данные "защелкиваются" в пакет. Все идущие следом данные в этот пакет уже не попадут и будут переданы следующим пакетом.

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

При этом получатель получит все 100 байт данных, 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 \$PACT.

По умолчанию \$PACT=005, что соответствует 5 мс и паузе в ~5 байт на скорости по умолчанию 9600. Это достаточно для большинства применений.

Кроме паузы между байтами также следует учитывать максимальный размер данных, которые модуль может передать в эфир за 1 раз (512 байт). 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 метода конкатенации данных.

5.3.7 БУФЕРИЗАЦИЯ ПРИНЯТЫХ ИЗ ЭФИРА ДАННЫХ. КОНКАТЕНАЦИЯ ДАННЫХ ПРИ ПРИЕМЕ

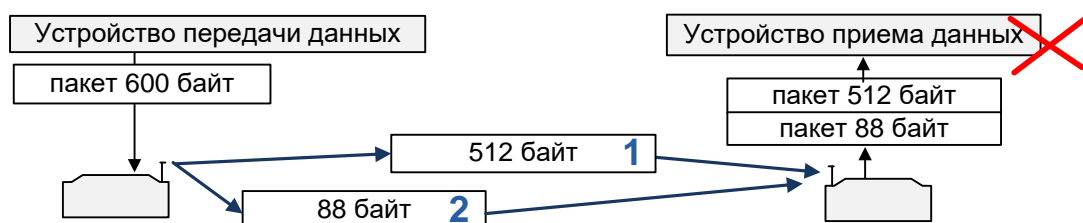
На принятые из эфира данные в модуле предусмотрен буфер. В случае если скорость обмена по эфиру намного превышает скорость обмена по последовательному порту это предотвращает потерю данных. Данные приёмных буферов последовательно передаются на последовательный порт модуля в порядке поступления их из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



Модуль не может передать все 600 байт одним пакетом. Он разбивает его на части и каждую часть маркирует специальными маркерами. Приемный модуль, при активации данной функции может их конкатенировать (объединить) в один пакет и выдать в порт единым целым.

Для функции конкатенации в модуле выделены 2 буфера по 1024 байт (общий размер 1 пакета для конкатенации не может превышать 1024 байт).

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

- **получен признак последних данных;**
- **буфер полон** (приходящие данные имеют признак не последних данных и их общий объем превышает или равен размеру буфера).

Конкатенация активизируется установкой флага \$RCONC = 1, иначе данные выдаются на последовательный порт модуля по мере поступления из эфира.

5.4 РЕЖИМЫ РАБОТЫ ПО ЭФИРУ

5.4.1 АДРЕСАЦИЯ



Данный раздел описывает возможности модуля при передачи данных по порту данных. Если в модуле используется только модуль ввода-вывода (IO), то ознакомление с текущим разделом можно пропустить - пакеты модуля IO всегда передаются в широковещательном режиме.

В модуле возможно использование 65534 (0001...FFFF) адресов. Адреса разделены на индивидуальные, групповые и широковещательный. Адрес содержит 4 цифры в 16-ричном формате.

- Адрес является **широковещательным**, если он равен FFFF.
- Адрес является **групповым**, если он начинается или заканчивается FF.
- Все остальные адреса являются **индивидуальными**.

Каждый модуль имеет два адреса – собственный \$MYID и адрес получателя \$TXID. Адрес получателя может быть любым, а адрес отправителя 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 1200, 1210,...121E образуют группу. Для передачи данных всем адресатам данной группы необходимо адресу получателя присвоить значение 12FF.

Пакеты, передаваемые в эфире, содержат информацию об адресах, на основании этой информации каждый принявший пакет модуль может судить о «принадлежности» и «назначении» данного пакета. Например, если один из модулей имеет TXID=12FF, его пакеты будут «принимать» все модули, адреса MYID которых начинаются с 12. Если же, например, адрес TXID=2050, его пакеты будет «принимать» только модуль с адресом MYID=2050.

Модуль может работать в эфире в трех режимах:

- широковещательный (или групповой);
- адресный без подтверждения;
- адресный с подтверждением.

5.4.2 ШИРОКОВЕЩАТЕЛЬНЫЙ РЕЖИМ

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



В ширококвещательном 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 (протоколы ModBus и им подобные).

Вероятность доставки пакетов может быть увеличена вспомогательными методами:

- помехоустойчивым кодированием;
- мультипликацией данных;
- снижением скорости передачи в эфире.

Также, для увеличения вероятности доставки пакета в ширококвещательном режиме задействованы два дополнительных параметра - COPY и PAUSE. Параметр COPY определяет число копий ширококвещательных пакетов, параметр PAUSE - паузу в 100мс интервалах между передачей копий пакетов. Например, COPY=3 и PAUSE=10 означает, что каждый ширококвещательный пакет будет передаваться по 3 раза с паузой между ними 1с. Если значение COPY = 1, то параметр PAUSE не имеет значения.

Примеры конфигурации (при выводе команды \$DMP):

| | |
|---------------------------|---|
| TXID=FFFF COPY=1 | - ширококвещательный режим. Передается 1 копия пакета. |
| TXID=00FF COPY=1 | - групповой режим. Передается 1 копия пакета. |
| TXID=FFFF COPY=2 PAUSE=10 | - ширококвещательный режим. Каждый пакет передается по 2 раза с паузой в 1 с. |

Особенностью ширококвещательного режима является то, что получив ширококвещательный пакет модуль выдает его на свой порт данных в независимости от своего значения \$MYID. Данный режим необходимо устанавливать на базовом модеме в случае конфигурации сети "точка-многоточка".

По умолчанию, параметр COPY=1 и в подавляющем большинстве случаев нет необходимости его изменять.

5.4.3 АДРЕСНЫЙ РЕЖИМ БЕЗ ПОДТВЕРЖДЕНИЯ

Для работы в адресном режиме без подтверждения параметр TXID должен содержать индивидуальный адрес, а параметр WA **у передающего модуля** установлен в 0.

В данном режиме отсутствует гарантия доставки. Методы для увеличения вероятности доставки аналогичны ширококвещательному режиму. В адресном режиме без подтверждения задействованы дополнительные параметры COPY и PAUSE. Параметр COPY определяет число копий индивидуальных пакетов, параметр PAUSE - паузу в 100мс интервалах между передачей копий пакетов.

Примеры конфигурации (при выводе команды \$DMP):

| | |
|--------------------------------|---|
| TXID=0001 WA=0 COPY=1 | - передается 1 копия пакета. |
| TXID=0001 WA=0 COPY=2 PAUSE=10 | - каждый пакет передается по 2 раза с паузой в 1 с. |

Особенностью данного режима является то, что при получении модулем адресного пакета, на свой порт данных его выдаст только тот модуль, параметр MYID которого равен адресу получателя этого пакета. Данный режим можно устанавливать на всех модулях кроме базового в случае конфигурации сети "точка-многоточка".

По умолчанию, параметр COPY=1 и в подавляющем большинстве случаев нет необходимости его изменять.

5.4.4 АДРЕСНЫЙ РЕЖИМ С ПОДТВЕРЖДЕНИЕМ

Для работы в адресном режиме с подтверждением параметр TXID должен содержать индивидуальный адрес, а параметр WA **у передающего модуля** отличен от 0.

При получении такого пакета приемный модуль отправляет короткое подтверждение (ACK) о его приёме отправителю (если пакет принят с ошибками, подтверждение не

отправляется). Отправитель же при неполучении подтверждения повторяет пакет. Таким образом, в данном режиме имеются **гарантии** доставки данных.

Не нулевой параметр WA задает время ожидания подтверждения, которое выражается и кратно времени передачи 1 пакета максимального размера при текущей скорости в эфире и способа кодирования.

В данном режиме задействован дополнительный параметр TRY, который задает число попыток (0-7) передачи пакета данных в случае отсутствия подтверждения. Если параметр TRY равен 0, то при отсутствия связи с удаленным модулем, пакет данных будет ретранслироваться **бесконечное** число раз. Если параметр TRY равен от 1 до 7, то по истечении числа передач TRY и отсутствия подтверждения, передающий модуль **удаляет** текущий пакет данных из своего буфера.

Удаленный модуль при этом может быть как в широковещательном, так и в одном из адресных режимах.

Примеры конфигурации (при выводе команды \$DMP):

TXID=0001 WA=1 TRY=0 время ожидания подтверждения равно времени передачи 1-го пакета максимального размера. В случае отсутствия связи с модулем 0001, пакет передается бесконечное число раз.

TXID=0010 WA=2 TRY=3 время ожидания подтверждения равно времени передачи 2х пакетов максимального размера. В случае отсутствия связи с модулем 0010, пакет передается 3 раза после чего удаляется.

При отсутствии повторителей в системе, параметр WA необходимо устанавливать в 1.

5.4.5 ОСОБЕННОСТИ ПРИМЕНЕНИЯ АДРЕСНОГО РЕЖИМА С ПОДТВЕРЖДЕНИЕМ

Если предполагается использовать модули для связи только двух объектов между собой, то на первый взгляд более логичным выглядит использование адресного режима с подтверждением, а не широковещательного, поскольку в режиме с подтверждением модули «гарантируют» доставку данных.

Тем не менее, когда внешнее оборудование использует протоколы с собственным механизмом проверки целостности и доставки сообщений (Болид, Modbus и подобные), даже в системах из 2-х объектов лучше использовать широковещательный режим или адресный без подтверждения.

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, оборудование работает по принципу «запрос-ответ» - мастер посылает запрос и ждет 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 с подтверждением начинает повторять его несколько раз. 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 адресного режима с подтверждением 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 широковещательном же режиме (или адресный без подтверждения) неуправляемых и необъяснимых ситуаций 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.



В отличие от остальных режимов, в адресном режиме с подтверждением в случае установки параметра \$TRY в не нулевое значение и передачи пакета несколькими частями, в случае отсутствия связи (достижение счетчика TRY заданной величины) не переданные оставшиеся данные удаляются из буфера.

Допустим, что в режиме EOT на порт данных пришел единый пакет размером 600 байт. Модуль не может передать 600 байт одним пакетом и разобьет его на 2 пакета, например, 512 и 88 байт. В случае, если счетчик неполучения подтверждения достигнет своего заданного значения при передаче первого пакета 512 байт, то оставшийся пакет 88 байт не будет передаваться и будет удален из буфера.

5.5 РЕТРАНСЛЯЦИЯ ПАКЕТОВ

Модуль способен ретранслировать пакеты, не утрачивая своих основных функций.

В системе может быть до 8 ретрансляторов, номера которых задаются командой **\$RT**. 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 RIDx. Если в РП отсутствует маркер ретрансляции для данного модуля и адрес в пакете совпал с одним из RIDx, 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 **\$RT**) 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»](#).

5.6 АЛГОРИТМ ВЫХОДА В ЭФИР (LBT)

В нормальном режиме работы, при наличии данных и отсутствия приема полезных данных в текущий момент, модуль сразу начинает передавать свои данные в эфир. Этого достаточно для большинства применений построенных по системе "запрос-ответ". Однако возможны варианты построения систем, когда несколько модулей должны независимо друг от друга передавать данные. В этом случае, с большей долей вероятности могут возникнуть коллизии при передаче данных, что приведет к их потере уже на начальном этапе передачи.

Для таких случаев, в модуле реализован простой алгоритм множественного доступа к эфиру (Listen Before Transmit). Алгоритм заключается в псевдослучайной задержке перед передачей, в течении которой осуществляется анализ эфира на предмет приема пакета. Если в течении этой задержки из эфира принят пакет (или начинается прием), после его обработки задержка генерируется заново. Если приема пакета не было, модуль начинает передавать свои данные. Алгоритм задается параметром **\$LBT** и может принимать следующие значения:

- 0 - отсутствие задержек перед передачей (по умолчанию);
- 1 - "длинный" алгоритм. Задержка для анализа эфира кратна времени передачи 1/2 от длины преамбулы. Максимальная задержка равна времени передачи пакета 512 байт;
- 2 - "короткий" алгоритм. Задержка для анализа эфира кратна времени передачи 1/2 от

длины преамбулы. Максимальная задержка равна времени передачи пакета 256 байт;

К примеру, на скорости $AR=3$, $MN=3$ и выключенном FEC , время передачи пакета максимальной длины составляет 523 мс. В случае "длинного" алгоритма модуль генерирует псевдослучайную задержку кратную 8мс в пределах 523 мс (в случае "короткого" - кратную 8 мс в пределах 262 мс). Если в течении этой задержки нет приема пакета модуль выходит на передачу, иначе принимает пакет и заново генерирует задержку.

Как видно из сравнения, "короткий" алгоритм более "быстрый" и в случае небольшого количества модулей в системе, позволяет увеличить пропускную способность не тратя лишнее время на анализ эфира.



В системах с одним "мастером" рекомендуется отключать алгоритм LBT т.к. он может привести к не контролируемому времени отклика "слэйв" модулей из-за генерации псевдослучайных задержек перед передачей данных в эфир.

6 ФОРМАТ ПАКЕТА В ЭФИРЕ.

Максимальный *общий* размер пакета в эфире составляет 512 байт. Перед передачей в эфир данные проходят следующие этапы предварительной обработки:

- проверка данных на 7 битный формат. Сжатие в случае положительной проверки;
- добавление избыточности (помехоустойчивое кодирование);
- разбиение пакета на 32-байтные блоки и вычисление контрольной суммы для каждого блока.

При приёме пакета из эфира модем создает маркеры правильности приёма каждого блока информации - вычисляется его контрольная сумма 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).

В случае повторной ретрансляции одного и того же пакета (в адресном режиме с подтверждением) или передачи копий пакета (в широковещательном или адресном без подтверждения) модуль передает блоки внутри пакета в обратном порядке. Например, задан режим передачи пакета по 2 копии и необходимо передать 128 байт данных. Модем сформирует пакет из 4 блоков. В первой копии блоки будут передаваться в последовательности 0,1,2 и 3, а во второй - 3,2,1 и 0.

Применение маркеров уменьшает время достоверной и гарантированной передачи информации в случае не прохождения пакета с первого раза.

6.1 ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ

Для уменьшения повторных передач информационных пакетов, а также для более надёжной передачи данных кроме использования подтверждений, можно использовать прямое исправление ошибок (FEC). Для этого в модуле реализованы несколько способов помехоустойчивого кодирования: 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 | - |

При выборе FEC необходимо учитывать характер помех в эфире или режим работы модуля. Например, в адресном режиме с подтверждением, когда неправильно принятый пакет будет ретранслирован, как правило, целесообразно использовать менее мощные коды или вовсе обойтись без них. В режиме же широковещательный гарантия доставки данных отсутствует, и для повышения вероятности доставки использование FEC оправдано.

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.

Тип помехоустойчивого кодирования и перемежения задается командой \$FEC.



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

6.2 ПЕРЕМЕЖЕНИЕ

На практике часто искажаются не отдельные биты, а целые *последовательности* бит, поэтому при использовании FEC эффективно применение перемежения информационных и проверочных символов. Процесс заключается в передаче сначала первых символов каждого кодового слова, потом вторых, третьих и т.д. В случае возникновения пакетной ошибки

после процедуры де перемежения 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 |

6.3 Мультипликация данных

Если размер пользовательских данных для текущего пакета значительно меньше максимального размера пакета в эфире, модуль копирует данные по блокам. 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 | Мультипликация |
|---------------------------------|---------------------|----------------|
| RS(7,5) | 1 ... 66 | х 3 |
| | 67 ... 100 | х 2 |
| | 101 ... 512 | отсутствует |
| RS(7,3) | 1 ... 64 | х 3 |
| | 65 ... 97 | х 2 |
| | 98 ... 512 | отсутствует |
| RS(15,11) | 1 ... 66 | х 3 |
| | 67 ... 99 | х 2 |
| | 100 ... 512 | отсутствует |
| RS(15,9) | 1 ... 72 | х 3 |
| | 73 ... 108 | х 2 |
| | 109 ... 512 | отсутствует |
| HAM(12,8) или отсутствует | 1 ... 64 | х 3 |
| | 65 ... 96 | х 2 |
| | 96 ... 512 | отсутствует |

Например, для кода RS(7,5) размер мультипликации 3 означает, что если пользовательские данные для текущего пакета будут от 1 до 66 байт, то они будут дублированы 3 раза в одном пакете. Если размер данных превышает 101 байт, то мультипликации не будет.



Возможность мультипликации задается командой \$DM.
По умолчанию мультипликация выключена.



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

6.4 РАНДОМИЗАЦИЯ (СКРЕМБЛИРОВАНИЕ)

Для корректного функционирования приемника модема при выделении данных из эфира поток данных не должен иметь длинных (более 8 бит) последовательностей «0» или «1». Для обеспечения этого условия модем при передаче рандомизирует поток данных с помощью 16-битного генератора псевдослучайной последовательности. Вероятность наличия в потоке данных длинных последовательностей «0» или «1» уменьшается.

