

Режим мониторинга

Некоторые рекомендации для системных интеграторов

Версия 1.1



www.parsec.ru



Оглавление

Введение	3
Подключение считывателей.....	3
Физический интерфейс	3
Назначение проводов	4
Принцип работы считывателя	4
Механизм антиколлизии	4
Один или два канала?.....	5
Цикл передачи – чтения	5
Внутренний стек меток.....	6
Буфер транзакций	8
Обработка транзакций считывателя.....	8
Дополнительная информация	9

Введение

Данный документ адресован системным интеграторам и программистам, разрабатывающим приложения мониторинга с использованием считывателей дальней идентификации PR-G07.

В документе описан принцип работы самого считывателя, механизмы, влияющие на качество идентификации, а также даны рекомендации по построению программного обеспечения прикладного уровня с использованием поставляемых со считывателем библиотек.

В качестве примера рассмотрен принцип построения приложения для динамического мониторинга объектов, находящихся в разных помещениях (при этом каждое помещение обслуживается отдельным считывателем).

Подключение считывателей

Напомним некоторые положения в части подключения считывателей к персональному компьютеру (ПК). При этом будем считать, что со стороны ПК поддержка протокола обмена осуществляется с помощью штатных динамических библиотек.

Физический интерфейс

Считыватели имеют гальванически развязанный интерфейс RS-485, позволяющий теоретически подключить на одну шину до 30 считывателей. На практике за счет невысокой скорости обмена последовательный опрос всех подключенных на одну линию считывателей приведет к задержкам поступления информации, поэтому не рекомендуется подключать на одну линию более 5...6 считывателей. Если, однако, не требуется режим реального времени или производится отслеживание очень медленно перемещающихся объектов, количество считывателей на линии может быть и больше.

Принцип подключения считывателей показан на рисунке 1.

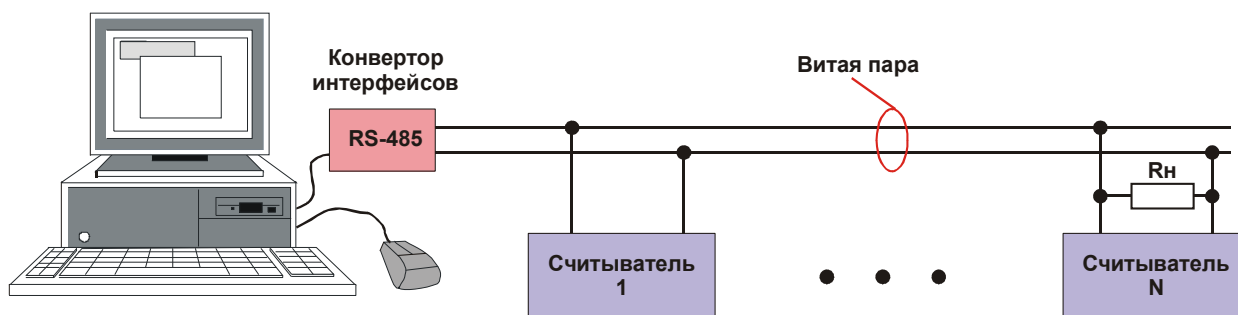


Рисунок 1. Подключение считывателей к ПК

Считыватели всегда выступают в роли ВЕДОМОГО. Это означает, что при обмене со считывателями некое устройство должно инициировать такой обмен путем отправки запросов (команд) и получения затем ответов. В качестве ведущего чаще всего выступает ПК.

Как правило, в ПК отсутствует интерфейс RS-485, и для его организации необходимо использовать внешний конвертор интерфейсов, среди которых самым удобным является конвертор USB <-> RS-485. Можно использовать и специальные встраиваемые в ПК платы расширения, имеющее от 1 до 4 и более портов RS-485, представляемых в системе как последовательные (COM) порты.

Поставляемые со считывателями библиотеки драйверов поддерживают работу как через COM – порты, так и через интерфейс NIP-A01, представляемый в системе как самостоятельное USB – устройство. Такой интерфейс можно заказать вместе со считывателем.

Обращаем внимание на то, что при использовании длинных линий связи (десятки метров и более) параллельно последнему считывателю витая пара должна быть нагружена на резистор (R_n на рисунке 1) величиной 120...150 Ом.

