PRG-USB-Protocol

Оглавление

PRG-USB-Protocol 1
Оглавление 1
Общее описание протокола 1
Физический уровень 1
Физическая среда RS-232 1
Физическая среда USB 1
Транспортный уровень 2
Кадр запроса от компьютера считывателю 2
Кадр ответа считывателя компьютеру 2
Кадры ответа ACK/NACK 2
Байтстаффинг 3
FCS 3
Многобайтовые переменные 3
О повторном запросе 3
Общие рекомендации по протоколу 4
Рекомендации по разбору кадра 4
Таймауты 4
Назначение идентификатора кадра 4
Смена скорости интерфейса 4
Пример исходника для приема данных 4
Примеры корректных кадров 5
Инструменты работы с протоколом 6
Описание команд 7
Коды команд 7
Общие команды 7
0x00 – "Заголовок устройства" 7
Параметры считывателя 8
0x01 – "Запись параметров" 8
0x02 – "Чтение параметров" 8
Команды карт формата EM-Marin 9
0x10 – "Чтение карты EM-Marin" 9
Команды карт формата HID 9
0x14 – "Чтение карты HID" 9
Команды карт формата Motorola 9
0x18 – "Чтение карты Motorola" 9

Общее описание протокола

Физический уровень

Имеются версии считывателей с физическими средами передачи RS-232, USB 1.1.
Работа ведется с виртуальными или физическими COM-портами.

Физическая среда RS-232

Для связи между компьютером и считывателем используется последовательный асинхронный интерфейс со следующими настройками: 8 data bits, 1 stop bit, no parity, no flow control. Скорость при поставке 9600, может быть изменена в процессе работы.

Электрические параметры интерфейса: см. соответствующие стандарты. Используются линии RxD, TxD, GND.

Протокол дуплексный, наличие эха не допускается.

Физическая среда USB

Для связи между компьютером и считывателем используется интерфейс USB. Со стороны компьютера интерфейс представляется стандартным виртуальным COM-портом. Необходимый драйвер под ОС Windows идет в комплекте (SDK). Никаких изменений ПО компьютера не требуется.

Транспортный уровень

Связь осуществляется кадрами в режиме «запрос-ответ». Инициатором обмена может быть только мастер системы (компьютер). Считыватель обязан подтвердить получение ответом.

Кадр запроса от компьютера считывателю

Поле	Длина, байт	Данные
Стартовое условие	1	0xFD
Идентификатор кадра	1	
Код команды	1	См. Таблицу команд
Данные	XX	Зависит от команды
FCS кадра	2	Начиная с поля "Идентификатор кадра" и заканчивая полем "Данные"
Стоповое условие	1	0xFE

После получения запроса считыватель должен проверить правильность приема путем FCS. Подтверждением правильного приема является передача запрашиваемых данных либо, для команд, не запрашивающих данные, кадр ответа ACK (acknowledged). При неправильном приеме для всех команд передается кадр NACK (not acknowledged).

Кадр ответа считывателя компьютеру

Поле	Длина, байт	Данные
Стартовое условие	1	0xFD
Идентификатор кадра	1	Повторение идентификатора кадра
Код команды	1	Повторение кода команды компьютера
Данные	XX	Зависит от команды
FCS кадра	2	Начиная с поля "Идентификатор кадра" и заканчивая полем "Данные"
Стоповое условие	1	0xFE

Кадры ответа ACK/NACK

Поле	Длина, байт	Данные
Стартовое условие	1	0xFD
Идентификатор кадра	1	Повторение идентификатора кадра
Признак кадра ACK/NACK	1	0x2A
Данные	1	0x55 – ACK 0x01 – NACK 1 (несовпадение контрольной суммы) 0x02 – NACK 2 (неверная команда) 0x03 – NACK 3 (неверные данные) 0x05 – NACK 5 (аппаратная авария при исполнении) 0x06 – NACK 6 (валидная карточка не обнаружена)
FCS кадра	2	Начиная с поля "Идентификатор кадра" и заканчивая полем "Данные"
Стоповое условие	1	0xFE

Для формирования ответа считывателем должны быть приняты как минимум стартовый и стоповый байты, идентификатор кадра, код команды и FCS. Также должны отсутствовать ошибки байтстаффинга в пределах кадра. Иначе никакие ответы не формируются и мастер должен повторить запрос по окончании таймаута, выставив соответствующий признак повторного запроса.

