# **Стек протоколов DLMS:**

## **Физический уровень:**

Интерфейс RS485, который предназначен для обмена данными между счетчиком с внешними устройствами. Максимальная скорость интерфейса связи RS485 – 19200 бод.

## **Уровень HDLC**

*High-Level Data Link Control (HDLC) — бит-ориентированный протокол канального уровня сетевой модели OSI, разработанный ISO.*

*Текущим стандартом для HDLC является ISO 13239.*

*HDLC может быть использован в соединениях с множественным доступом, но в настоящее время в основном используется в соединениях точка-точка с использованием асинхронного сбалансированного режима (ABM*

**Уровень HDLC** (High-level data link control) – соответствует 2-4 (канальный, сетевой, транспортный) уровню OSI.

**Уровень HDLC является ориентированным на соединение**. Это означает, что на HDLC уровне должна сначала быть установлена логическую связь друг с другом путем обмена и обсуждения некоторых параметров соединения. Затем, они могут обмениваться данными, сколько потребуется и, наконец, закрыть соединение. Элементы данных, которыми обмениваются на HDLC уровне называются **HDLC кадрами**.

Весь двунаправленный обмен данными на HDLC уровне является достаточно сложным: HDLC уровень способен сегментировать длинные данные в информационные кадры (I-кадр) и собирать информационные кадры в длинные данные, он проверяет последовательность кадров, проверяет целостность кадров используя циклический избыточный код (CRC), он контролирует различные тайм-ауты. Мы не будем вдаваться в подробности обмена на HDLC уровне. Вместо этого, в нашей целевой программе, мы будем использовать нашу библиотеку ezhdlc, которая реализует весь HDLC обмен и просто выставляет три функции:

Существует несколько параметров, которые устанавливаются перед вызовом HDLC маршрутов, наиболее важными являются HDLC адреса (также называемых MAC адресами). Клиентский MAC-адрес – это однобайтовое значение, он идентифицирует клиента. Мы будем использовать 16 (десятичное), что означает публичный клиент (**public client**).

Серверный (или счетчик) MAC-адрес состоит из двух частей, верхняя часть (**upper** part) – это логический адрес устройства (**logical device address**), а нижняя часть (**lower** part) – это физический адрес устройства (**physical device address**). В некоторых случаях, нижняя часть может быть опущена. Например, в режиме "точка-точка" в конфигурации с одним счетчиком, мы не заботимся о физическом адресе счетчика. Заметим, однако, что эта особенность реализуется не всеми производителями.

Подведем итог: на HDLC уровне, адрес клиента всегда один байта, адрес сервера состоит из двух частей и возможны три варианта:

* Однобайтовая адресация. Есть только верхний адрес. Это однобайтовое значение.
* Двухбайтовая адресации. Есть верхний адрес – 1 байт и нижний адрес – 1 байт.
* Четырехбайтовая адресация. Есть верхний адрес – 2 байта и нижний адрес – два байта.

Обратите внимание, что не все производители поддерживают три варианта.

### **Управление потоком**

**Управление потоком** в HDLC осуществляется с помощью передающих и принимающих окон. Окно устанавливается на каждом конце канала связи, чтобы обеспечить резервирование ресурсов обеих станций. Этими ресурсами могут быть ресурсы вычислителя или пространство буфера. В большинстве случаев окно обеспечивает и буферное пространство, и правила нумерации (сообщений). Окно устанавливается во время инициирования сеанса связи между станциями. Если станция А и станция В должны обменяться данными, А резервирует окно для В, а В резервирует окно для А. Использование окон необходимо для полнодуплексных протоколов, потому что они подразумевают непрерывный поток кадров в принимающий узел без периодических подтверждений с остановкой и ожиданием.

**Переменные состояния станции V(S) и V(R).** Окна в принимающем и передающем узлах управляются переменными состояния, которые представляют по сути состояние счетчика. Передающий узел поддерживает переменную состояния посылки V(S). Это порядковый номер следующего по очереди I-кадра, который должен быть передан. Принимающий узел поддерживает переменную состояния приема V(R), которая содержит номер, который, как ожидается, является порядковым номером следующего I-кадра. V(S) увеличивается на 1 при передаче каждого кадра и помещается в поле порядкового номера посылки кадра. Получив кадр, принимающий узел производит проверку наличия ошибок передачи и сравнивает порядковый номер со своим V(R). Если кадр может быть принят, узел увеличивает V(R) на 1, помещает его в поле порядкового номера приема кадра подтверждения АСК и посылает этот кадр в узел-отправитель, завершая квитирование передачи.