Реально максимальная длина шины при использовании витой пары 5-й категории может составлять 1000...1500 метров

Считыватели настроены при производстве на работу со следующими параметрами обмена:

- Скорость обмена 9600 бод
- Число битов данных 8
- Контроль четности нет
- Число стоповых битов 1

Назначение проводов

В таблице 1 приведено назначение выводов считывателя при работе его в режиме мониторинга. В этом режиме используется только один кабель (в отличие от варианта для систем управления доступом, для которых считыватель поставляется с двумя кабелями).

Таблица 1. Сигнальный кабель			
Цвет провода	Обозначение	Назначение	Примечание
Красный	+12V	Питание считывателя	Питание считывателя
Черный	GND	Общий провод	
Зеленый	W0-1	Выход W0 канала 1	В режиме мониторинга не используются, оставить не подключенными
Белый	W1-1	Выход W1 канала 1	
Голубой	W0-2	Выход W0 канала 2	
Оранжевый	W1-2	Выход W1 канала 2	
Желтый	+RS-485	Линия прямого сигнала	Интерфейс RS-485 с гальванической развязкой
Синий	-RS-485	Линия инверсного сигнала	
Серый	CMN	Общий провод RS-485	
Розовый	NC		Не используется
Фиолетовый	NC		Не используется
Коричневый	GND	Общий провод	Дополнительная «земля» (общий провод)

Принцип работы считывателя

Механизм антиколлизии

Считыватель в состоянии одновременно контролировать до нескольких десятков меток, находящихся в поле чтения. Стандартно в режиме мониторинга максимальное количество одновременно контролируемых меток – 64.

Система использует наиболее простой, не требующий двухстороннего обмена механизм антиколлизии, именуемый иногда в литературе free running (работа в свободном режиме). В этом режиме каждая метка передает свой код считывателю в произвольный момент времени, без синхронизации с остальными метками, находящимися в поле чтения (рисунок 2).

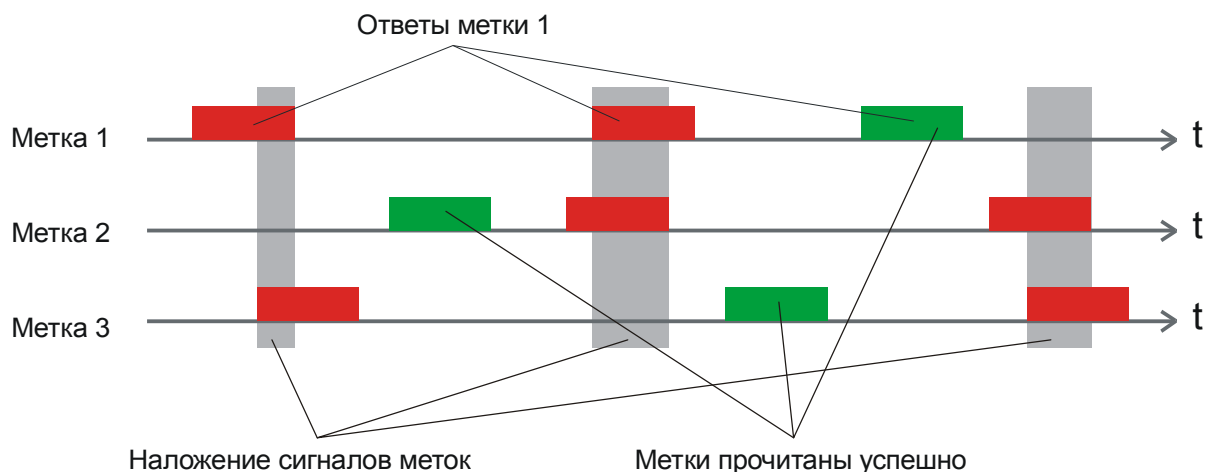


Рисунок 2. Механизм антиколлизии «free running».

За простоту реализации в данном случае приходится платить несколько худшим быстродействием, что проявляется только при большом количестве меток.

Поскольку метки на производстве программируются на один и тот же период обмена, для улучшения разрешения коллизий каждая метка по случайному закону от периода к периоду в небольших пределах меняет интервал между обменами.