Рандомизация не может полностью исключить вероятность появления в потоке длинных последовательностей «0» или «1», поэтому с целью повышения надежности при приеме данных в модуле реализована функция принудительной вставки в поток данных «перепадов» уровней. Для этого биты в потоке данных дублируются своими инверсными значениями. Командой \$MN задается уровень, который определяет число бит передаваемых в эфир без изменений (после рандомизатора), после чего в выходной поток вставляется 1 бит, являющийся инверсией предыдущего. Таким образом, выходной поток данных будет обязательно иметь необходимые перепады. Определены следующие уровни:

| | |
|-------|--|
| \$MN= | Инверсия после: |
| 0 | каждого бита. (код Манчестер 2) |
| 1 | каждого 6-го бита |
| 2 | каждого 15-го бита |
| 3 | каждого 10-го бита (по умолчанию) |

6.5 ОЦЕНКА ВРЕМЕНИ ПЕРЕДАЧИ

В данном разделе приведена оценка общего времени передачи в миллисекундах на разных скоростях в эфире, уровнях MN и помехоустойчивого кодирования FEC пакетов с "полезной" нагрузкой 256 и 512 байт. Данная информация необходима для настройки тайм-аута у внешнего оборудования работающего по системе "запрос-ответ" через модули.

Если размер информации не превышает 256 байт, то необходимо пользоваться первой колонкой, если превышает - второй. Если же, информация не превышает 128 байт, то время из первой колонки допустимо разделить на 2.

Следует иметь ввиду, что это физическое время передачи информации при выбранных условиях кодирования. Оно не учитывает случаи, когда активен алгоритм LBT или в момент начала передачи модем обнаруживает начало пакета и занимается его приемом в следствии чего, время передачи пакета может быть увеличено.

Допустим, модуль настроен на следующие параметры:

- \$AR=3 - Скорость в эфире 9600 бит/сек
- \$FEC=3 - Код FEC RS(15.9)
- \$MN=3 - Вставка инверсного бита после каждого 10-го
- Размер пользовательских данных в запросе и ответе не будет превышать 256 байт.

Исходя из таблицы, максимальное время передачи пакета составляет 433 мс. Учитывая систему "запрос-ответ", общее время передачи по эфиру составит $2 \times 433 = 866$ мс. К этому времени нужно прибавить еще время передачи 256 байт по проводному интерфейсу и время отклика оборудования. Например, если скорость по интерфейсу 9600, то это время составит $4 \times 256 \times 10 / 9600 = 1.07$ с. Таким образом, тайм-аут ожидания ответа у внешнего мастера следует выбирать начиная от 2 с.

Если обмен в системе происходит короткими пакетами, для увеличения пропускной способности (частоты опроса слэйвов) время ожидание ответа можно постепенно уменьшать экспериментальным путем

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

Скорость в эфире 4800 (AR 0, AR 1, AR 10)*пакет 256 байт*

| | | MN = 0 | = 1 | = 2 | = 3 |
|---|-----------|--------|------|------|------|
| 0 | RS(7.5) | 1287 | 773 | 724 | 742 |
| 1 | RS(7.3) | 2083 | 1228 | 1148 | 1176 |
| 2 | RS(15.11) | 1263 | 759 | 712 | 728 |
| 3 | RS(15.9) | 1513 | 902 | 845 | 865 |
| 4 | HAM(12.8) | 1368 | 818 | 768 | 785 |
| 5 | HET | 942 | 575 | 542 | 553 |

пакет 512 байт

| | | MN = 0 | = 1 | = 2 | = 3 |
|--|--|--------|------|------|------|
| | | 2512 | 1485 | 1387 | 1421 |
| | | 4105 | 2395 | 2234 | 2290 |
| | | 2465 | 1456 | 1362 | 1394 |
| | | 2961 | 1738 | 1625 | 1664 |
| | | 2675 | 1575 | 1475 | 1508 |
| | | 1821 | 1088 | 1021 | 1045 |

Скорость в эфире 9600 (AR 2, AR 3, AR 11)*пакет 256 байт*

| | | MN = 0 | = 1 | = 2 | = 3 |
|---|-----------|--------|-----|-----|-----|
| 0 | RS(7.5) | 644 | 387 | 362 | 371 |
| 1 | RS(7.3) | 1042 | 614 | 574 | 588 |
| 2 | RS(15.11) | 632 | 380 | 356 | 364 |
| 3 | RS(15.9) | 757 | 451 | 423 | 433 |
| 4 | HAM(12.8) | 684 | 409 | 384 | 393 |
| 5 | HET | 471 | 288 | 271 | 277 |

пакет 512 байт

| | | MN = 0 | = 1 | = 2 | = 3 |
|--|--|--------|------|------|------|
| | | 1256 | 743 | 694 | 711 |
| | | 2053 | 1198 | 1117 | 1145 |
| | | 1233 | 728 | 681 | 697 |
| | | 1481 | 869 | 813 | 832 |
| | | 1338 | 788 | 738 | 754 |
| | | 911 | 544 | 511 | 523 |

Скорость в эфире 14400 (AR 12)*пакет 256 байт*

| | | MN = 0 | = 1 | = 2 | = 3 |
|---|-----------|--------|-----|-----|-----|
| 0 | RS(7.5) | 429 | 258 | 241 | 247 |
| 1 | RS(7.3) | 694 | 409 | 383 | 392 |
| 2 | RS(15.11) | 421 | 253 | 237 | 243 |
| 3 | RS(15.9) | 504 | 301 | 282 | 288 |
| 4 | HAM(12.8) | 456 | 273 | 256 | 262 |
| 5 | HET | 314 | 192 | 181 | 184 |

пакет 512 байт

| | | MN = 0 | = 1 | = 2 | = 3 |
|--|--|--------|-----|-----|-----|
| | | 837 | 495 | 462 | 474 |
| | | 1368 | 798 | 745 | 763 |
| | | 822 | 485 | 454 | 465 |
| | | 987 | 579 | 542 | 555 |
| | | 892 | 525 | 492 | 503 |
| | | 607 | 363 | 340 | 348 |

Скорость в эфире 19200 (AR 4, AR 5, AR 13)*пакет 256 байт*

| | | MN = 0 | = 1 | = 2 | = 3 |
|---|-----------|--------|-----|-----|-----|
| 0 | RS(7.5) | 322 | 193 | 181 | 186 |
| 1 | RS(7.3) | 521 | 307 | 287 | 294 |
| 2 | RS(15.11) | 316 | 190 | 178 | 182 |
| 3 | RS(15.9) | 378 | 226 | 211 | 216 |
| 4 | HAM(12.8) | 342 | 205 | 192 | 196 |
| 5 | HET | 236 | 144 | 136 | 138 |

пакет 512 байт

| | | MN = 0 | = 1 | = 2 | = 3 |
|--|--|--------|-----|-----|-----|
| | | 628 | 371 | 347 | 355 |
| | | 1026 | 599 | 559 | 573 |
| | | 616 | 364 | 341 | 349 |
| | | 740 | 435 | 406 | 416 |
| | | 669 | 394 | 369 | 377 |
| | | 455 | 272 | 255 | 261 |

Скорость в эфире 38400 (AR 6, AR 7)*пакет 256 байт*

| | | MN = 0 | = 1 | = 2 | = 3 |
|---|-----------|--------|-----|-----|-----|
| 0 | RS(7.5) | 161 | 97 | 91 | 93 |
| 1 | RS(7.3) | 260 | 154 | 144 | 147 |
| 2 | RS(15.11) | 158 | 95 | 89 | 91 |
| 3 | RS(15.9) | 189 | 113 | 106 | 108 |
| 4 | HAM(12.8) | 171 | 102 | 96 | 98 |
| 5 | HET | 118 | 72 | 68 | 69 |

пакет 512 байт

| | | MN = 0 | = 1 | = 2 | = 3 |
|--|--|--------|-----|-----|-----|
| | | 314 | 186 | 173 | 178 |
| | | 513 | 299 | 279 | 286 |
| | | 308 | 182 | 170 | 174 |
| | | 370 | 217 | 203 | 208 |
| | | 334 | 197 | 184 | 189 |
| | | 228 | 136 | 128 | 131 |

Скорость в эфире 57600 (AR 8)

пакет 256 байт

| | | MN = 0 | = 1 | = 2 | = 3 |
|---|-----------|--------|-----|-----|-----|
| 0 | RS(7.5) | 107 | 64 | 60 | 62 |
| 1 | RS(7.3) | 174 | 102 | 96 | 98 |
| 2 | RS(15.11) | 105 | 63 | 59 | 61 |
| 3 | RS(15.9) | 126 | 75 | 70 | 72 |
| 4 | HAM(12.8) | 114 | 68 | 64 | 65 |
| 5 | HET | 79 | 48 | 45 | 46 |

пакет 512 байт

| | | MN = 0 | = 1 | = 2 | = 3 |
|--|--|--------|-----|-----|-----|
| | | 209 | 124 | 116 | 118 |
| | | 342 | 200 | 186 | 191 |
| | | 205 | 121 | 114 | 116 |
| | | 247 | 145 | 135 | 139 |
| | | 223 | 131 | 123 | 126 |
| | | 152 | 91 | 85 | 87 |

Скорость в эфире 76800 (AR 9)

пакет 256 байт

| | | MN = 0 | = 1 | = 2 | = 3 |
|---|-----------|--------|-----|-----|-----|
| 0 | RS(7.5) | 80 | 48 | 45 | 46 |
| 1 | RS(7.3) | 130 | 77 | 72 | 74 |
| 2 | RS(15.11) | 79 | 47 | 45 | 46 |
| 3 | RS(15.9) | 95 | 56 | 53 | 54 |
| 4 | HAM(12.8) | 86 | 51 | 48 | 49 |
| 5 | HET | 59 | 36 | 34 | 35 |

пакет 512 байт

| | | MN = 0 | = 1 | = 2 | = 3 |
|--|--|--------|-----|-----|-----|
| | | 157 | 93 | 87 | 89 |
| | | 257 | 150 | 140 | 143 |
| | | 154 | 91 | 85 | 87 |
| | | 185 | 109 | 102 | 104 |
| | | 167 | 98 | 92 | 94 |
| | | 114 | 68 | 64 | 65 |

7 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. 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 \$RIDx вводятся следующие параметры:

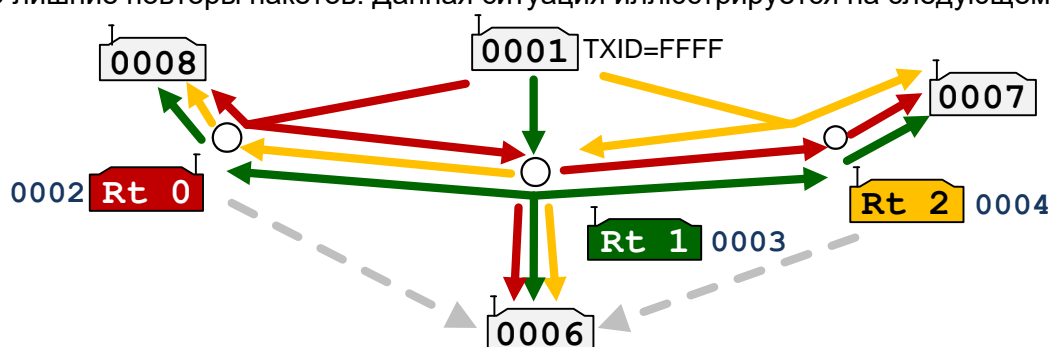
- признак адреса повторяемого пакета (адрес отправителя/адрес получателя);
- признак разрешения на повтор пакета с РП, равным «0»;
- специальная маска и логическая операция, которая определяет действие между запрограммированной маской и полученным РП пакета.



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 пакетов из-за множественной ретрансляции.

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

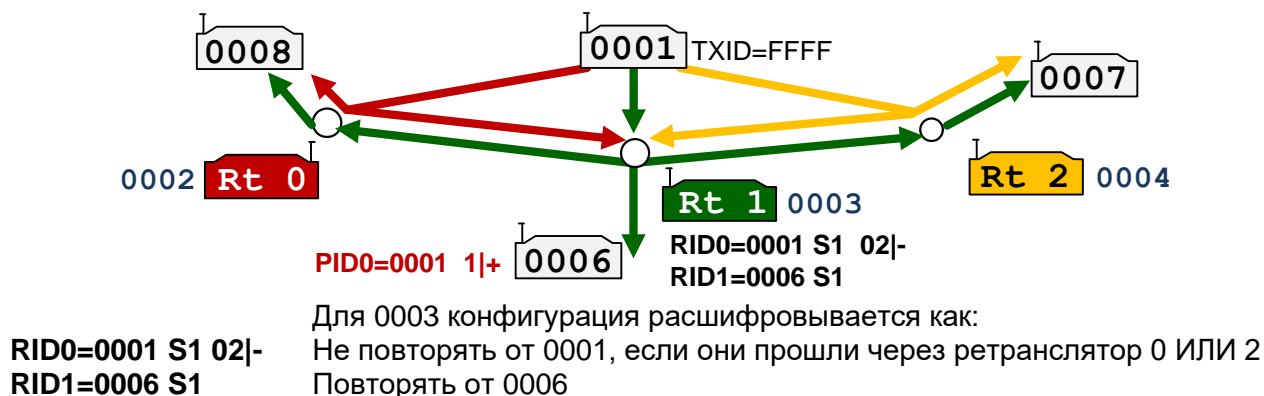
В случае если ретрансляторы находятся в прямой видимости друг от друга, могут иметь место лишние повторы пакетов. Данная ситуация иллюстрируется на следующем рисунке:



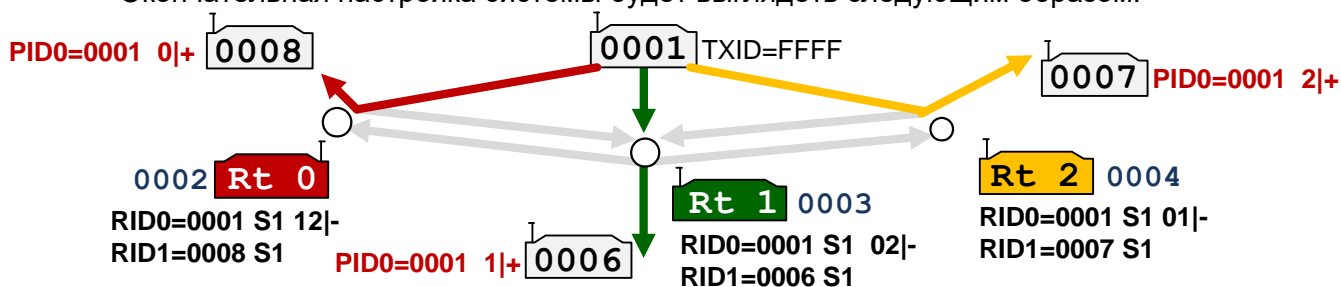
Модули 0002, 0003 и 0004 «слышат» друг друга и работают в режиме ретрансляторов пакетов для модулей 0008, 0006 и 0007 от базового 0001.

Рассмотрим прохождение широковекового пакета от 0001 до 0006. В данной ситуации 0003 повторит прямой пакет от 0001 и пакеты, ретранслированные модулями 0002 и 0004. Следовательно, в направлении 0006 будет 3 пакета от 0001. Также, на модуль 0006 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 0006 и 0003 теперь такую расширенную конфигурацию:



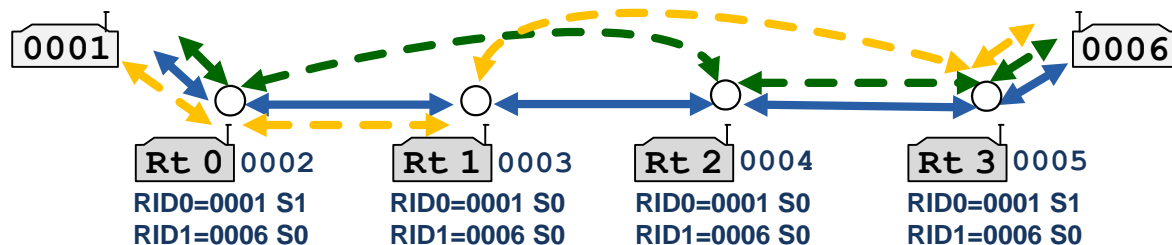
Окончательная настройка системы будет выглядеть следующим образом:

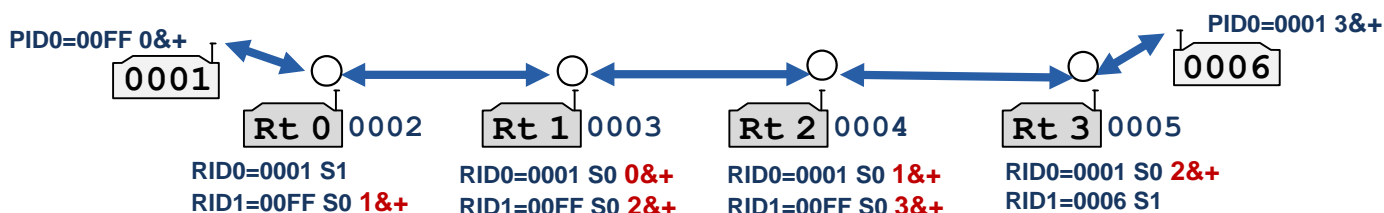


Примеры программирования масок:

- RID1=0001 S1 57&+** Ретранслировать пакеты от 0001, только если они уже ретранслированы ретрансляторами 5 И 7. В любом другом случае они не ретранслируются.
- RID7=0001 D1 57&-** Не ретранслировать пакеты, предназначенные модулю 0001, которые уже ретранслированы ретрансляторами 5 И 7. В любом другом случае они ретранслируются.
- RID1=0001 S0 07|+** Ретранслировать пакеты от 0001, только если они уже были ретранслированы ретрансляторами 0 ИЛИ 7. В любом другом случае они не ретранслируются.
- RID1=00FF S1 07|-** Не ретранслировать пакеты от группы абонентов (0001, 0002...000E), которые уже были ретранслированы ретрансляторами 0 ИЛИ 7. В любом другом случае они ретранслируются.