Считыватель вправе игнорировать новый запрос если он не успел полностью обработать предыдущий. Гарантируется что считыватель готов к приему новой команды к моменту окончания передачи первого кадра ответа на последний запрос.

Считыватель вправе игнорировать запрос если количество байт между стартовым и стоповым условиями больше чем оговорено самой большой командой.

Ответ NACK 1 посылается в случае ошибки FCS. Единственная цель такого ответа - ускорить повтор запроса до окончания таймаута.

Ответ NACK 2 посылается на все неверные коды команд. Цель ответа – уведомить компьютер об отсутствии команды в данной версии считывателя.

Ответ NACK 3 посылается на неприемлемый для данной команды набор данных. Это может быть

неверное количество байт данных или их неприемлемое значение.

Ответ NACK 5 посылается некоторыми командами при выявлении аппаратных сбоев устройства. Это может быть вызвано как неисправностью (случайным сбоем) считывателя, так и ошибками при программировании считывателя.

Ответ NACK 6 посылается командами работы с картой если карта в поле не обнаружена, дала нераспознанный ответ или отказала в выполнении команды.

Конкретные причины возможных ситуаций расшифровываются в описании команд.

От компьютера к считывателю кадры ACK / NACK не посылаются: если компьютер не получил валидный ответ, запрос передается повторно согласно правилам повторного запроса.

Байтстаффинг

Если между флагами начала и конца команды встречаются специальные символы (0xFD, 0xFE, 0xFF), то они кодируются в соответствии с таблицей байтстаффинга:

Специальный символ	Кодирование
0xFD	0xFF 0x02
0xFE	0xFF 0x01
0xFF	0xFF 0x00

FCS

FCS есть средство контроля целостности кадра. FCS считается над всеми полями кадра кроме стартового и стопового байт и поля самого FCS.

FCS в данном протоколе реализована согласно стандартам CCITT X.25 он же ISO/IEC 3309 он же RFC1331 (PPP). Контрольные последовательности, быстрые табличные реализации и другие фишки FCS могут быть найдены в соответствующих стандартах. Здесь приведен предельно упрощенный алгоритм реализации применительно к одному байту.

```
Unsigned int    FCS;

Void X_FCS (unsigned char _D)
{
unsigned char _i;
unsigned int _w;

_w = (_D ^ FCS) & 0xFF;
_i = 8;
do {
        if (_w & 1) {
                _w >>= 1;
                _w ^= 0x8408;
        } else {
                _w >>= 1;
        }
} while (--_i);
FCS = _w ^ (FCS >> 8);
}
```

Для кадра имеем начальное значение FCS = 0xFFFF, байты считаются начиная с первого. По окончании расчета FCS инвертируется.

Вычисление FCS при передаче производится ДО проведения байтстаффинга, а при приеме сначала производится байтстаффинг, а потом производится проверка FCS.

Многобайтовые переменные

Целые многобайтовые значения передаются младшим байтом вперед. Исключение - код карты, он передается старшим вперед.

Числа с плавающей запятой имеют формат согласно IEEE754 и передаются экспонентой вперед.

Строки передаются первым символом вперед. Конец строки либо отмечается нулевым байтом либо по достижении указанного размера. Если строка короче отведенного для нее поля, оставшиеся байты игнорируются (заполняются произвольным значением).

О повторном запросе

Если мастер не смог получить от считывателя ответ на запрос, он повторяет запрос. Однако мастер не

знает доподлинно, выполнил ли считыватель команду в первый раз или нет. Многие команды недопустимо или нежелательно выполнять повторно.