Один или два канала?

Иногда требуется «выжать» максимальное разрешение считывателя по времени чтения заданного (более десятка) количества меток. В этом случае следует пользоваться следующими соображениями.

Цикл передачи – чтения

Для понимания дальнейшего изложения рассмотрим, как происходит цикл передачи кода метки считывателю и его обработка.

Метка передает свой код считывателю в течение определенного времени, определяемого скоростью передачи одного бита и длиной посылки, в состав которой включены служебные поля (преамбула, адрес канала и другие параметры), непосредственно четыре байта серийного номера метки, а также контрольная сумма передаваемой посылки для контроля корректности принятой считывателем информации. В реальном времени такой пакет обрабатывается в приемнике считывателя, и после того, как весь пакет принят, код метки извлекается из приемника с конечной скоростью. Соответственно, пока один пакет не извлечен из буфера приемника, последний заблокирован, то есть не может принимать пакет от другой метки. Если во время извлечения пакета из буфера приемника другая метка уже передает свой пакет, этот пакет будет считывателем пропущен.

Сказанное поясняется рисунком 3.

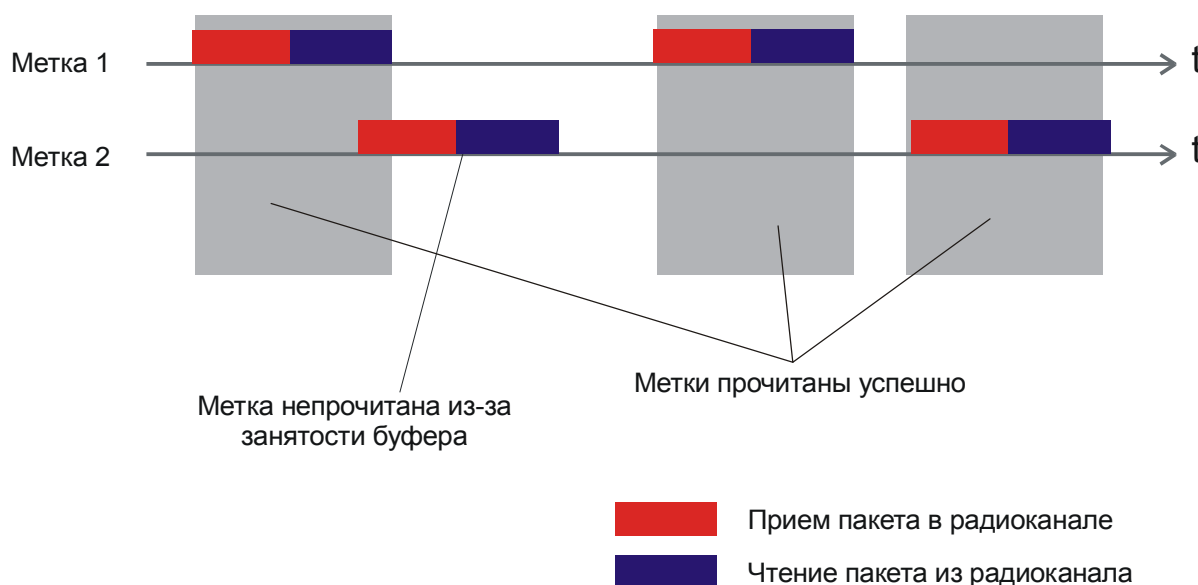


Рисунок 3. Потеря пакетов в момент чтения буфера приемника.

Поскольку считыватель имеет два радиоканала, обрабатываемых последовательно, с увеличением числа меток потери от наложений возрастают. Для получения максимально возможного быстродействия при мониторинге большого количества меток можно рекомендовать два решения:

1. Использовать только один канал считывателя, программно полностью отключив второй канал (заманчиво использовать каждый канал для мониторинга своего, не пересекающегося с другим каналом пространства, но тогда не следует требовать максимального быстродействия).
2. Использовать два канала для мониторинга одного и того же пространства – в этом случае буфера радиоканалов будут работать попеременно, также обеспечивая минимальную потерю информации.

Еще раз отметим, что при небольшом количестве меток и невысоких требованиях к быстродействию можно каждый канал использовать для мониторинга своего, непересекающегося с другим каналом, пространства.