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

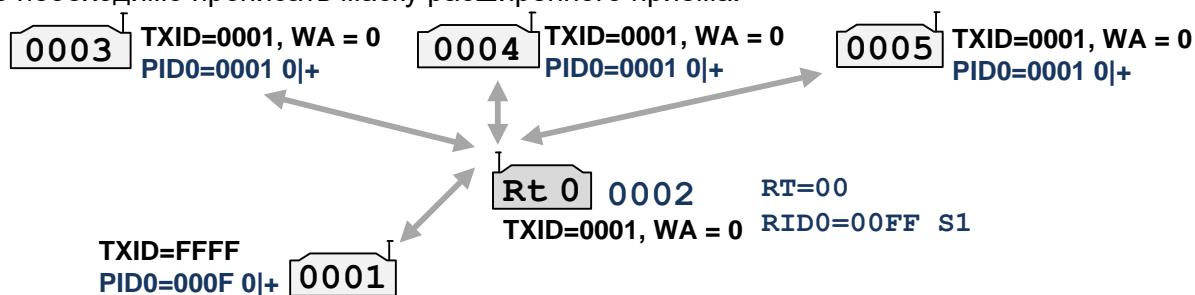




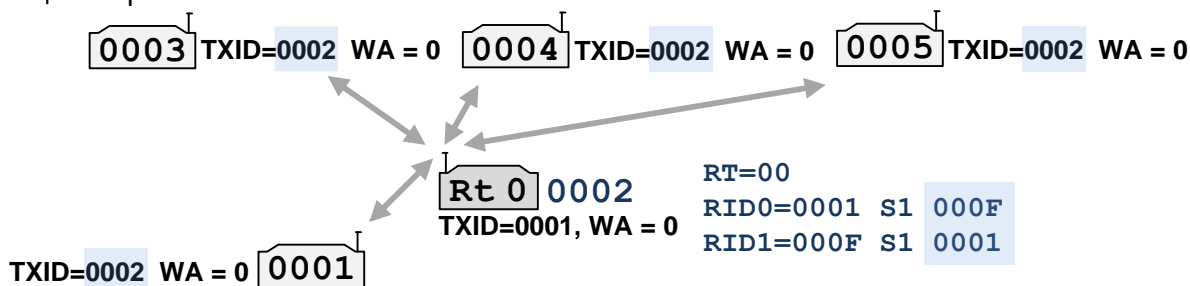
Введение соответствующих масок у ретрансляторов позволило строго определить путь прохождения пакетов между модулями 0001 и 0006 не обращая внимание если ли прямое прохождение сигнала между модулями ретрансляторами или нет.

7.1 РЕТРАНСЛЯЦИЯ С ПОДМЕНОЙ АДРЕСА ПОЛУЧАТЕЛЯ

Рассмотрим пример, когда с базового модуля 0001 необходимо опрашивать устройства подключенные к 0002-0005. Причем группа модулей 0003-0005 находится в не зоны устойчивой связи. После ввода повторителя Rt0 у модулей 0003-0005 необходимо прописать маски расширенного приема, чтобы исключить прием прямых пакетов. Базовому модулю также необходимо прописать маску расширенного приема:



Данная схема требует дополнительных настроек - программирование масок. С помощью подмены адреса получателя у ретранслятора настройки будут выглядеть следующим образом:



У всех модулей кроме ретранслятора 0002 настройки будут **одинаковые** и **отсутствуют** маски расширенного приема. Это позволит быть им взаимозаменяемыми - любой модем можно поменять местами в т.ч. с базовым. Также, конфигурация позволяет "отвязаться" от номера ретранслятора - его можно менять не конфигурируя остальные модемы.

Принцип работы данной схемы следующий: все модули (кроме ретранслятора) передают информацию в адресном режиме без подтверждения модулю 0002 (\$TXID=0002 WA=0), который является ретранслятором. Модуль 0002 получив пакет от модуля 0001 ретранслирует его и подменяет адрес назначения на 000F (\$RID0=0001 S1 000F). Пакеты от 0003-0005 также ретранслируются, но адрес назначения подменяется на 0001 (\$RID1=000F S1 0001).

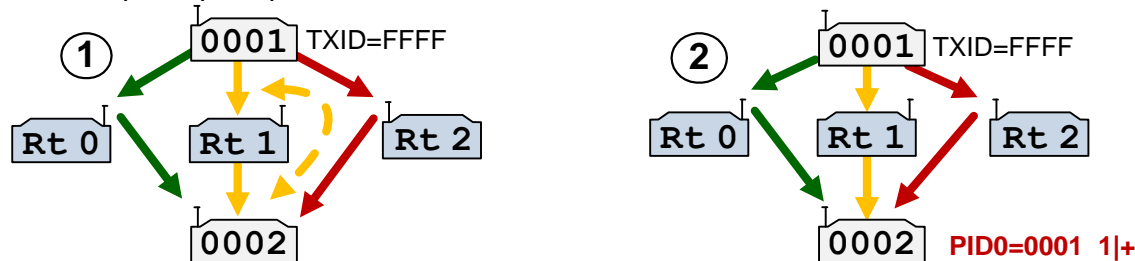
7.2 ПРИЁМ РЕТРАНСЛИРОВАННЫХ ПАКЕТОВ

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 ретрансляторов осуществляется командой **\$PIDx**. Всего может быть запрограммировано до 8 значений PID. Синтаксис ввода и логика маски аналогична маске при анализе РП процесса ретрансляции пакетов.

Рассмотрим пример:



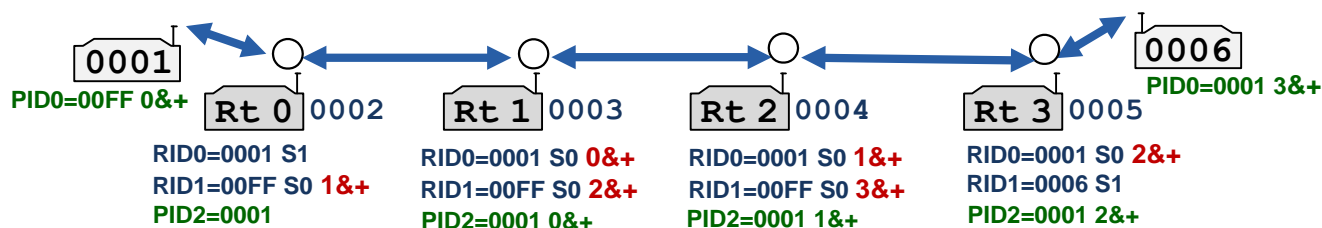
Допустим, что в данном примере неизвестно, как запрограммированы ретрансляторы 0,1 и 2. Но известно, что 0002 находится в радиовидимости всех 3-х. Также, есть частичное прямое прохождение пакетов между 0001 и 0002. Необходимо, чтобы пакеты от 0001 воспринимались модулем 0002 по **единственному** пути - прошедшие через ретранслятор 1.

После ввода модулю 0002 значение **PID0=0001 1|+**, он будет обрабатывать пакеты **только** прошедшие через ретранслятор 1. Все остальные будут игнорироваться.

Примеры программирования маски:

| | |
|-----------------------------|--|
| \$PID0=0001 57&+ | Принимать пакеты от 0001, только если они были ретранслированы ретрансляторами 5 И 7. В любом другом случае они игнорируются. |
| \$PID3=0004 57&- | Не принимать пакеты от 0004, которые были ретранслированы ретрансляторами 5 И 7. В любом другом случае они принимаются. |
| \$PID7=0011 07 + | Принимать пакеты от 0011, только если они уже были ретранслированы ретрансляторами 0 ИЛИ 7. Иначе они игнорируются. |
| \$PID3=00FF 07 - | Не принимать пакеты от группы 00, которые были ретранслированы ретрансляторами 0 ИЛИ 7. В любом другом случае они принимаются. |

Рассмотрим пример из цепочки ретрансляторов из предыдущего примера:



Если между ретрансляторами возможно прямое прохождение пакетов, то допустима ситуация когда, например, модуль 0003 получает "свой" запрос непосредственно от базового модуля 0001 и начинает передавать ответ. Но в это же время начинается ретрансляция этого же пакета-запроса модулем 0002. Возможна коллизия в эфире в следствии чего, на ответ от 0003 будет "наложен" ретранслируемый пакет от 0002. В итоге запрос-ответ от модуля 0001 на 0003 не пройдет.

Задание соответствующих масок PID (на рисунке отмечены зеленым) исключает эту проблему - модуль примет и выдаст на свой порт запрос только тогда, когда совпадет условие в маске PID.

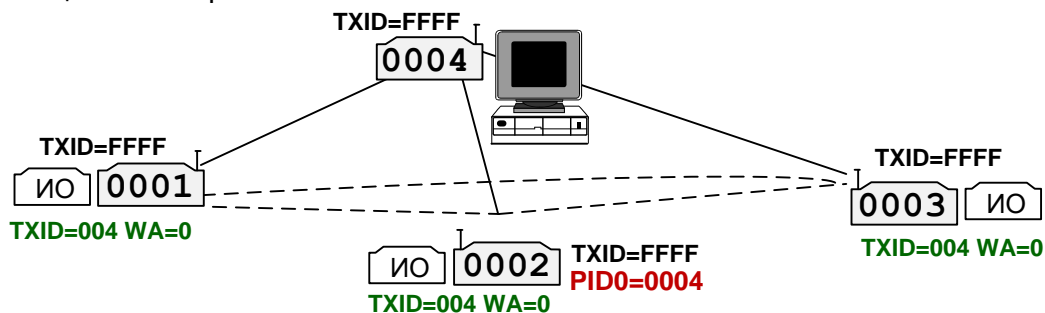
7.3 ИГНОРИРОВАНИЕ ПАКЕТОВ

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 широкоэвещательном режиме с одной или несколькими базовыми станциями, когда приём пакетов от абонентов, не являющихся базовой станцией, нежелателен. Адрес базовой станции задаётся командой

\$PIDx, в которой отсутствует маска анализа РП. Значение FF 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 широковещательного пакета может вызвать коллизию в сети, хотя пакет может быть предназначен только одному абоненту, подключённому в сеть RS-485.

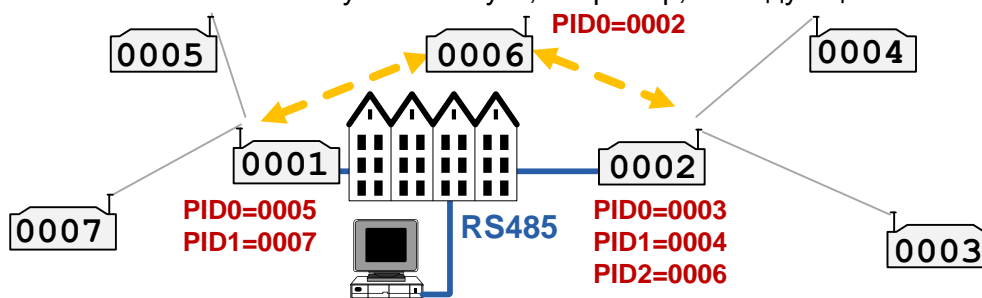
Рассмотрим пример конфигурации сети с одной базовой станцией, работающей в широковещательном режиме.



Базовая станция 0004 передаёт запрос на модули 0001-0003. 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 0003 необходимо не допустить получение информации от ИО других модулей, достаточно установить в 0003 значение PID, равное 0004. В этом случае на последовательный порт 0003 будет поступать информация только от базовой станции 0004.

Однако эту задачу можно решить другим способом (более правильным) - установить у модулей 0001-0003 адресный режим с 0004 без подтверждения (TXID=0004 WA=0).

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 0001 и 0002 по разные стороны здания. В данной ситуации выяснилось, что пакет от 0006 проходит как в 0001, так и в 0002. Оба модуля при этом одновременно выдают эти пакеты в сеть RS-485, что вызывает коллизию. Установка соответствующих значений PID на модулях 0001 и 0002 решает проблему. В данной конфигурации 0001 будет игнорировать данные от 0006. Информация от 0006 будет поступать в центральный пункт только через 0002.



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.

8 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+aa-bb Включение диапазона адресов aa-bb для разрешения фильтрации;
\$xFL-aa-bb 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. Иначе удаляется. 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 \$PACT - 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.

8.1 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).

8.2 ФИЛЬТРАЦИЯ ВХОДЯЩИХ ДАННЫХ

Фильтр входящих данных из эфира может принимать следующие значения:

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

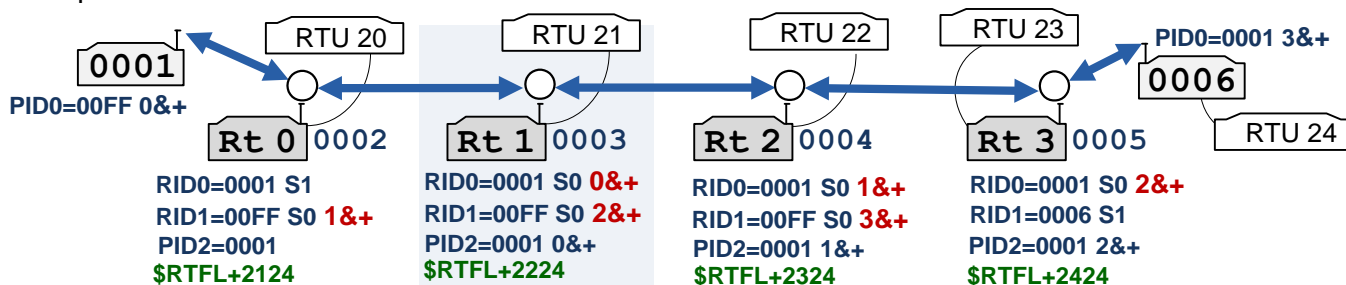
При значении фильтра NONE, данные полученные из эфира не выдаются на последовательный порт вообще.

8.3 ФИЛЬТРАЦИЯ ДАННЫХ ПРИ РЕТРАНСЛЯЦИИ

Фильтр ретранслируемых данных может принимать следующие значения:

----, RTUA, DL8A, 1BYT

Рассмотрим применение фильтра при ретрансляции на примере. Известно, что в системе применяется протокол modbus rtu. Ретрансляция осуществляется по цепочке. К модулям 0002-0006 подключены slave устройства с адресами 0x20...0x24 соответственно. Конфигурация на уровне расширенного приема и ретрансляции выглядит следующим образом:



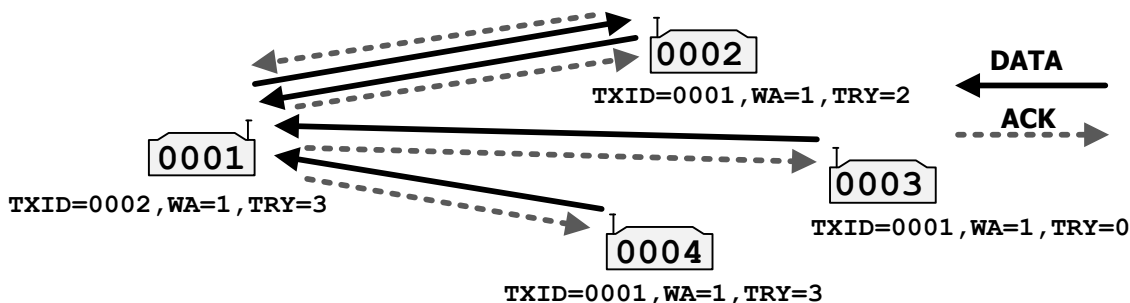
Без задания фильтра возникает ситуация когда, например, модуль 0003 получает modbus запрос 21. Сначала он выдает его на свой порт, а потом ретранслирует этот запрос, который далее будет ретранслирован по цепочке до модуля 0006. В этот момент устройство RTU 21 отвечает. В итоге имеем лишнюю ретрансляцию запроса, на который в цепочке далее никто не ответит.