Если V(R) не равен порядковому номеру посылки в кадре или обнаружена ошибка, значит, что-то произошло, и после тайм-аута в узел-отправитель посылается NAK [с порядковым номером приема, содержащим значение V(R)]. В большинстве протоколов этот NAK называется Неприем (REJ) или Выборочный неприем (SREJ). Значение V(R) уведомляет передающее устройство о том, что ожидается посылка нового кадра. Т. к. передатчик восстанавливает старое значение V(S) и повторяет передачу кадра, порядковый номер которого совпадает со значением V(S).

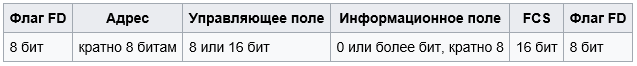
Во многих системах для V(S) и V(R) у порядковых номеров в кадре используются числа 0-7. если переменные состояния в результате последовательного увеличения достигли 7, то, начиная с 0, эти числа снова используются. Вследствие повторного использования чисел устройствам станциям не разрешено посылать кадр с порядковым номером, который не был подтвержден. Например, протокол должен дождаться подтверждения кадра с номером 6, прежде чем он опять использует V(S)=6. Этот процесс показан на (Рисунок 1.1). Здесь кадры с 6 по 4 еще не подтверждены. Если бы был послан еще один кадр с порядковым номером 6, соответствующее подтверждение АСК с номером 6 не позволило бы определить, приход какого кадра с порядковым номером 6 подтверждается.

Использование номеров 0-7 позволяет семи кадрам быть в активном состоянии, прежде, чем "закроется" окно. Несмотря на то что диапазон 0-7 дает восемь порядковых номеров, V(R) содержит значение следующего ожидаемого кадра, что ограничивает число активных кадров до 7.

### **Формат кадра HDLC**

**На канальном уровне используется термин кадр** для обозначения независимого объекта данных, передаваемого от одной станции к другой (Рисунок 1.1).

**Флаг**. Все кадры должны начинаться и заканчиваться полями флага "**01111110**" (0x**7E**). Станции, подключенные к каналу, постоянно контролируют двоичную последовательность флага. Флаги могут постоянно передаваться по каналу между кадрами HDLC. Для индексации исключительной ситуации в канале могут быть посланы семь подряд идущих единиц. Пятнадцать или большее число единиц поддерживают канал в состоянии покоя. Если принимающая станция обнаружит последовательность битов, не являющихся флагом, она тем самым уведомляется о начале кадра, об исключительной (с аварийным завершением) ситуации или ситуации покоя канала. При обнаружении следующей флаговой последовательности станция будет знать, что поступил полный кадр.



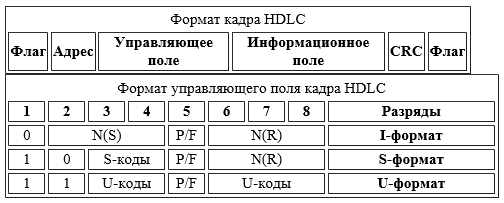


Рисунок 1.1 Формат кадра и управляющего поля HDLC,

где:

N(S) - порядковый номер передаваемого кадра,

N(R) - порядковый номер принимаемого кадра,

P/F - бит опроса/окончания

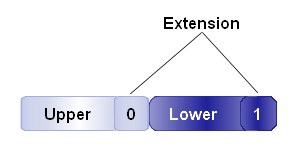
**Адресное поле** определяет первичную или вторичную станции, участвующие в передаче конкретного кадра. Каждой станции присваивается уникальный адрес. В несбалансированной системе адресные поля в командах и ответах содержат адрес вторичной станции. В сбалансированных конфигурациях командный кадр содержит адрес получателя, а кадр ответа содержит адрес передающей станции.