Внутренний стек меток

Считыватель производит предварительную обработку меток, находящихся в поле чтения. Для этого в каждом канале считывателя существует стек меток. В стандартной версии для мониторинга размер этого стека составляет 64 метки, то есть именно столько меток считыватель может обрабатывать одновременно в каждом канале.

Принцип работы стека состоит в контроле состояний метки. Каждая метка обменивается со считывателем периодически. Для мониторинга на производстве программируется период обмена, равный примерно одной секунде. Каждая метка, впервые появившийся в поле считывателя, заносится в стек, при этом метке приписывается начальное значение времени памяти (устанавливается при конфигурировании считывателя отдельно для каждого канала). Далее это время декрементируется до тех пор, пока метка вновь не появится в поле считывателя, либо пока это время не станет равным нулю. Если метка появился вновь в поле считывателя до истечения времени памяти, то время вновь переводится до установленного значения. Если же время достигло нулевой отметки, то метка считается потерянной.

При первом появлении метки и при его исчезновении из поля чтения формируются соответствующие транзакции (об этом речь пойдет ниже).

Описанный механизм работы стека иллюстрируется рисунком 4.

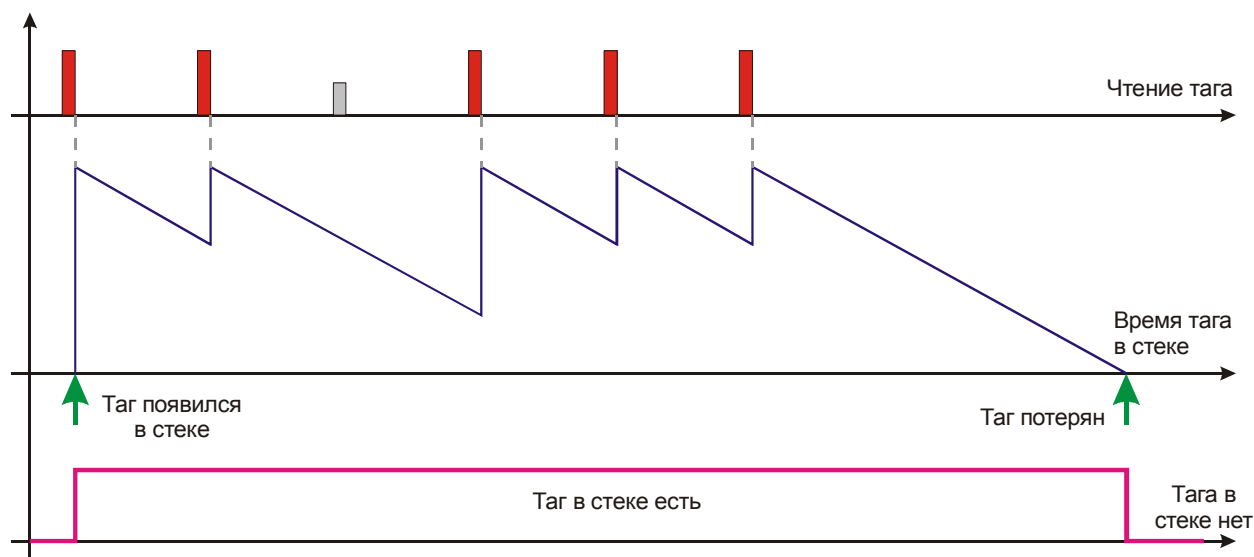


Рисунок 4. Работа стека меток одного канала.

На верхней оси показаны моменты чтения метки, причем серым более низким прямоугольником показано теоретическое время появления метки в поле считывателя, но по какой-то причине он не был прочитан.

На средней оси показан процесс обратного отсчета времени памяти метки между соседними чтениями.

На нижней оси показан результат, который считыватель передает наружу ведущему устройству.

Из приведенного выше описания следуют такие выводы:

1. Появление метки в поле считывателя фиксируется практически мгновенно.
2. Пропадание метки из поля считывателя фиксируется с задержкой, равной времени памяти метки в стеке.
3. С увеличением времени памяти метки уменьшаются «ложные срабатывания», возникающие за счет пропусков меток (коллизии, помехи), но при этом затягивается во времени принятие решение о пропадании метки из поля чтения.