Задание фильтра \$RTFL+2224 будет означать, что модуль 0003 будет ретранслировать modbus пакеты только с адресами 0x22, 0x23 и 0x24. Итоговая конфигурация фильтров для каждого устройства показана зеленым цветом. Также, необходимо будет выставить фильтр на ретрансляцию modbus пакетов - \$RTFL=RTUA.

9 ПРИМЕРЫ ОРГАНИЗАЦИИ СЕТЕЙ

9.1 АДРЕСНЫЙ РЕЖИМ

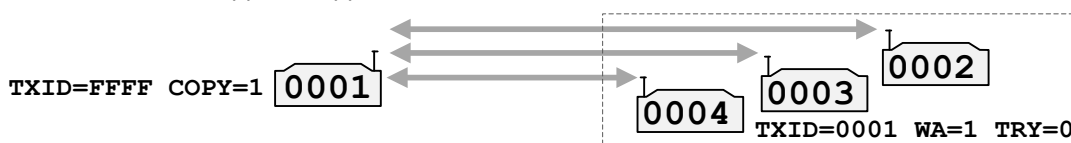
9.1.1 АДРЕСНЫЙ РЕЖИМ С ПОДТВЕРЖДЕНИЕМ.



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 равного времени передачи пакета максимального размера на текущей скорости.

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

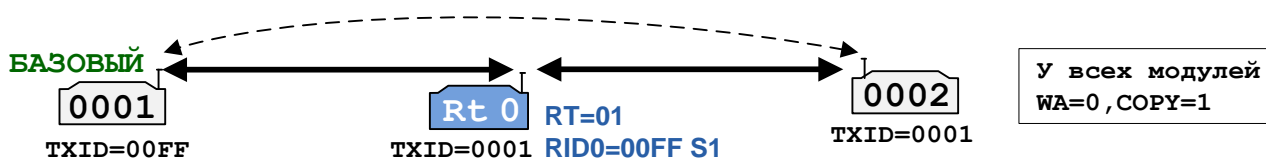
9.1.2 АДРЕСНЫЙ РЕЖИМ БЕЗ ПОДТВЕРЖДЕНИЯ.



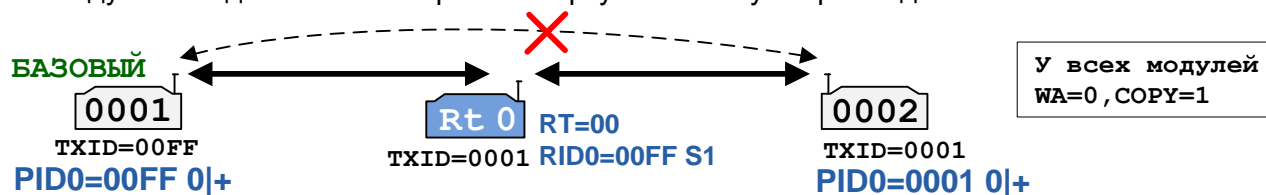
Модуль 0001 выступает в качестве "базы" (с т.з. внешнего оборудования) и работает в широкоэмитерном режиме - пакеты от него принимают все модули. Модули 002, 003 и 004 настроены на базовый модем без подтверждения. Информационные пакеты от них будут выдаваться только на порт данных базового модуля.

9.1.3 АДРЕСНЫЙ РЕЖИМ БЕЗ ПОДТВЕРЖДЕНИЯ С ПОВТОРИТЕЛЕМ.

Режим повторителя у модуля активируется установкой параметра \$RT и заданием фильтра \$RID.

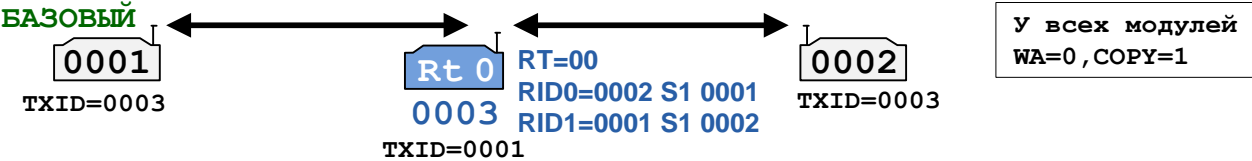


Такая конфигурация не исключает приёма модулями 0001 и 0002 «прямых» пакетов друг от друга, что может привести к избыточным пакетам в эфире. Поэтому, при вводе повторителя в систему рекомендуется устанавливать маски расширенного приема всем модулям создавая таким образом "виртуальный" путь прохождения пакетов:



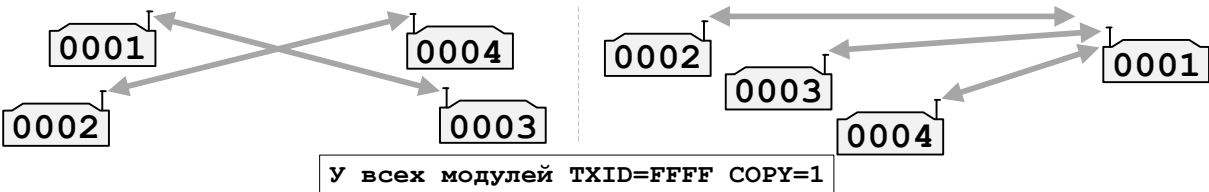
В этом случае, 0001 будет реагировать на пакеты от 0002 только, если они прошли через повторитель и будет игнорировать прямые пакеты. Тоже самое верно и для 002.

Альтернативная схема с подменой адресов при ретрансляции показана на следующем рисунке:



9.2 РЕЖИМ «ШИРОКОВЕЩАТЕЛЬНЫЙ»

9.2.1 ШИРОКОВЕЩАТЕЛЬНАЯ СЕТЬ БЕЗ ГАРАНТИИ ДОСТАВКИ СООБЩЕНИЙ

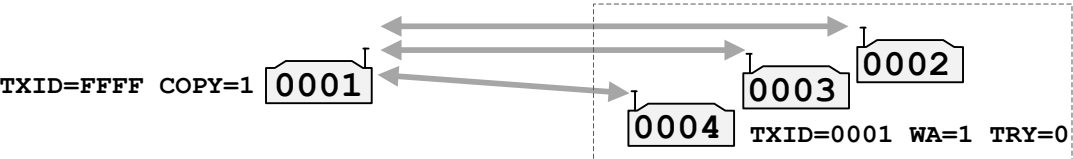


Модули 0001-0004 являются равноправными членами сети и могут принимать пакеты друг от друга.



Несмотря на то, что в данном 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**

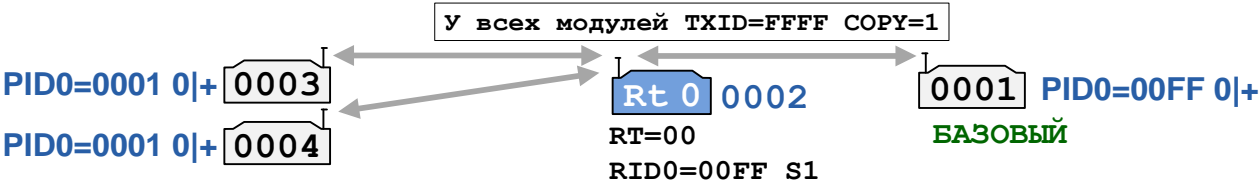
9.2.2 ШИРОКОВЕЩАТЕЛЬНАЯ СЕТЬ С ГАРАНТИЕЙ ДОСТАВКИ СООБЩЕНИЙ



Модуль 0001 работает в широковещательном режиме. Модули 0002-0004 входят в адресный режим с 0001 с гарантией доставки данных.

9.2.3 ШИРОКОВЕЩАТЕЛЬНАЯ СЕТЬ С ПОВТОРИТЕЛЕМ

Широковещательная сеть с повторителем показана на следующем рисунке (для модулей 0001, 0003 и 0004 крайне желательно установить маски расширенного приема, чтобы исключить прием прямых пакетов):



10 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 только заголовков.



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.

10.1 ШИФРОВАНИЕ AES

Шифрование AES осуществляется с помощью 16 битного ключа. Ключ задается в командном режиме с помощью команд \$KEA (в ASCII 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.

Максимальный размер полезных данных передаваемых в эфир в 1 пакете при использовании шифрования AES составляет 508 байт.

10.2 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 заголовок пакета. Если на удаленном модуле ключ SK будет задан неверно, то у заголовка пакета не совпадет контрольная сумма и пакет не будет декодироваться. Данное свойство будет полезно, когда в пределах радиовидимости текущей сети появляется аналогичная другая. Это может исключить некорректное поведение, например, встроенного IO модуля модуля.

11 Модуль ввода вывода (IO)

11.1 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.

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**

11.2 Входы IN1...IN4

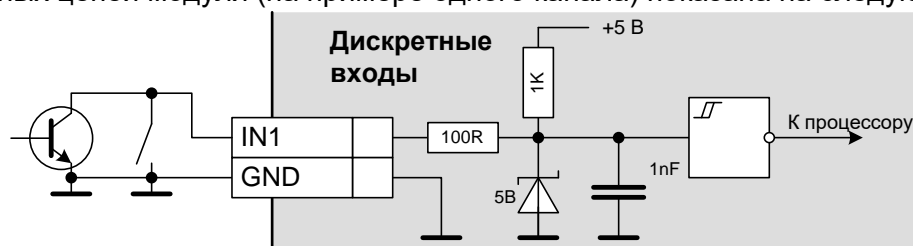
К четырем парам клемм IN1...IN4 подключаются внешние дискретные датчики типа «сухой контакт» (выключатели, кнопки, концевые выключатели, контакты реле или датчики различного типа с транзисторным выходом) или транзисторные ключи n-p-n типа с открытым коллектором.

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.



11.3 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 В и С и размыкание клемм В и А).

Выходы реле модуля гальванически развязаны между собой и внутренними цепями.

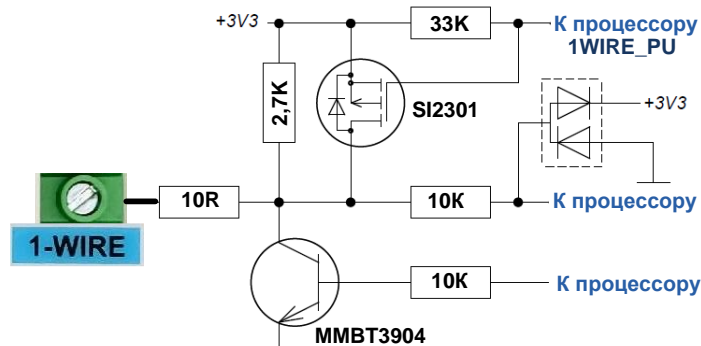
11.4 ТЕСТ ВХОДОВ/ВЫХОДОВ

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`.

11.5 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

- 8) Тип датчика Температура

butn dev: |———— Список ключей iButton

- 1) 0EF3548B
2)
3) Серийный
4) код ключа

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.

11.6 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.

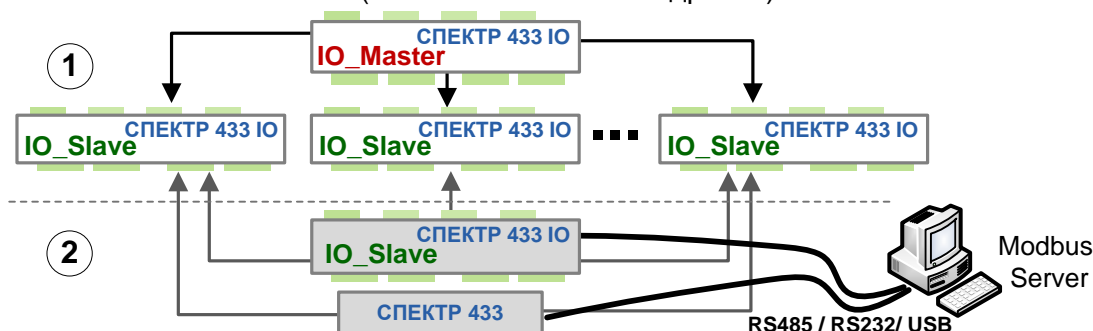
11.7 РЕЖИМ IO_SLAVE

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 433 IO» в режиме IO_Slave. При этом в качестве «мастера» используется модуль «Спектр 433 IO» в режиме IO_Master.

На рис.2 в качестве «мастера» может быть использован Modbus сервер подключенный по активному порту либо к модулю «Спектр 433 IO» в режиме IO_Slave, либо к радиомодему «Спектр 433». В таком варианте, число опрашиваемых модулей «Спектр 433 IO» в режиме IO_Slave может быть более 8 (вся емкость modbus адресов).



11.8 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:

- \$IOF15** Набор индексов модулей IO_Slave которым будет посылаться запрос на установку **ВЫХОДОВ** функцией 0Fh. Например, значение 310 означает, что IO_Master будет генерировать запросы функцией F15 с адресами IOADR+0, IOADR+1 и IOADR+3;
- \$IOF15BC** 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;
- \$IOF02** Набор индексов модулей IO_Slave которым будет посылаться запрос на чтение **ВХОДОВ** функцией 02h. Смысл аналогичен набору параметра \$IOF15;
- \$IOF23** 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.

- \$IOQPT** 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.
По умолчанию, опрос всех модулей ведётся по радио.
- \$IOPT** Пауза между опросами (Pause Time). Задаёт паузу в секундах после опроса очередного модуля IO_Slave.
- \$IOWA** Время, в течении которого IO_Master ожидает ответа (Wait Answer) от IO_Slave. Время выражается в числе пакетов максимального размера на текущей скорости. Например, на скорости AR=3 время передачи пакета максимального размера составляет 715 мс. По умолчанию, параметр IOWA =1.
- \$IOSMx** 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.
- \$IOMXO** 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:

- \$IOF23 = 1** | Включена функция F23 (чтение-установка) за 1 запрос
- \$IOPT = 1** | Пауза после опроса очередного модуля 1с
- \$IOQPT = 11111111** | Опрос по радио
- \$IOF15 =** | Функции F15 не нужна (используется "оптимизированная" функция F23)
- \$IOF02 = 210** | Задаем в соответствии с задачей применение функции "02 Read Input

\$IOSM0 = 0001
\$IOSM1 = 0010
\$IOSM2 = 0100

Status" (чтение состояние дискретных входов) для всех 3-х удаленных модулей с индексами 0,1 и 2.

Задаем игнорирование состояний всех входов, кроме IN1, для удаленного модуля n0, всех входов, кроме IN2, для удаленного модуля n1 и всех входов, кроме IN3, для удаленного модуля n2.