Чтобы избежать повторного выполнения команды, реализован простой механизм с использованием идентификатора кадра.

Мастер, не получив ответ, передает снова тот же запрос с тем же идентификатором кадра.

Считыватель, приняв кадр, анализирует идентификатор кадра и если последняя выполненная команда имела те же код команды и идентификатор кадра, игнорирует выполнение самой команды, а только повторяет передачу последнего ответа. При этом данные в новом кадре игнорируются. Если же код команды или идентификатор кадра не совпали с последними выполненными, команда исполняется полностью.

Общие рекомендации по протоколу

Рекомендации по разбору кадра

При разборе кадра следует придерживаться следующих правил.

Стартовое условие обладает наивысшим приоритетом при разборе потока.

Стоповое условие обладает вторым уровнем приоритета. Принятие стопового условия есть сигнал к началу разбора принятого кадра. Весь поток до следующего стартового условия отбрасывается.

Байтстаффинг обладает третьим уровнем приоритета. Ошибка байтстаффинга в процессе разбора потока есть основание для отбрасывания всего потока до следующего стартового условия.

Разбор кадра начинается с проверки минимально допустимой длины кадра (длина кадра без поля данных) и его FCS (ищется как смещение от стопового условия). Все остальные поля ищутся как смещения от стартового условия.

Таймауты

Задержки на ответ считывателя по протоколу определяются большим количеством факторов. Учесть их можно только очень грубо

При передаче данных считывателем максимальная задержка между байтами (окончанием одного и началом другого) не нормируется.

При передаче кадра компьютером не ставится условий по максимальной задержке между байтами одного кадра.

Назначение идентификатора кадра

В медленных компьютерах с громоздкими ОС при плотном трафике наблюдается эффект рассинхронизации запросов и ответов. Идентификатор кадра позволяет решить эту проблему. Компьютер назначает каждому новому запросу новый идентификатор и при приеме точно знает, какому запросу соответствовал ответ. Рекомендуется инкрементировать идентификатор каждый раз перед посылкой нового запроса.

Кроме того, идентификатор кадра используется в алгоритме опознавания повторных запросов и множественных ответов на одну команду.

Смена скорости интерфейса

Смена скорости производится немедленно после ответа ACK на команду смены скорости.

Потенциально возможна ситуация рассогласования рабочей скорости на компьютере и считывателе. Абстрагируясь от причин такой ситуации, предложим универсальное решение этой проблемы. В случае если считыватель не отвечает на запросы мастера, рекомендуется выполнить опрос считывателя какой-либо нейтральной командой на всех разрешенных скоростях и автоматически переустановить скорость на рабочую.

Пример исходника для приема данных

Другие исходные тексты программ можно запросить в техподдержке.

```
Void RX_Int(char _UDR)
{
static char  _Q;
static u16_t  _CRC;
static char  _State = 0;
if (_UDR == 0xFD) {
      _State = 1;
      _Q = 0;
      _CRC = 0xFFFF;
```

```
} else {
    switch (_State) {
    default:
        _State = 0;
        break;
    case 1:
            switch (_UDR) {
            case 0xFF:
                _State = 2;
                break;
            case 0xFE:
                if (_Q >= 4) {
                    if (_CRC == 0xf0b8) {
                            // Кадр с данными успешно принят…
                    }
                }
                _State = 0;
                break;
            default:
                goto _Rcv_GetChar;
            }
            break;
        case 2:
            if (_UDR <= 0x02) {
                _UDR = 0xFF - _UDR;
                _State = 1;
            } else {
                _State = 0;
            }
        _Rcv_GetChar:
            if (_Q >= UART_RX_BufLength) {
                _State = 0;
            } else {
                u16_t  _w;
                // Сохранение принятого байта в буфере
                UART_RX_Buf[_Q++] = _UDR;
                // Расчет CRC принятого байта
                _w = (_UDR ^ _CRC) & 0xFF;
                _UDR = 8;
                do {
                    if (_w & 1) {
                        _w >>= 1;
                        _w ^= 0x8408;
                    } else {
                        _w >>= 1;
                    }
                } while (--_UDR);
                _CRC = _w ^ (_CRC >> 8);
            }
            break;
        }
    }
}
```