Таким образом, в силу вероятностной природы самого механизма работы системы теоретически невозможно избавиться от фактов периодического пропадания меток из поля чтения, а на практике необходимо искать компромисс между «помехозащищенностью» (ложными срабатываниями системы при случайном пропадании метки) и быстродействием.

Рекомендуется время памяти метки ставить не меньше, чем 5 . . . 10 периодов излучения метки. Для стандартного режима мониторинга это будет соответствовать установке времени памяти от 5 до 10 секунд.

Состояние стека можно контролировать в режиме on-line, используя команды получения числа меток в стеке канала считывателя, а также команды последовательного получения номеров всех меток, находящихся в стеке.

Дополнительный способ повышения устойчивости детектирования меток будет рассмотрен в одном из следующих разделов, а сейчас поговорим о буфере транзакций считывателя.

Буфер транзакций

Перепады самой нижней кривой рисунка 4, то есть моменты фиксации новой метки в поле считывателя и моменты его пропадания фиксируются считывателем как события (мы именуем их транзакциями), и с меткой даты – времени заносятся в энергонезависимую память считывателя – буфер транзакций. Буфер выполнен кольцевым, то есть вновь поступающие транзакции заносятся в «голову», а считываемые хостом транзакции извлекаются из «хвоста» этой кольцевой очереди. Если буфер переполняется, то первыми начинают затираться самые старые по времени транзакции.

Принцип работы буфера транзакций иллюстрируется рисунком 5.

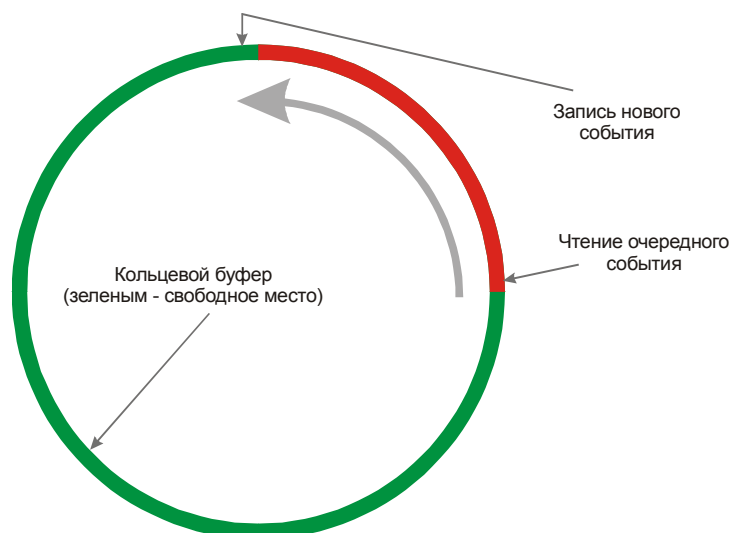


Рисунок 5. Работа буфера транзакций.

В стандартном исполнении буфер транзакций имеет емкость 1023 события.

Наличие буфера транзакций позволяет, во-первых, получить информацию о событиях, которые считыватель зафиксировал при автономном режиме работы, а, во-вторых, организовать по транзакциям дополнительную обработку находящихся в поле считывателя меток на уровне хоста, повысив эквивалентную «помехозащищенность» системы мониторинга.

Обработка транзакций считывателя

Система мониторинга предполагает наличие хоста, то есть устройства, которое получает информацию от считывателя, производит ее обработку (при необходимости) и принимает решения.

Ниже мы рассмотрим один из методов обработки транзакций считывателя, позволяющий уменьшить число «ложных» транзакций, то есть ситуаций, когда метка реально не покидала зону чтения, но в силу случайного стечения внешних обстоятельств на короткое время все-таки пропадала из зоны чтения. Этот метод позволяет получить лучшие результаты чем те, которые могут быть получены простым увеличением времени памяти метки в стеке считывателя.

Метод особенно эффективно работает в случаях, когда метка находится в граничной зоне (на краю зоны чтения), и считывается считывателем достаточно редко.