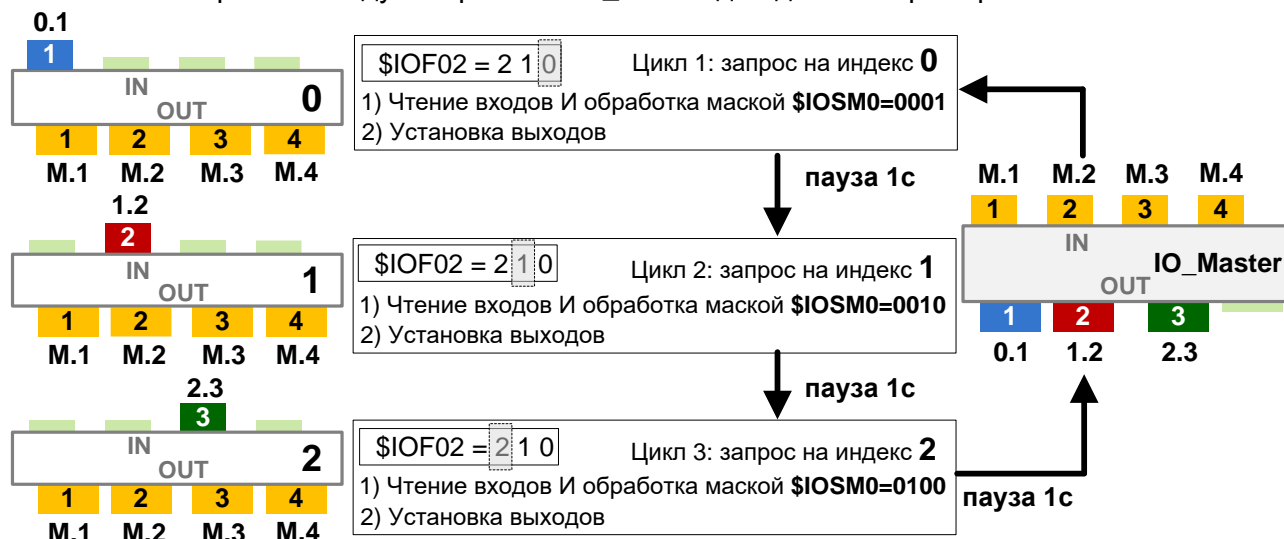
Теперь:

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:

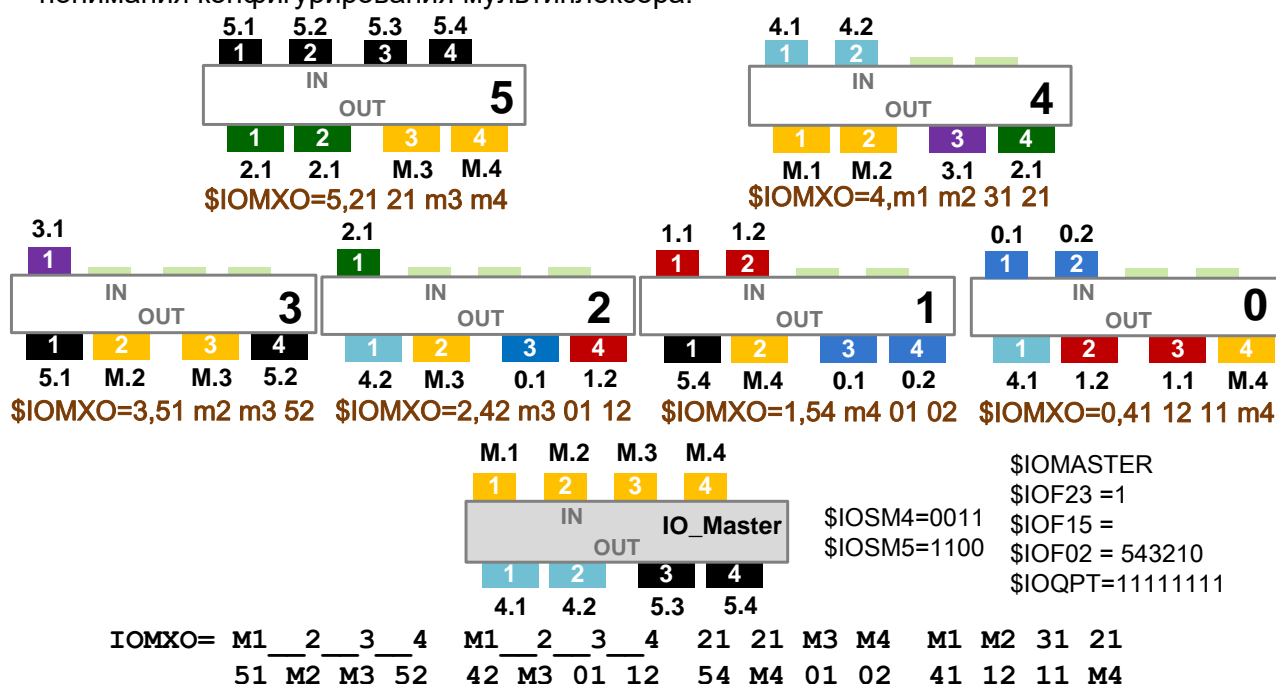


Таким образом, полный цикл работы модуля IO_Master будет состоять из 3х запросов и 3х ответов. Если выключить флаг IOF23 и установить соответствующий параметр IOF15 для установки выходов, то цикл бы состоял уже из 12 пакетов (6 запросов и 6 ответов).

После передачи очередной команды модуль IO_Master выдерживает паузу 1 секунду.

11.9 РЕЖИМ IO_MASTER. МУЛЬТИПЛЕКСИРОВАНИЕ ВХОДОВ НА ВЫХОДЫ.

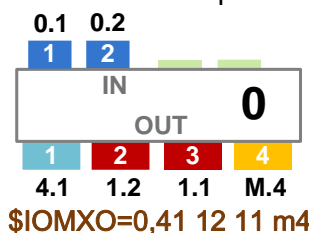
В режиме IO_Master имеется возможность мультиплексировать входы одних модулей IO_Slave на выходы других модулей IO_Slave участвующих в опросе. Мультиплексирование выходов самого модуля IO_Master недоступно. Далее приведена абстрактная схема, для понимания конфигурирования мультиплексора:



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,".
- 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 ЭНЗУ 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](#)").

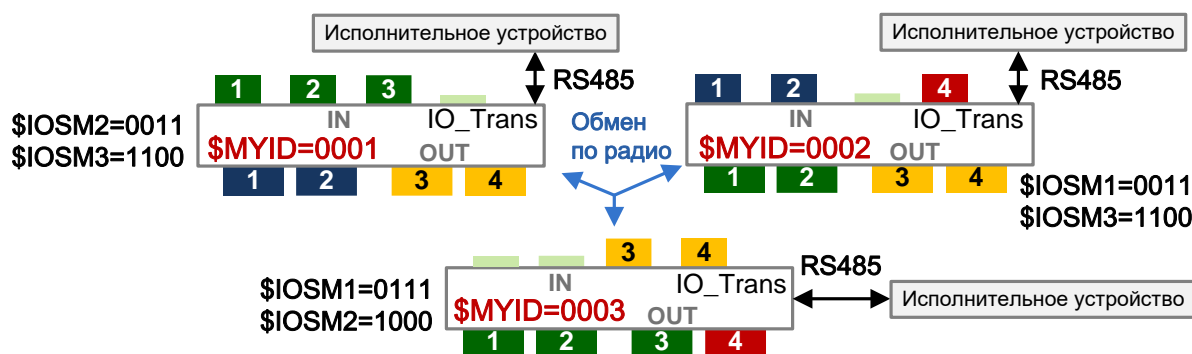
11.10 РЕЖИМ IO_TRANS

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, при получении пакетов выходы не переключали свое состояние от разных модулей, осуществляется фильтрация адресов с помощью масок \$IOSMx.

Номер в маске \$IOSMx соответствует младшей цифре \$MYID удаленного модуля. Таким образом, принимать значение входов можно только от модулей, у которых значения \$MYID равны от xxx0 до xxx7 (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=0001 \$IOSM2=0011 от модуля с MYID=0002 выставить выходы OUT1 и OUT2;
 \$IOSM3=1100 от модуля с MYID=0003 выставить выходы OUT3 и OUT4;

MYID=0002 \$IOSM1=0011 от модуля с MYID=0001 выставить выходы OUT1 и OUT2;
 \$IOSM3=1100 от модуля с MYID=0003 выставить выходы OUT3 и OUT4;

MYID=0003 \$IOSM1=0111 от модуля с MYID=0001 выставить выходы OUT1,OUT2 и OUT3;
 \$IOSM2=1000 от модуля с MYID=0002 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**.

11.11 БЕЗОПАСНОЕ СОСТОЯНИЕ ВЫХОДОВ

В процессе обмена между модулем IO_Master и IO_Slave может возникнуть ситуация, когда 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.

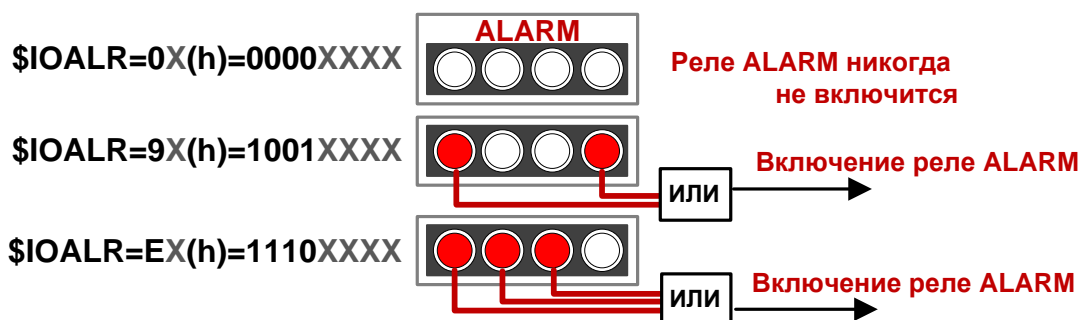
С безопасным состоянием выходов связаны следующие регистры:

\$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

11.11.1 ИНДИКАТОРЫ И РЕЛЕ ALARM.

Модули имеют 4 индикатора и выход (реле) ALARM (авария), которые используются для отображения отсутствия связи. Под отсутствием связи подразумевается отсутствие в течение заданного времени либо запросов от мастера (в режиме IO_Slave или IO_Trans), либо 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.



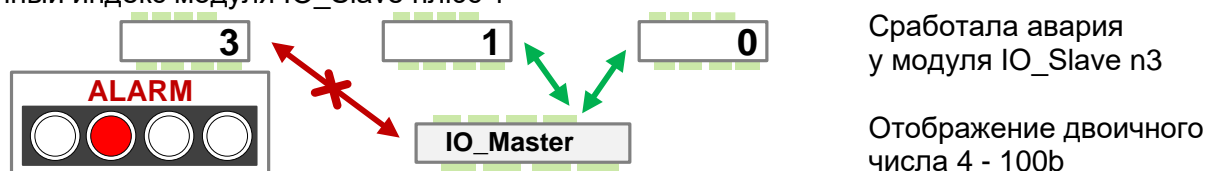
При включении питания, режим модуля кратковременно отображается на индикаторах ALARM (само реле ALARM при этом не переключается):

| 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.

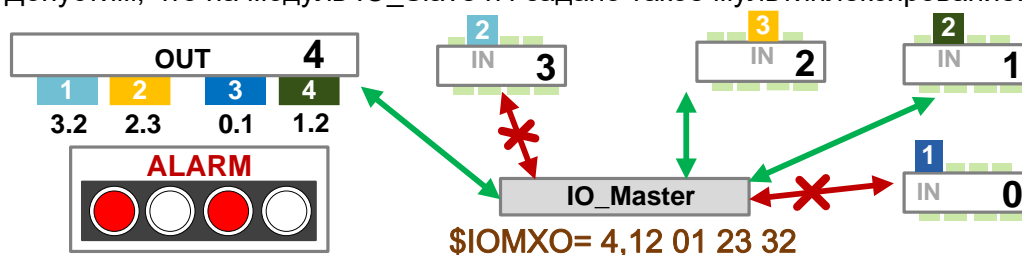
Режим аварии задается в младших 4-х битах регистра $\$IOALR$.

$\$IOALR[3:0] = 0$. 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 загорятся 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.

11.11.2 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

11.11.3 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.

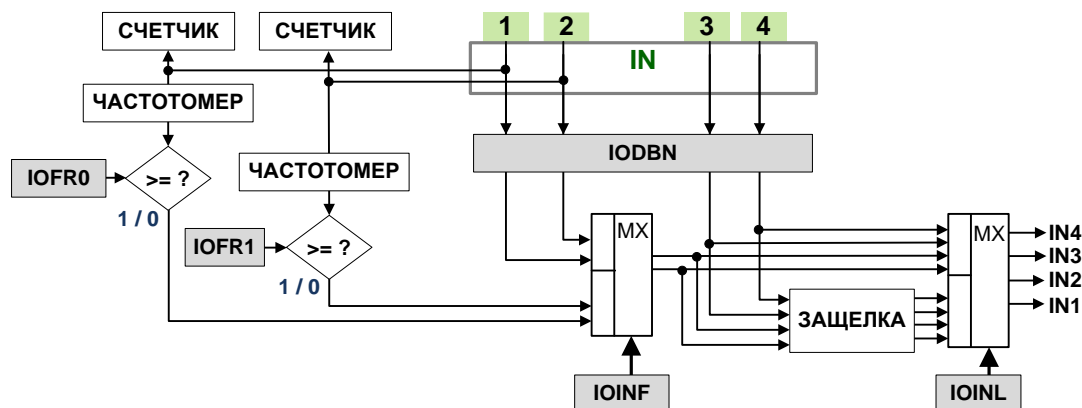
Разрешенные выходы в регистре \$IOSFV устанавливаются в безопасное состояние согласно индексу модуля IO_Slave, с которым пропала связь. Например, если пропала связь с модулем n1, то в безопасное состояние перейдет только выход OUT2 (если он разрешен).



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.

11.12 Функции входов

Структурная схема входов IN1-IN4 представлена на следующем рисунке:



Входы IN1 и IN2 имеют функции счета. Все входы проходят через функцию устранениядребезга 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 **подавлениедребезга контактов не осуществляется** - входы "считываются" непосредственно с клемм.

11.12.1 "ДРЕБЕЗГ" КОНТАКТА.

Устранениедребезга 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 подавлениедребезга). 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.

11.12.2 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.

11.12.3 ЗАЩЕЛКА.

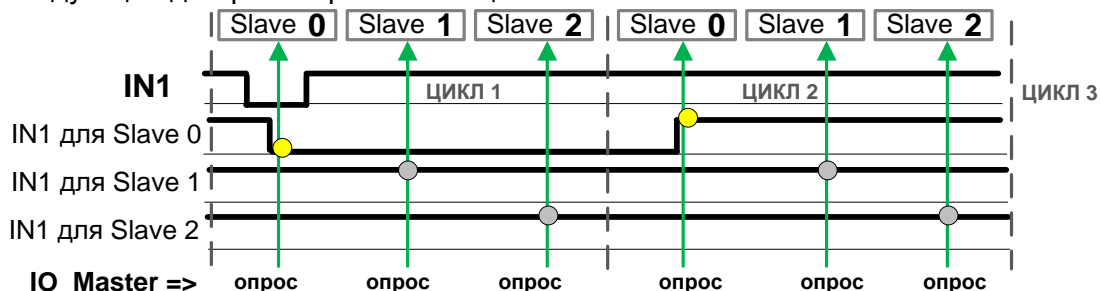
Под функцией "защелка" понимается способность запоминание перехода уровня на входе из "1" в "0" или из "0" в "1". Последующие изменения уровней не меняют значения входа до тех пор, пока текущее состояние не будет передано в режиме IO_Master или не будет считано в режиме IO_Slave.

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 по времени он меньше полного цикла опроса всех устройств в режиме IO_Master или возникает не в момент опроса в режиме IO_Slave.

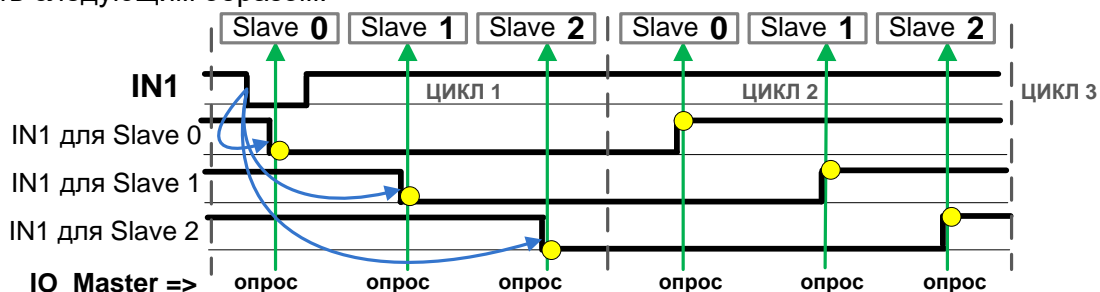
Проиллюстрируем работу IO_Master в режиме защелки.

На следующей диаграмме режима защелки нет:



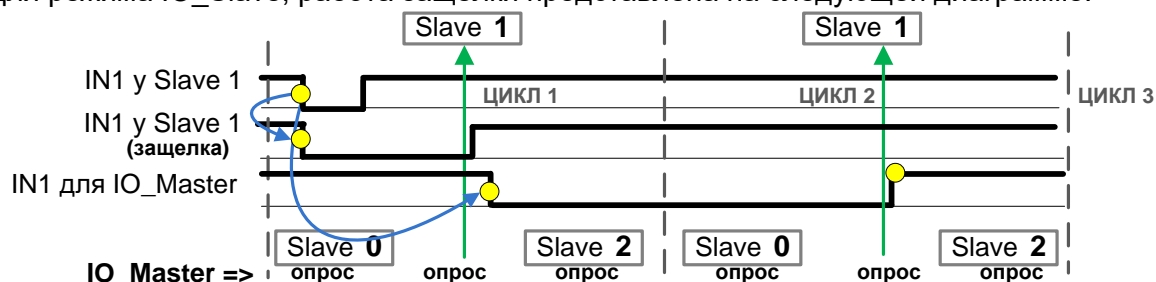
Короткий импульс '1' -> '0' возник в момент опроса IO_Slave n0 и значение вернулось в '1' сразу же после опроса. IO_Master "успел" передать текущее мгновенное состояние IN1 только одному IO_Slave. Для остальных устройств, значение входа IN1 не изменилось, соответственно соответствующий OUTx у них также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

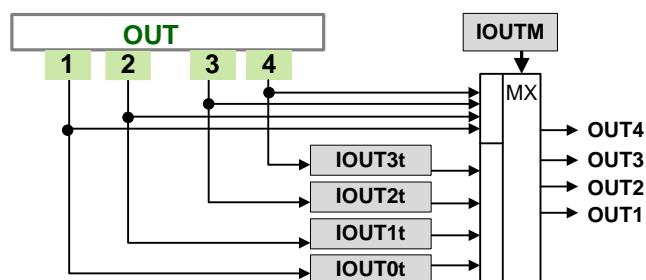


Может возникнуть ситуация, когда IO_Slave получил запрос от IO_Master, сформировал ответ и сбросил защелку, но ответ до IO_Master не прошел. В данной ситуации изменения значения IN1 IO_Slave для модуля IO_Master **НЕ БУДЕТ** (как будто режим защелки у IO_Slave отсутствует)

11.13 Функции выходов

Структурная схема выходов OUT1- OUT4 представлена на рисунке.