Примеры корректных кадров

Все значения в примерах - шестнадцатиричные. Точки показаны только для удобства чтения.
Запрос заголовка устройства:
FD.00.00.47.0F.FE
Ответ ACK:
FD.00.2A.55.A7.1D.FE
Ответ NACK 2:

FD.00.2A.02.9D.3B.FE

Инструменты работы с протоколом

Имеется два основных инструмента для работы с данным протоколом. Оба ориентированы на демонстрацию и тестирование функциональности считывателя.

Первый инструмент – низкоуровневая терминальная программа W_Term.exe, заточенная на работу с данным протоколом. Ее назначение – работа с на уровне транспортного протокола. Терминалка поддерживает работу с COM портами и с IP-стеком; упаковывает заданную пользователем команду и данные в кадр протокола; распознает и распаковывает кадр ответа; считает таймауты. Ее всегда можно скачать с нашего сайта или получить по адресам техподдержки.

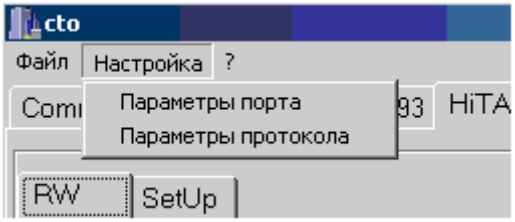
Второй инструмент – высокоуровневая программа демонстрации функциональности собственно считывателей карт CashTest.exe. Позволяет эффективно работать с конкретными командами считывателей и с картами соответствующих стандартов.

Оба инструмента основаны на PROX232.dll, реализующей низкий уровень данного протокола. Описание интерфейса dll доступно в отдельном документе.

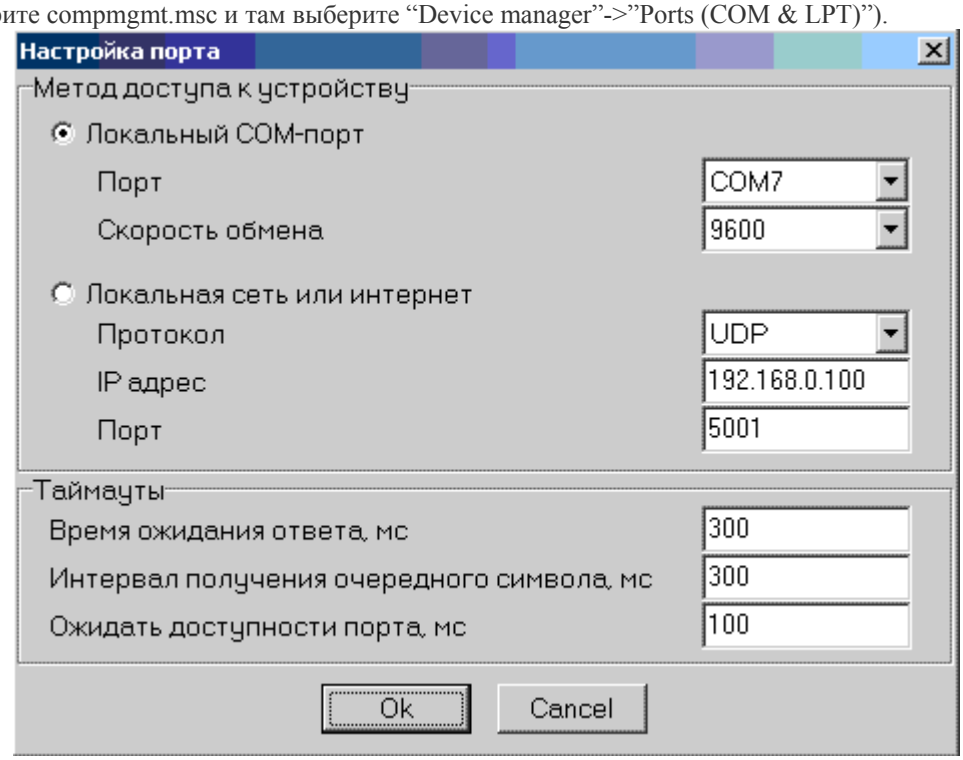
Далее кратко изложен порядок работы с инструментами.