В основе метода лежит классический принцип двухстороннего интегратора (стек считывателя в этих терминах аналогичен одностороннему интегратору).

А поясняется предлагаемый метод рисунком 6.

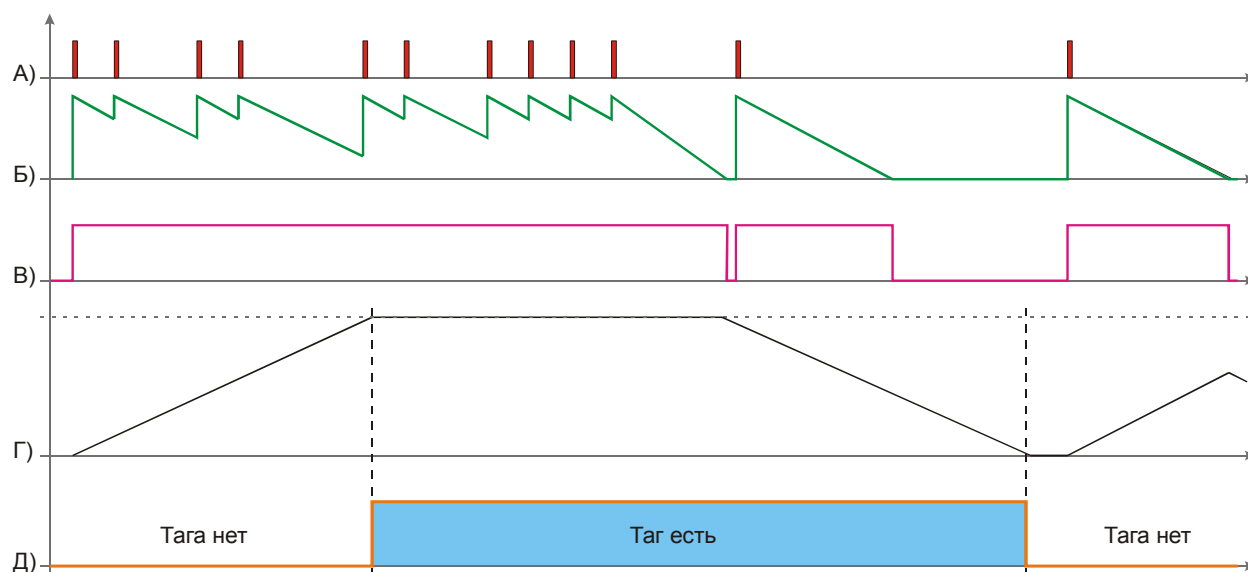


Рисунок 6. Использование метода двухстороннего интегратора.

На рисунке 6 обозначены:

- А) – чтение метки считывателем
- Б) – обработка метки в стеке считывателя
- В) – транзакции считывателя, управляющие направлением интегрирования
- Г) – двухсторонний интегратор с порогом, равным «0» и «max»
- Д) – решение на выходе порогового устройства после интегратора

Как видно из рисунка, вместо последовательности транзакций с выхода считывателя вида «метка есть – метки нет – метка есть – метки нет – метка есть – метки нет» на том же временном отрезке появление и пропадание метки при принятии решения на выходе интегратора происходит единожды.

Причем главное преимущество двухстороннего интегрирования в том, что полностью фильтруются как кратковременные пропадания метки, так и его кратковременные появления.

Естественно, что любая фильтрация вносит задержку в принятие решения – такова физика самого процесса фильтрации, но зато отпадает необходимость «ручной» постобработки информации, получаемой со считывателя.

При реализации предложенного алгоритма время памяти в радиоканале считывателя необходимо ставить меньше, чем указывалось ранее для штатного механизма обработки меток в стеке. Мы бы рекомендовали ставить его в пределах 2...3 периода излучения метки (то есть для режима мониторинга – 2...3 секунды), в то время как время интегрирования от нижнего до верхнего порога устанавливать равным 10 или более периодам излучения метки.

В любом случае, чем большее количество меток мы хотим одновременно контролировать в одном канале считывателя, тем большее значение времени рекомендуется устанавливать в любом из алгоритмов для уменьшения «ложных» событий в системе.

Дополнительная информация

Всю дополнительную информацию можно получить по адресу:

support@parsec.ru