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



11.13.1 Выходной импульс.

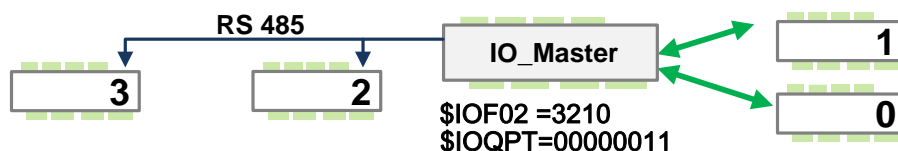
Помимо установки на выходах постоянного уровня, возможно формирование импульса заданной длительности после получения команды на установку выходов. В зависимости от конфигурации, импульс может быть положительным (переход из '0' в '1'), так и отрицательным (переход из '1' в '0'). Активация режима импульса 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.

11.14 Опрос по RS485

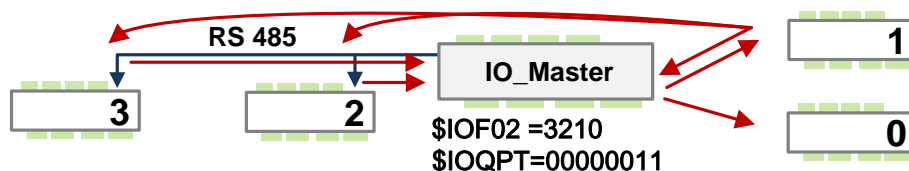
Модуль может опрашивать удаленные модули как по радиоэфиру, так и по интерфейсу RS485. Способ опроса задается в регистре \$IOQPT. Чтобы модуль в режиме IO_Master опрашивал модуль IO_Slave по RS485, необходимо установить соответствующий бит в 0.

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.

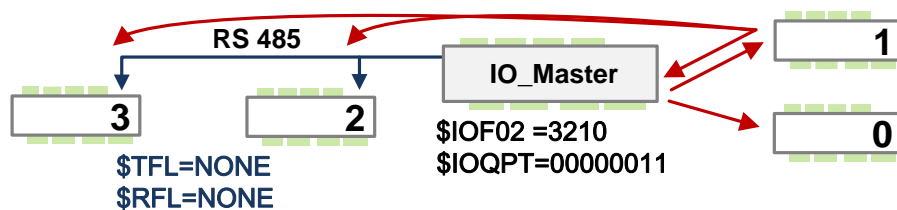
Рассмотрим опрос модуля 1 по эфиру из примера выше:



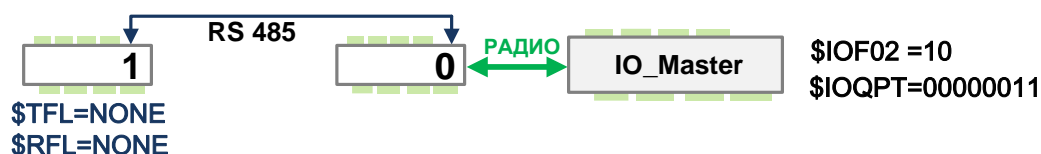
Красными стрелками показано "движение" запроса от IO_Master к модулю n1 и его ответ. Модуль n1 получив по эфиру "свой" запрос, отвечает также по эфиру. Ответ принимают все модули в сети, включая модули n3 и n2. Если получение и выдача ответа модулем n0 на свой порт RS485 не критично, т.к. его порт никуда 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:

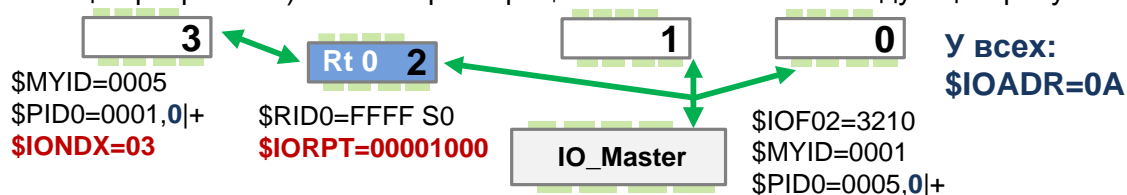


Модули с n0 и n1 опрашиваются по радио которые соединены друг с другом по RS485. Для модуля n1 также нужно отключить прием/передачу из эфира.

11.15 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:



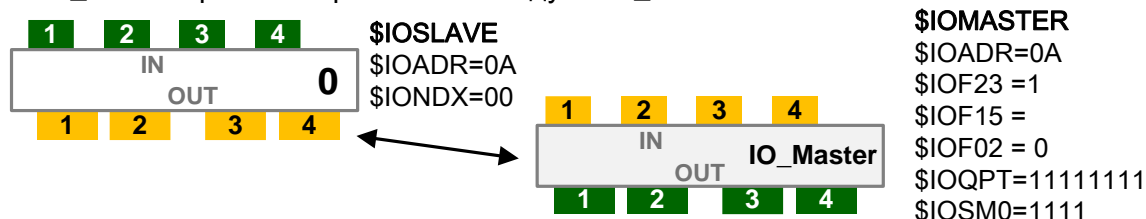
Допустим, модуль IO_Master опрашивает 4 модуля IO_Slave. Модули n0-n2 находятся в прямой радиовидимости, а с модулем n3 связь не стабильна. Модуль n2 назначен ретранслятором (\$RT=00). Если фильтр разрешен полностью (по умолчанию), то модуль n2 будет ретранслировать абсолютно все пакеты, даже для модулей 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.

11.16 ТИПОВЫЕ СХЕМЫ ВКЛЮЧЕНИЯ

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

В схеме работают 2 модуля - один IO_Master, другой 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.

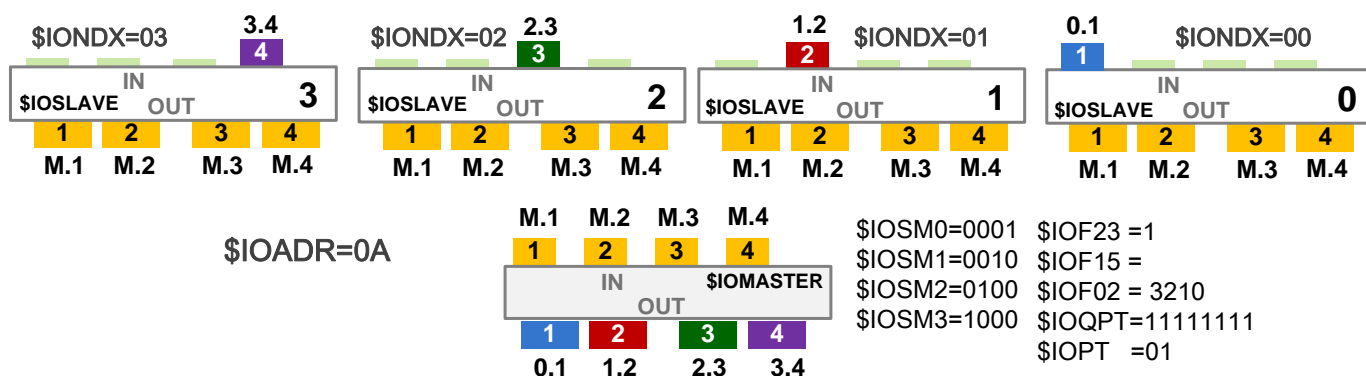
При такой схеме модули IO_Slave и IO_Master работают полностью симметрично.

11.16.2 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 радиоэфиру 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.

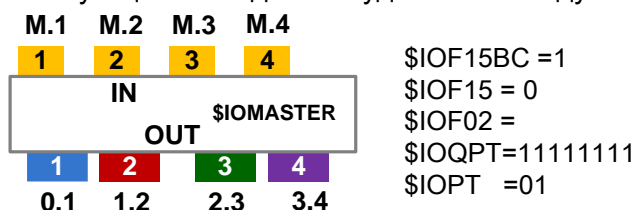
Поскольку у модуля имеются только четыре выхода, суммарное количество транслируемых на них удаленных входов не может превышать четырех (иначе входы одного удаленного модуля начнут "перебивать" входы другого). Поэтому трансляцию состояния входов от четырех модулей IO_Slave на один IO_Master можно обеспечить 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.

11.16.3 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 радиоканале, 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.

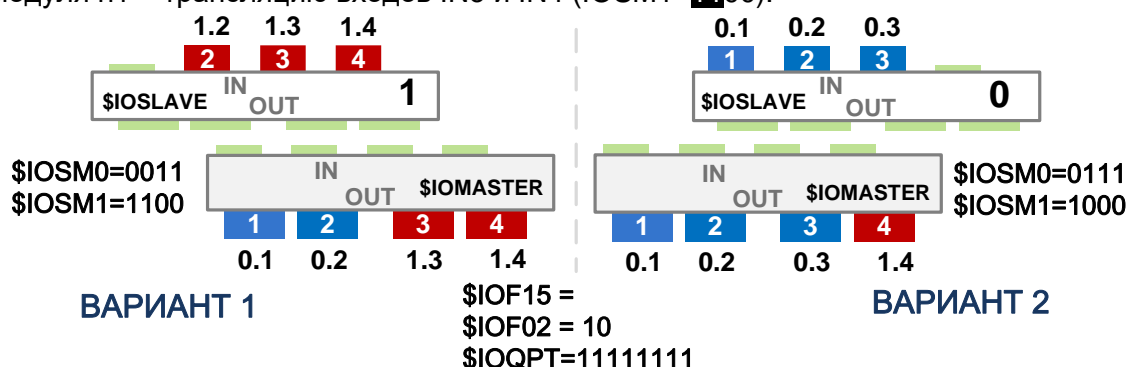
11.16.4 ТРАНСЛЯЦИЯ НА ВЫХОДЫ ОДНОГО МОДУЛЯ ВХОДОВ НЕСКОЛЬКИХ УДАЛЕННЫХ МОДУЛЕЙ

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).

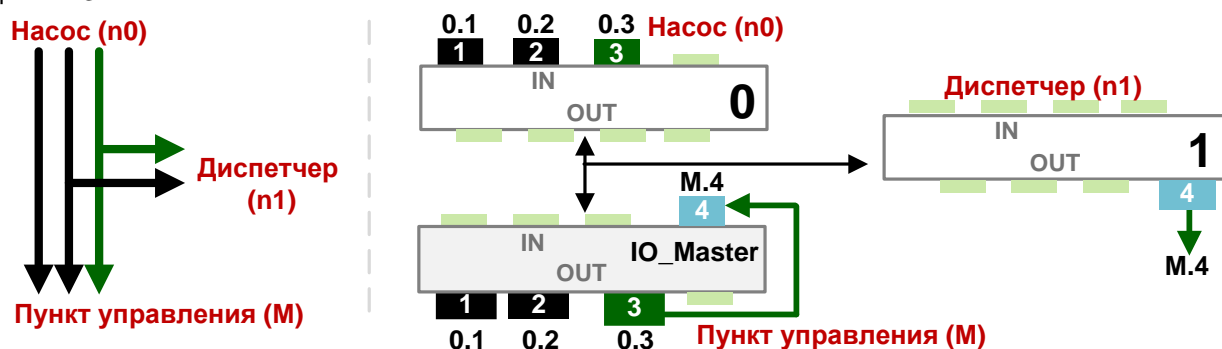


Теперь входы IN1 и IN2 модуля n0 будут управлять выходами OUT1 и OUT2 мастера, а входы IN3 и IN4 модуля n1 – выходами OUT3 и OUT4 мастера (**ВАРИАНТ 1**).

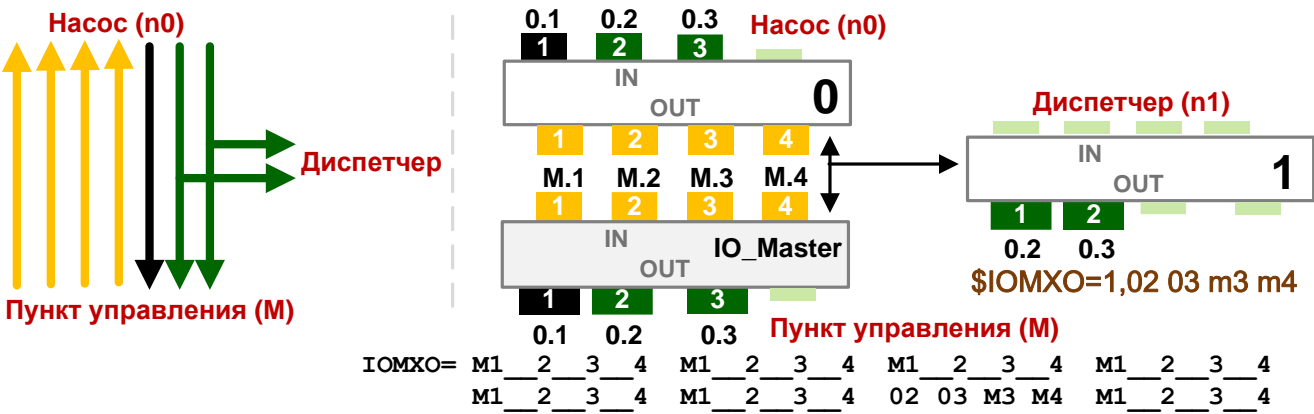
Очевидно, данную схему легко модифицировать для трансляции других комбинаций входов удаленных модулей IO_Slave – нужно просто изменить маски IOSM0 и IOSM1. Например, можно транслировать три входа с первого модуля и один – со второго (IOSM0=0111, IOSM1=1000) (**ВАРИАНТ 2**).

11.16.5 МУЛЬТИПЛЕКСИРОВАНИЕ ВЫХОДОВ

Рассмотрим ситуацию, когда из пункта управления (М) необходимо получать 3 аварийных сигнала от насоса n0 и один сигнал 0.3 передавать в диспетчерскую n1. Данная задача может быть решена путем соединения проводом неиспользуемого входа M.4 с выходом M.3:



Усложним задачу - из пункта управления необходимо передавать на насос 4 управляющих сигнала, а от насоса передавать 2 аварийных сигнала также в диспетчерскую. Ввиду ограниченного числа входов/выходов у модулей эта задача не решается в "стандартной" конфигурации, однако с помощью мультиплексирования она легко решается:



11.16.6 УПРАВЛЕНИЕ МОДУЛЯМИ IO_SLAVE ВНЕШНИМ "МАСТЕРОМ"

Пожалуй, одна из распространенных задач при использовании модулей "Спектр 433 IO" – 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 433").

Таким образом, для встраивания модулей в систему сбора/управления достаточно установить на стороне мастера (ПЛК, OPC сервер и т.д.) радиомодем "Спектр 433". Управление модулями "Спектр 433 IO" 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 433 IO" нужно запрограммировать в режим IO_Slave и присвоить им уникальные Modbus адреса. Мастер системы будет работать с модулями по этим адресам через радиомодем "Спектр 433".

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

12 УДАЛЕННОЕ КОНФИГУРИРОВАНИЕ

В модуле имеется возможность удалённого конфигурирования по радио. Для понимания процесса удалённого конфигурирования вводится понятие *профиль* - набор всех параметров модуля, необходимых для его работы. Профиль хранится в ЭНОЗУ модуля.

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

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 и значения фильтров.

12.1 УДАЛЁННОЕ КОНФИГУРИРОВАНИЕ В КОМАНДНОМ РЕЖИМЕ

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

- \$RPRF - чтение профиля модуля;
- \$WPRF - запись профиля модуля;
- \$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 командой \$RPRF. Далее, с помощью команды \$DMPR узнать текущую конфигурацию.



Команда \$DMPR выводит результат всегда - в не зависимости считан профиль удаленного модуля или нет. Если профиль не был считан ранее, команда будет выводить **"мусор"** вместо 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 команды \$WPRF после получения которой, удаленный модуль автоматически перезагрузится через 5 сек.