Оба инструмента не имеют инсталлятора и запускаются «как есть». В одной папке должны находиться для CashTest: CashTest.exe, PROX232.dll, CashTest.ini. Для терминалки: W_Term.exe, PROX232.dll. При этом для CashTest желательна возможность перезаписи CashTest.ini, иначе невозможно будет сменить настройки программы. Терминалка хранит свои настройки в реестре (путь).

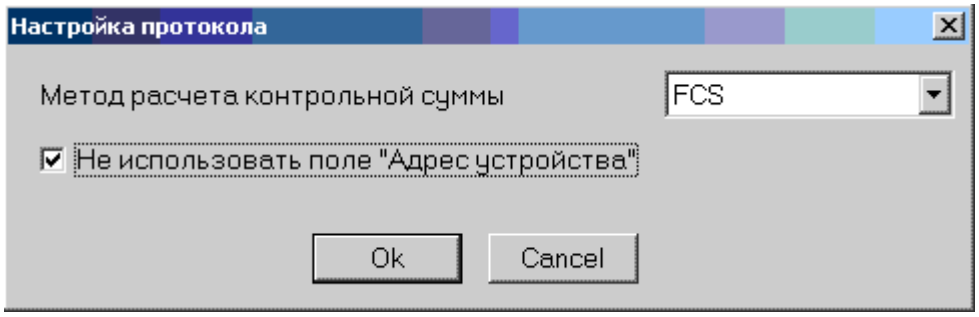
При первом запуске необходимо установить параметры порта считывателя (в первую очередь номер COM порта или параметры IP соединения для Ethernet считывателей). Как это сделать – поясняют картинки ниже.



К настройке порта следует сказать, что USB считыватели видны в системе как виртуальный COM-порт (после установки драйверов, естественно). Номер COM порта можно узнать в свойствах системы (например, наберите compmgmt.msc и там выберите “Device manager”->”Ports (COM & LPT)”).



Параметры настройки протокола следует просто проверить (или выставить заново как на картинке если был утерян файл конфигурации):



Описание команд

Коды команд

Код команды имеет длину 1 байт. Коды команд считывателя сведены в таблицу:
Таблица команд считывателя.

Команда	Код Команды	Возможные ответы
Общие команды		
Заголовок устройства	0x00	Данные / NACK
Запись параметров	0x01	ACK / NACK
Чтение параметров	0x02	Данные / NACK
Команды карт формата EM-Marin		
Чтение карты EM-Marin	0x10	Данные / NACK
Команды карт формата HID		
Чтение карты HID	0x14	Данные / NACK
Команды карт формата Motorola		
Чтение карты Motorola	0x18	Данные / NACK

Общие команды

0x00 – "Заголовок устройства"

Команда считывает из устройства его тип, номер версии, номер устройства.
Код команды: 0x00. Команда поля данных не имеет.
Варианты ответа: кадр ответа NACK или ответ со структурой данных, описывающей устройство:

Поля данных	Длина, Байт	Данные
Тип устройства	20	Текстовая строка с названием устройства.
Идентификатор устройства	4	unsigned long
Номер версии устройства	4	unsigned long
Номер версии протокола	4	unsigned long
Серийный номер изделия	4	unsigned long
Дополнительные флаги	4	unsigned long

Назначение полей:
«Тип устройства» - для справки.
«Идентификатор устройства» - идентификатор типа считывателя.
«Номер версии устройства» - для справки в разделе About.
«Номер версии протокола» - для справки в разделе About. Нужно предусмотреть возможность сверки со списком поддерживаемых версий.
«Серийный номер изделия» - для справки в разделе About. Некоторые изделия не поддерживают энергонезависимое хранение этого параметра, при снятии питания сбрасывают его в 0.
«Дополнительные флаги» - инструмент индикации поддерживаемых карт. Битовый массив, назначение бит:

- Бит 0 – имеется поддержка чтения карт формата EM-Marin EM4100.
- Бит 1 – Не используется.
- Бит 2 – имеется поддержка чтения карт HID ProxCard.
- Бит 3 – Не используется.
- Бит 4 – имеется поддержка чтения карт Motorola (Indala) ASP.
- Бит 5 – Не используется.
- Биты 6, 7 - резерв.
- Биты 8…31 - резерв.

Параметры считывателя

В считывателе есть ряд параметров. Каждый параметр снабжен кодом и доступен по чтению и/или по записи с помощью команд чтения и записи параметров. Код параметра имеет длину 1 байт.

При записи любого параметра реальное его значение в устройстве обновляется немедленно после ответа ACK.

Несуществующие коды параметров или ненормированный способ доступа к ним вызывает ответ NACK 3.

Таблица параметров считывателя

Код	Описание	Длина, байт	Возможные значения	Доступ
0x02	Скорость интерфейса	1	Перечислимый тип: 3 - скорость 9600 4 - скорость 19200 5 - скорость 38400 6 - скорость 57600 7 - скорость 115200 8 - скорость 230400** 9 - скорость 460800** 10 - скорость 921600**	Rd/Wr*

* - некоторые изделия могут иметь жестко заданное при производстве значение параметра либо запись разрешается только после авторизации.

** - скорости выше 115200 поддерживаются не всеми считывателями (см. историю реализаций). Если эти скорости не поддерживаются, считыватель ответит NACK3 на попытку их установить. Установка скорости выше 115200 не сохраняется энергонезависимо и сбрасывается на 9600 при включении питания.

0x01 – "Запись параметров"

Команда записывает в устройство различные параметры его работы.

Код команды: 0x01.

Тело команды:

Тело команды	Длина, Байт	Данные
Код параметра	1	См. таблицу "Параметры считывателя".
Данные параметра	1…20	Длина зависит от кода параметра

Коды параметров:

Варианты ответа: Кадр ответа ACK / NACK.

0x02 – "Чтение параметров"

Команда читает из устройства значения параметров его работы. Команда дуальна по отношению к команде 0x01 – "Запись параметров".

Код команды: 0x02.

Тело команды содержит только код параметра.

Поле	Длина, байт	Данные
Код параметров	1	См. таблицу "Параметры считывателя".

Коды параметров совпадают по назначению и длине возвращаемых данных с кодами команды 0x01.

Варианты ответа: Кадр ответа NACK или код параметра и значение указанного параметра:

Поле данных ответа	Длина, Байт	Данные
Данные параметра	1…20	Зависит от кода параметра

Команды карт формата EM-Marin

0x10 – "Чтение карты EM-Marin"

Команда считывает код карточки формата EM-Marin.
Код команды: 0x10.
Команда данных не содержит.
Варианты ответа: Кадр ответа NACK или прочитанные данные:

Поле данных ответа	Длина, Байт	Данные
Код карты	5	Байтовый массив

Команды карт формата HID

0x14 – "Чтение карты HID"

Команда считывает код карточки формата HID.
Команда данных не содержит.
Варианты ответа: Кадр ответа NACK или прочитанные данные:

Поле данных ответа	Длина, Байт	Данные
Тип Wiegand	1	
Код	5	байтовый массив

Возможные значения поля «Тип Wiegand»:

- 26 – Wiegand 26
- 34 – Wiegand 34
- 37 – Wiegand 37
- 0xFF – unknown format or format error

Команды карт формата Motorola

0x18 – "Чтение карты Motorola"

Команда считывает код карточки формата Motorola.
Команда данных не содержит.
Варианты ответа: Кадр ответа NACK или прочитанные данные:

Поле данных ответа	Длина, Байт	Данные
Код	5	байтовый массив