Пример удалённого конфигурирования:

```

OK> $RPRF 0002           чтение профиля удаленного модема с адресом 0002
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 0002           запись профиля обратно в модем 0002
Press 's' to stop...
OK>
** profile write done.   Модем 0002 перезагрузится через 5 с
OK>

```



При вводе команды \$WPRF модуль в автоматическом режиме посылает удаленному модулю данную команду до тех пор, пока ответ не будет получен. Если удаленный модуль недоступен, запросы будут слаться бесконечно. Для остановки процесса необходимо нажать клавишу 's'.

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.

12.2 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](#)".

13 ТЕСТИРОВАНИЕ СВЯЗИ.

Тестирование связи между модулями возможно следующими способами:

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

13.1 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=0002 (0A:01) ,v1.01 (F) ,RSSI=-87,-82
** tx
** 2: ID=0005 (0A:02) ,v1.01 (F) ,RSSI=-109,-96
** tx
** tx
OK>

```

Версия ПО
ID устройства: F - СПЕКТР 433 IO V2
Уровень сигнала запроса
Уровень сигнала ответа
IOADB : IOADR
MYID удаленного модема
RPT=5:0
Номер повторителя (если активен)

\$RCHK - опрос удаленного модуля с заданным параметром \$MYID:

```

OK> $RCHK 0002
press 's' to stop...
OK>
** v1.01 (F) ;Rt=525 ms,RSSI=-42,-40
** v1.01 (F) ;Rt=550 ms,RSSI=-42,-40
** v1.01 (F) ;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 0002 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

13.2 ТЕСТИРОВАНИЯ СВЯЗИ В ПРОЗРАЧНОМ РЕЖИМЕ

В прозрачном режиме передачи данных необходимо послать в активный порт локального модуля ASCII строку без пауз между символами вида \$hhhhPING_RATEOSpm, где **hhhh** - адрес удаленного модуля, **p** - мощность (0...3), на которой удаленный модуль будет передавать ответ, **m** - мультипликация посылки (0...1). 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 \$hhhhPING_RATEOSpm удаленный модуль с адресом **hhhh** переключает текущую мощность на значение 'p' (если задано) и задает мультипликацию ответа 'm'. 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_RATEOS00 (при получении, передавать ответ на PWR=0)

Ответ: \$0002PING_v01.00_ID:0F,0208,-049,-045

В сети обнаружен модуль 0002. Версия ПО 1.00. ID устройства 0F (СПЕКТР 433IO V2). 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_RATEOS11 (при получении, передавать ответ на PWR=1)

Ответ: \$0002PING_v01.00_ID:0F,0208,-049,-045

\$0003PING_v01.00_ID:0F,0108,-049,-045

В сети присутствуют два модуля с адресами 0002 и 0003.

При групповом запросе обнаружение всех модулей не гарантируется.

Запрос: \$1234PING_RATEOSx0 (при получении, не изменять мощность)

Ответ: \$1234PING_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.

13.3 РЕЖИМ МАЯК

В режиме маяка, модуль автоматически, выдает в эфир строку \$hhhhPING_RATEOSpm. Где hhhh - адрес удаленного модуля, который задается регистрами \$RG04-05, 'p' - текущая мощность модуля в профиле (0-3), а 'm' - параметр мультипликации (0-1). 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.

14 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 \$EXCHP . |
| 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 | - | Резерв |
| 15 | dbg_dump_cfg (0xFF) | Печать пользовательских данных на отладочном порту. см. раздел " Отладочный порт ". |

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

15.1 \$DMP(R) — ВЫВОД ПРОФИЛЯ (УДАЛЁННОГО) МОДУЛЯ

Ввод: **\$DMP** или **\$DMPR**

Пример:

```
OK> $DMP
P2 TXID=FFFF COPY=1 ,MYID=0004      ОБЩИЙ ПРОФИЛЬ
CH=92 PWR=0 COM=9600,8N1
AR=03 LBT=0 DAT=EOT PACT=005 PSL=000 PL=7
MN=3  DM=0 FEC=- RCONC=0 TCONC=0
IOADR=0A+IONDX=00 (0Ah) TFL=----- RT=-
IORPT=11111111 RFL=----- RTFL=-----
MaxTt=522 ms *SK,*KEY,*LID
-----
IOSLAVE IOALR=F0
IOSFT=030 IOSFV=-----
IOINL=----- IOBN=00000000h      ПРОФИЛЬ МОДУЛЯ
IOINF=00 IOFR[1:0] =0000,0000      ВВОДА-ВЫВОДА
IOUTM=----- IOUT[3:0] t=0000,0000,0000,0000
IOINT=00000000
IOMOD=00000000
OK>
```

В выводе команды отображаются следующие "подсказки":

- MaxTt=** максимальное время передачи пакета при выбранных параметрах \$AR, \$MN и \$FEC;
- *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.

15.2 \$IEE — СБРОС ПАРАМЕТРОВ ПО УМОЛЧАНИЮ

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

После ввода команды происходит инициализация параметров, хранящихся в ЭНОЗУ значениями по умолчанию. Адрес MYID модуля становится равным hhhh и модуль автоматически пересбрасывается.



Команда \$IEE на 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**);

15.3 \$CH — ИЗМЕНЕНИЕ РАБОЧЕГО ЧАСТОТНОГО КАНАЛА

Ввод: **\$CH=dd**, где dd - номер рабочего канала 01...69, 92

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

15.4 \$PWR — УСТАНОВКА МОЩНОСТИ ПЕРЕДАТЧИКА

Ввод: **\$PWR=d**, где d:

| PWR | Значение мощности, мВт | |
|-----|------------------------|--------------|
| 0 | 10 | по умолчанию |
| 1 | 150 | |
| 2 | 350 | |
| 3 | 550 | |

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

15.5 \$AR — СКОРОСТЬ ПЕРЕДАЧИ ДАННЫХ ПО ЭФИРУ

Ввод: **\$AR=dd**, где **dd**- значение 0-13.

| AR | Скорость, бит/с | Девияция, кГц | |
|----|-----------------|---------------|--|
| 0 | 4800 | ±5 | |
| 1 | 4800 | ±20 | |
| 2 | 9600 | ±10 | |
| 3 | 9600 | ±20 | по умолчанию |
| 4 | 19200 | ±20 | После ввода команды, модуль сразу перенастраивается на новую скорость и сохраняет ее в ЭНОЗУ |
| 5 | 19200 | ±40 | |
| 6 | 38400 | ±40 | |
| 7 | 38400 | ±100 | |
| 8 | 57600 | ±100 | |
| 9 | 76800 | ±100 | |
| 10 | 4800 | ±2.4 | Для совместной работы с радиомодемом СПЕКТР 9600 GM |
| 11 | 9600 | ±4.8 | |
| 12 | 14400 | ±7.2 | |
| 13 | 19200 | ±9.6 | |

При выборе варианта следует иметь ввиду, что большее значение девиации делает связь более надежной, но при этом расширяется полоса частот, занимаемая сигналом в эфире.

15.6 \$MN — ЧИСЛО БИТ, ПЕРЕДАВАЕМЫХ БЕЗ ИНВЕРСИИ ПОСЛЕДНЕГО БИТА

Ввод: **\$MN=d** где **d**: Инверсия после:
0 - каждого бита. (код Манчестер 2)
1 - каждого 6-го бита
2 - каждого 15-го бита
3 - каждого 10-го бита (**по умолчанию**)

15.7 \$LBT — АЛГОРИТМ ВЫХОДА В ЭФИР

Ввод: **\$LBT=d** где **d**: **0** - задержки перед передачей данных в эфир отсутствуют;
1 - "длинный" алгоритм выхода в эфир;
2 - "короткий" алгоритм выхода в эфир;

15.8 \$P2/P1 — ВЫБОР ПРОТОКОЛА ПРИ ПЕРЕДАЧЕ

Ввод: **\$P1** - протокол работы для совместимости с линейкой модемов Спектр 433 V1.
\$P2 - протокол для работы в режиме V2. (**по умолчанию**)

15.9 \$MYID — ИЗМЕНЕНИЕ СОБСТВЕННОГО АДРЕСА МОДУЛЯ

Ввод: **\$MYID=hhhh**, где **hhhh** — любое значение, кроме FFFF, FFxx, xxFF или 0000.

15.10 \$TXID — ИЗМЕНЕНИЕ АДРЕСА ВЫЗЫВАЕМОГО МОДУЛЯ

Ввод: **\$TXID=hhhh**, где **hhhh** — любое значение, кроме 0000

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

Ввод значений вида FFFF, FFxx или xxFF означает широковещательный (групповой) режим передачи данных иначе - адресный режим с модулем, чей адрес MYID совпадает с введенным значением hhhh.

15.11 \$WA — ОЖИДАНИЕ ПОДТВЕРЖДЕНИЯ В АДРЕСНОМ РЕЖИМЕ

Ввод: **\$WA=h**, где **h** — шестнадцатеричное число (0-F).

Задается в числе пакетов максимального размера на текущей скорости. Значение 0 означает отсутствие ожидания подтверждения. Если по истечении времени WA после передачи пакета не получено подтверждения о доставке от адресуемого модуля, отправка пакета повторяется.



Несмотря на то, что данный параметр является внутренним параметром модуля **у внешнего оборудования** также необходимо выставлять корректное значение ожидания ответа. Время передачи пакета данных максимальной длины на текущей скорости отображается в результате вывода команды \$DMP в строке "MaxTt=".

15.12 \$TRY — ЧИСЛО РЕТРАНСЛЯЦИЙ ПАКЕТОВ, ПРИ ОТСУТСТВИЯ ПОДТВЕРЖДЕНИЯ

Ввод: **\$TRY=d**, где d — число от 0 до 7. Значение 0 соответствует бесконечному числу попыток передать пакет, в случае отсутствия подтверждения в адресном режиме.

15.13 \$COPY — МАКСИМАЛЬНОЕ ЧИСЛО ОДИНАКОВЫХ ШИРОКОВЕЩАТЕЛЬНЫХ ПАКЕТОВ

Ввод: **\$COPY=d**, где d число от 1 до 7.

Для уменьшения вероятности потери данных в ширококвещательном режиме, когда подтверждения о доставке отсутствуют, можно последовательно передавать несколько копий ширококвещательного пакета. При получении адресатом дублируемые ширококвещательные пакеты отбрасываются.

15.14 \$PAUSE — ВРЕМЯ МЕЖДУ ПЕРЕДАЧАМИ КОПИЙ ШИРОКОВЕЩАТЕЛЬНЫХ ПАКЕТОВ

Ввод: **\$PAUSE=dd**, где dd значение от 00 до 99.

Шаг 100 мс. Значение 00 соответствует 05.

Параметр активизируется только в ширококвещательном режиме и задает время между последовательной передачей копий (параметр \$COPY > 1) ширококвещательного пакета.

15.15 \$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' - включено, 'N' - выключено

Пример: **\$FEC=0I**

\$FEC=5N

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

15.16 \$DM — МУЛЬТИПЛИКАЦИЯ ДАННЫХ

Ввод: **\$DM=b** где b: **0** - мультипликация запрещена;
1 - мультипликация разрешена.

15.17 \$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

15.18 \$DAT — ТИП ПРОТОКОЛА ВХОДЯЩИХ ДАННЫХ

Ввод: **\$DAT=EOT** (по умолчанию)

\$DAT=EOC

\$DAT=RTU

\$DAT=DL8

15.19 \$PL — МАКСИМАЛЬНОЕ ЧИСЛО ПАКЕТОВ В БУФЕРЕ

Ввод: **\$PL=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.

15.20 \$PACT — ТАЙМ-АУТ ПРИЕМА ПАКЕТА В БУФЕР НА ПЕРЕДАЧУ

Ввод: **\$PACT=ddd**, где ddd — число от 000 до 255. Минимальное значение 002.

Тайм аут задается числом символов на текущей скорости порта данных.

15.21 \$PSL — НАРЕЗКА ВХОДЯЩИХ ДАННЫХ

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

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

15.22 \$EOC — СИМВОЛ ПЕРЕДАЧИ ДАННЫХ

Ввод: **\$EOC=hhc**, где

hh - Символ передачи данных

c - признак передачи самого символа: **'T'** - передавать, **'N'** - не передавать.

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

15.23 \$RCONC — КОНКАТЕНАЦИЯ НА ПРИЕМ

Ввод: **\$RCONC=b**, где

b - конкатенация: **0** - запрещена, **1** - разрешена.

15.24 \$TCONC — КОНКАТЕНАЦИИ НА ПЕРЕДАЧУ

Ввод: **\$TCONC=d**, где **d** - уровень конкатенации (байт):

- 0 - конкатенация запрещена
- 1 - 32
- 2 - 64
- 3 - 96
- 4 - 128
- 5 - 160
- 6 - 192
- 7 - 255

15.25 \$RT — НОМЕР ПОВТОРИТЕЛЯ МОДУЛЯ

Ввод: **\$RT=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,diy:h**

15.26 \$RID — АДРЕС ПАКЕТА РАЗРЕШЕННОГО ДЛЯ РЕТРАНСЛЯЦИИ

Ввод: **\$RIDd=hhhh tb [dddddddddS₁S₂]**, где

d Номер ячейки от 0 до 7;

hhhh адрес, или маска адреса;

t Тип адреса: **'S'** - hhhh является адресом отправителя (**Source**), **'D'** - hhhh является адресом получателя (**Destination**)

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

0 - запрещено **1** - разрешено

Не обязательные поля:

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

S₁ символ логической операции между РП пакета и маской dddddddd:

«&» операция «И»;

«|» операция «ИЛИ».

S₂ действие над пакетом в случае совпадения РП пакета с маской dddddddd:

«+» Пакет может быть ретранслирован (после анализа RIDx)

«-» Пакет не будет ретранслирован (после анализа RIDx)

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

Примеры:

\$RID0=1234 S1 014&+

Адрес записывается в ячейку 0. Ретранслируются пакеты, в заголовке которых адрес отправителя равен 1234. Пакеты с нулевыми значениями РП разрешены для повторения. Пакет повторяется только в том случае, если он прошел через ретрансляторы 0 **И** 1 **И** 4.

\$RID7=03FF D0 401|+

Адрес записывается в ячейку 7. 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.

\$RID6=0123 S0

Адрес записывается в ячейку 6. Ретранслируются пакеты, в заголовке которых адрес отправителя равен 0123. Пакеты с нулевыми значениями РП не разрешены для повторения; это означает, что пакет уже должен был быть повторен каким-либо другим ретранслятором.

15.27 \$PID — АДРЕС ПАКЕТА ДЛЯ РАСШИРЕННОГО ПРИЁМА

Ввод: **\$PIDd=hhhh [ddddddddS₁S₂]**, где

d номер ячейки от 0 до 7;

hhhh адрес, или маска адреса;

Не обязательные поля:

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

S₁ символ логической операции между РП пакета и маской dddddddd:

«&» операция «**И**»;

«|» операция «**ИЛИ**».

S₂ действие над пакетом в случае совпадения РП пакета с маской dddddddd:

«+» 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 приёме/не приёме пакета принимается на следующем уровне приёма пакетов (как при обычном приёме).

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

Если в модуле есть хотя бы один активный адрес базовой станции, при приёме 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 RIDx). При наличии активных ячеек \$PID в выводе команды \$DMP отображается подсказка " *LID ".

15.28 \$LID / \$XID — ВЫВОД / УДАЛЕНИЕ АДРЕСОВ ДЛЯ РЕТРАНЛЯЦИИ И РАСШИРЕННОГО ПРИЁМА

Ввод: **\$LID** - Вывод списка адресов для расширенной ретрансляции и приема.

\$XIDd, где **d** — номер ячейки от 0 до 7. Команда удаляет ячейку с адресом RID или PID.

При вводе номера ячейки больше 7, удаляется весь список сразу.

Пример:

Вывод списка

```
OK> $LID
# xID Adr Rst RptPath
-----
0 RID 0001 S1 4567 &+
1 RID 0002 D0 01 | -
2 ---
3 RID 0003 S0
4 PID 0004
5 ---
6 ---
7 PID 0007 05 &-
OK>
```

OK> \$XID0

OK> \$XID1

OK>

Удаление 2-х
начальных ячеек

Вывод списка после удаления

```
OK> $LID
# xID Adr Rst RptPath
-----
0 ---
1 ---
2 ---
3 RID 0003 S0
4 PID 0004
5 ---
6 ---
7 PID 0007 05 &-
OK>
```

15.29 \$QLID — ПРОВЕРКА СПИСКА RID/PID

Ввод: \$QLID tttt rrrr [dddddddd], где

tttt Адрес получателя пакета (TXID) в шестнадцатеричном виде;

rrrr Адрес отправителя пакета (MYID) в шестнадцатеричном виде;

Не обязательное поле:

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

Команда эмулирует процесс прохождения пакетов через список RID/PID. С помощью нее можно проверить корректность введенных значений RID/PID путем генерации полей адреса и ретрансляционного поля пакета. После исполнении команды выводятся признаки ретрансляции и приема данного пакета. В случае проверки списка на ретрансляцию, необходимо корректно задать собственный номер ретранслятора (команда \$RT).

```
OK> $lid
# xID Adr  Rst RptPath
-----
0 RID FFFF S1
1 RID 0004 S0 0123  |+
2 PID 0002
3 PID FFFF 012  &+
-----
OK> $qlid ffff 0001      Не принимать пакет от отправителя 0001 - не выполнено условие по маске в ячейке N3.
RX:0  RT:1              Ретранслировать согласно условиям в ячейке N0.
-----
OK> $qlid 0005 0004      Не принимать пакет от отправителя 0004 - не выполнено условие по маске в ячейке N3.
RX:0  RT:0              Не ретранслировать - не выполнено условие по маске в ячейке N1.
-----
OK> $qlid 0005 0004 12   Не принимать пакет от отправителя 0004 - не выполнено условие по маске в ячейке N3.
RX:0  RT:1              Ретранслировать согласно условиям в ячейке N1.
-----
OK> $qlid ffff 0005      Не принимать пакет от отправителя 0005 - не выполнено условие по маске в ячейке N3.
RX:0  RT:1              Ретранслировать согласно условиям в ячейке N0.
-----
OK> $qlid ffff 0005 012  Принимать пакет от отправителя 0005, согласно условиям в ячейке N3.
RX:1  RT:1              Ретранслировать согласно условиям в ячейке N0.
```

15.30 \$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](#)".

15.31 \$xFLDMP — ВЫВОД ЗНАЧЕНИЙ ФИЛЬТРОВ

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

15.32 \$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 необходимо 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 значения 0x5A

15.33 \$xFL- — ИСКЛЮЧЕНИЕ ДИАПАЗОНА ЗНАЧЕНИЙ ИЗ ФИЛЬТРА

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

Аргументы аналогичны команде \$xFL+.

15.34 \$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 сохранятся в ЭНОЗУ по команде \$S)

15.35 \$RPRF — ЧТЕНИЕ ПРОФИЛЯ УДАЛЁННОГО МОДУЛЯ

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

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

15.36 \$WPRF — ЗАПИСЬ ПРОФИЛЯ УДАЛЁННОГО МОДУЛЯ

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

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

15.37 \$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.

15.38 \$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](#)»).

15.39 \$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.

15.40 \$RCHK — ОПРОС УДАЛЁННОГО МОДУЛЯ

Ввод: **\$RCHK hhhh**, где **hhhh** — адрес удалённого модуля.

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

15.41 \$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>
```

15.42 \$TEST — ЗАПРОС ТЕСТОВЫХ ПАКЕТОВ

Ввод: **\$TEST hhhh,dc** где, **hhhh** — 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:

```

OK> $TEST 0002 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>

```

15.43 \$DRG — ВЫВОД ВСЕХ ТЕХНОЛОГИЧЕСКИХ ПАРАМЕТРОВВвод: **\$DRG****15.44 \$RG — ЗАПИСЬ ТЕХНОЛОГИЧЕСКИХ ПАРАМЕТРОВ МОДУЛЯ**

Ввод: **\$RGdd=hh** - запись в регистр **dd** число **hh** в шестнадцатеричном формате
\$RGdd=ddd - запись в регистр **dd** число **ddd** в десятичном формате
\$RGdd=bbbbbbbb - запись в регистр **dd** число **bbbbbbbb** в двоичном формате

Пример: **\$RG15=FE**
\$RG15=254
\$RG15=11111110

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

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 параметры описаны в разделе «[Технологические параметры. Регистры RG](#)».**15.45 \$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.**15.46 \$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.**15.47 \$R — СБРОС ЛОКАЛЬНОГО МОДУЛЯ**Ввод: **\$R****15.48 \$RST — СБРОС УДАЛЁННОГО МОДУЛЯ**Ввод: **\$RST hhhh**, где **hhhh** – 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 0002
press 's' to stop...
OK>
** tx
** reset start
OK>

```

15.49 \$S — ЗАПИСЬ ВНУТРЕННИХ ПЕРЕМЕННЫХ МОДУЛЯ В ЭНОЗУВвод: **\$S**

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

15.50 \$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.

15.51 \$LOG — ВЫВОД ЖУРНАЛА СОБЫТИЙ В ЭФИРЕ МОДУЛЯ

Ввод: \$LOG

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.

15.52 \$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'.

15.53 \$KEA/КЕН — УСТАНОВКА ПАРОЛЯ AES ШИФРОВАНИЯ

Ввод: \$KEA=c1c2c3....c16

- установка пароля в символьном формате, где:
c1....c16 - ASCII символы кроме пробела и запятой;

\$KEN=hh1hh2hh3....hh16

- установка пароля в 16-ричном формате, где:
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>
```

15.54 \$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>
```

15.55 \$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>
```


15.56 \$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:

ОК> \$CFL=123
*** CFL:123-----
ОК> \$CFL=
*** key deleted
ОК>

15.57 \$CFU— ВВОД ПАРОЛЯ НА КОМАНДНЫЙ РЕЖИМ

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

15.58 \$LOCK— РЕГИСТР БЛОКИРОВОК

Ввод: \$LOCK=bbbbbbbb

| Бит | Назначение | 1 | 0 |
|-----|--|----|-----|
| 7-3 | Резерв | | |
| 2 | bBootCmdLock Блокировка конфигурирования модуля через режим загрузчика. | Да | Нет |
| 1 | bEkranCfgLock Блокировка конфигурирования модема через символьный индикатор. В модуле Спектр 433 IO V2 не имеет значение. | Да | Нет |
| 0 | bRemoteCfgLock Блокировка удалённого конфигурирования по эфиру | Да | Нет |

15.59 \$DMPLOCK— ВЫВОД РЕГИСТРА БЛОКИРОВОК

Ввод: \$DMPLOCK

16 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.

| | |
|--|---|
| <pre>P2 TXID=FFFF COPY=1 ,MYID=0004 CH=92 PWR=0 COM=9600,8N1 AR=3 LBT=0 DAT=EOT PACT=005 PSL=000 PL=7 MN=3 DM=0 FEC=R7,3 RCONC=0 TCONC=0 IOADR=0A+IONDX=00 (0Ah) TFL=---- RT=- IORPT=11111111 RFL=---- RTFL=---- MaxTt=715 ms 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</pre> | <pre>P2 TXID=FFFF COPY=1 ,MYID=0004 CH=92 PWR=0 COM=9600,8N1 AR=3 LBT=0 DAT=EOT PACT=005 PSL=000 PL=7 MN=3 DM=0 FEC=R7,3 RCONC=0 TCONC=0 IOADR=0A+IONDX=00 (0Ah) TFL=---- RT=- IORPT=11111111 RFL=---- RTFL=---- MaxTt=715 ms IOMASTER IOF23=1 IOF15BC=0 IOPT=01 IOWA=1 IOSM0=1111 IOF15=----- IOSM1=1111 IOF02=-----210 IOSM2=0000 IOQPT=11111111 IOSM3=1111 IOSFT=010 IOSFV=---- IOALR=F0 IOSM4=1111 IOINL=---- IODBN=00000000h 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 IOMXO= 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</pre> |
|--|---|



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.

16.1 \$IOIEE – инициализация модуля IO в начальное состояние

Ввод: \$IOIEE

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

16.2 \$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).

16.3 \$IOROUT – сброс значений выходов

Ввод: \$IOROUT

Сброс выходов.

16.4 \$IOMASTER – ввод модуля IO в режим IO_MASTER

Ввод: \$IOMASTER

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

16.5 \$IOSLAVE – ВВОД МОДУЛЯ IO В РЕЖИМ IO_SLAVE

Ввод: \$IOSLAVE

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

16.6 \$IOTRANS – ВВОД МОДУЛЯ IO В РЕЖИМ IO_TRANS

Ввод: \$IOTRANS

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

16.7 \$IOADR – БАЗОВЫЙ АДРЕС МОДУЛЯ

Ввод: \$IOADR=hh

16.8 \$IONDX – ИНДЕКС МОДУЛЯ

Ввод: \$IONDX=hh

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

16.9 \$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.

16.10 \$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

16.11 \$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

16.12 \$IOF15 – СПИСОК IO_SLAVE МОДУЛЕЙ, ОПРАШИВАЕМЫХ ФУНКЦИЕЙ F15

Ввод: \$IOF15=bbbbbbbb, где b - 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

16.13 \$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).

16.14 \$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.

16.15 \$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.

16.16 \$IOWA – ВРЕМЯ ОЖИДАНИЯ ОТВЕТА ОТ IO_SLAVE УСТРОЙСТВ

Ввод: **\$IOWA=d**

Время, в течении которого IO_Master ожидает ответа (Wait Answer) от IO_Slave. Время выражается в числе пакетов максимального размера на текущей скорости. Значение 0 соответствует 1.

16.17 \$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.

16.18 \$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 загорятся 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.

16.19 \$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.

16.20 \$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 аварии 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.

16.21 \$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)

16.22 \$IODBN – СЧЕТЧИК ДРЕБЕЗГА ВХОДОВ

Ввод: \$IODBN=hhhhhhhh

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.

16.23 \$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.

16.24 \$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;

16.25 \$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').

16.26 \$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.

16.27 \$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 $100 \times 100мс = 10 с$.

16.28 \$IOINT – ПЕРЕДАЧА В ЭФИР СОСТОЯНИЙ ВХОДОВ ПРИ ИЗМЕНЕНИИ СОСТОЯНИЯ ОДНОГО ИЗ

Ввод: \$IOINT=bbbbbbbb (в текущей реализации состояние регистра игнорируется.)

16.29 \$IOMOD – РЕЖИМ РАБОТЫ МОДУЛЯ ВВОДА/ВЫВОДА

Ввод: \$IOMOD=bbbbbbbb (в текущей реализации состояние регистра игнорируется.)

17 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.

17.1 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. |

17.2 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".

17.3 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...0x1A - регистры основного профиля 0x00...0x1E - 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.

17.4 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...0x1A - регистры основного профиля 0x01...0x1E - 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 | |

17.5 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 " Чтение профиля ") |
| 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. |

17.6 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 |

18 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...2 | Резерв |
| 1 | RfRxEventDmp - печать событий при приеме пакета из эфира. Рекомендуется отключать, т.к. нагружает отладочный порт. |

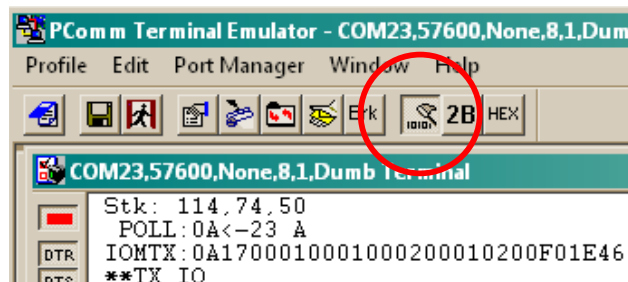
Для "быстрого" доступа к битам регистра \$RG15 доступны команды:

\$DD=TN - печать передаваемых данных как есть (NAT)
 \$DD=TH - печать передаваемых данных в шестнадцатеричном формате (HEX)
 \$DD=TD - передаваемые данные не печатаются (DIS)
 \$DD=RN - печать принимаемых данных как есть (NAT)
 \$DD=RH - печать принимаемых данных в шестнадцатеричном формате (HEX)
 \$DD=RD - принимаемые данные не печатаются (DIS)
 \$DD=E1 - печать событий разрешена
 \$DD=E0 - печать событий запрещена

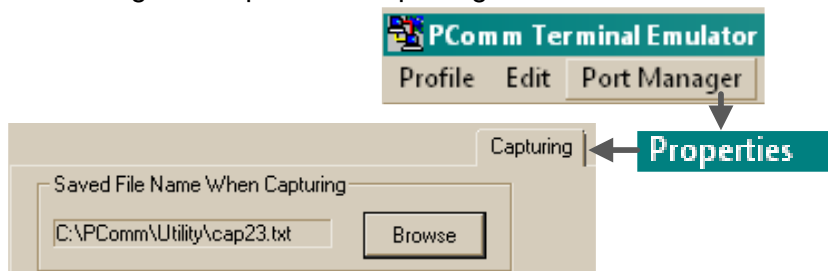
В случае возникновения ситуаций требующих анализа, через DEBUG порт можно записать лог работы модуля в реальном времени. Рассмотрим процесс с помощью терминальной программы Pcomm (<http://rateos.ru/files/PComm210.zip>).

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 отжатии - запись в файл прекращается.



Местоположение файла CAPx.txt можно узнать через меню PortManager->Properties->Capturing:



19 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".

При передаче данных передающий модем моргает светодиодами RX/TX на прием и передачу, однако у принимающего модема ничего не моргает.

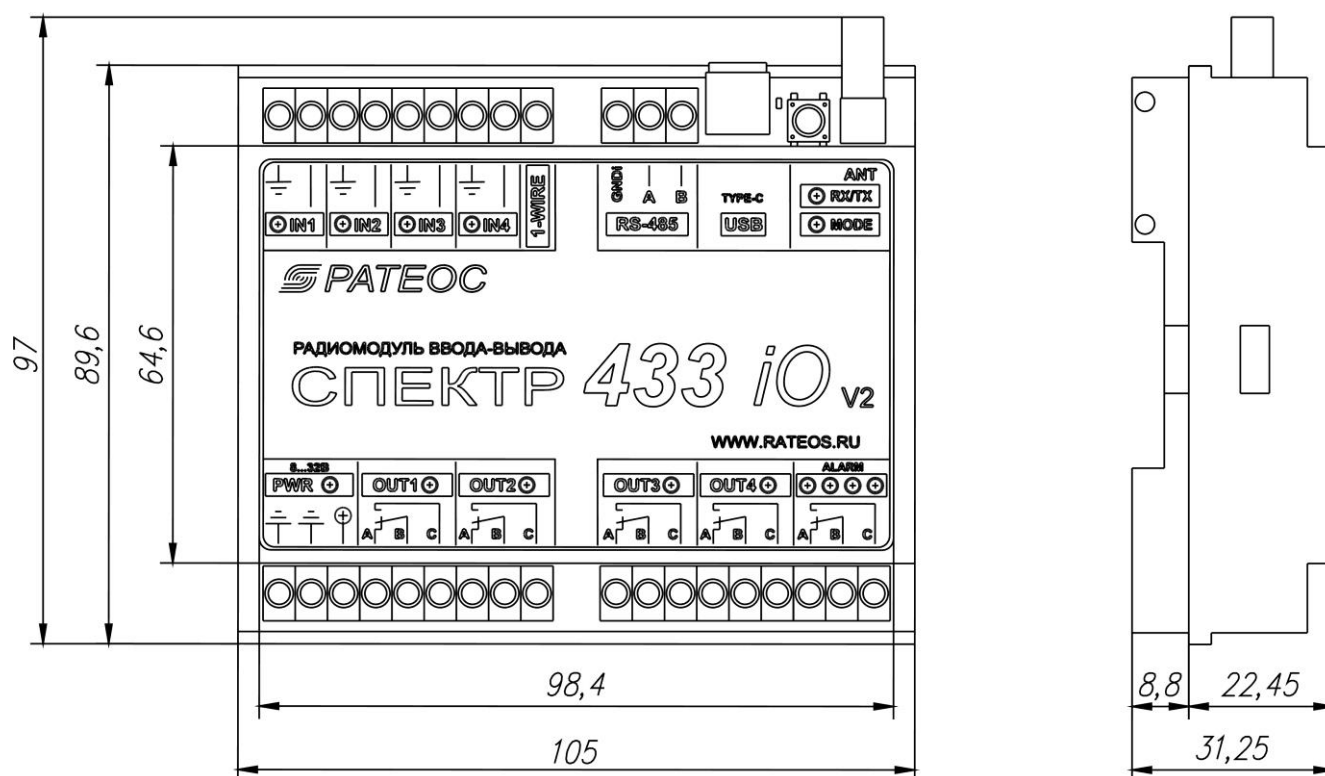
Скорее всего у принимающего модема перепутаны местами линии А и В т.к. подключенное оборудование не понимает запросы и, соответственно не отвечает.

МОДУЛЬ ВВОДА/ВЫВОДА

Вопрос

Ответ

20 ГАБАРИТНЫЕ РАЗМЕРЫ



21 ОСНОВНЫЕ ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

| | |
|---|---|
| Диапазон рабочих температур: | от -40 до +50 °C |
| Напряжение питания: | + (8...32) В |
| Потребляемая мощность (не более): | 0,5 Вт (средняя), 2,4 Вт (режим передачи) |
| Диапазон рабочих частот приемопередатчика: | 433,075... 434,775 МГц |
| Номинальная мощность передатчика: | 10 мВт |
| Чувствительность приемника: | -118 дБм, при скорости передачи 4800 бит/с |
| Шаг установки рабочей частоты: | 25 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) |