Рисунок 1.2 Правила адресации

#### **Кодирование HDLC адреса для сервера**

Младший бит (LSB) всегда устанавливается в ноль в адресе логического устройства (верхняя часть адреса). В адресе физического устройства (нижняя часть адреса) младший бит устанавливается в единицу.



**Пример:**

*Размер идентификатора сервера – один байт.*

*Адрес логического устройства – 1, и адрес физического устройства – 2.*

*В этом случае, адрес сервера формируется следующим образом:*

*Адрес логического устройства сдвигается (влево) на один, поэтому верхняя часть адреса становится равной 2.*

*Адрес физического сервера сдвигается (влево) на один и младший бит устанавливается в 1, и нижняя часть адреса становится равной 5.*

*Таким образом, в двоичной системе счисления это будет выглядеть 0010 0101, а в шестнадцатеричной 0x25.*

**Управляющее поле** задает тип команды или ответа, а также порядковые номера, используемые для отчетности о прохождении данных в канале между первичной и вторичной станциями. Формат и содержание управляющего поля (Рисунок 1.1) определяют кадры трех типов: информационные (I), супервизорные (S) и ненумерованные (U).

* **Информационный формат** (I - формат) используется для передачи данных конечных пользователей между двумя станциями.
* **Супервизорный формат** (S - формат) выполняет управляющие функции: подтверждение (квитирование) кадров, запрос на повторную передачу кадров и запрос на временную задержку передачи кадров. Фактическое использование супервизорного кадра зависит от режима работы станции (режим нормального ответа, асинхронный сбалансированный режим, асинхронный режим ответа).
* **Ненумерованный формат** (U - формат) также используется для целей управления: инициализации или разъединения, тестирования, сброса и идентификации станции и т.д. Конкретный тип команды и ответа зависит от класса процедуры HDLC.

**Информационное поле** содержит действительные данные пользователя. Информационное поле имеется только в кадре информационного формата. Его нет в кадре супервизорного или ненумерованного формата. [*Примечание: кадры "UI - ненумерованная информация" и "FRMR - Неприем кадра" ненумерованного формата имеют информационное поле*].

**Поле CRC** (контрольная последовательность кадра) используется для обнаружения ошибок передачи между двумя станциями. Передающая станция осуществляет вычисления над потоком данных пользователя, и результат этого вычисления включается в кадр в качестве поля CRC. В свою очередь, принимающая станция производит аналогичные вычисления и сравнивает полученный результат с полем CRC. Если имеет место совпадение, велика вероятность того, что передача произошла без ошибок. В случае несовпадения, возможно, имела место ошибка передачи, и принимающая станция посылает отрицательное подтверждение, означающее, что необходимо повторить передачу кадра. Вычисление CRC называется циклическим контролем по избыточности и использует некоторый производящий полином в соответствии с рекомендацией МККТТ V.41. Этот метод позволяет обнаруживать всевозможные кортежи ошибок длиной не более 16 разрядов, вызываемые одиночной ошибкой, а также 99,9984% всевозможных более длинных кортежей ошибок.

### **Кодонезависимость и синхронизация HDLC**

HDLC является **кодопрозрачным** протоколом. Он не зависит от конкретного кода (ASCII/IA5 или EBCDIC) при выполнении функции управления каналом. Восьмибитовая комбинация флага 01111110 помещается в начале и в конце кадра, чтобы дать возможность приемнику распознать начало и конец кадра. Возможны случаи, когда прикладной процесс помещает в данных пользователя последовательность 01111110, совпадающую с флагом. В этом случае передающая станция в поток выходных данных помещает 0 после 5 подряд идущих единиц, встретившихся в любом месте между начальным и конечным флагами кадра. Такая вставка производится в адресное, управляющее, информационное поля и поле CRC. Этот метод называется вставкой битов (**bit staffing**); такую же функцию выполняет знак DLE в протоколе BSC. После того как завершается вставка битов в кадр и по концам кадра помещаются флаги, кадр передается приемнику по каналу.

Приемник постоянно контролирует поток битов. При получении нуля с пятью далее идущими подряд единицами (011111) анализирует следующий (седьмой) бит. Если это нуль, он удаляет этот бит. Однако если седьмой бит является единицей (0111111), приемник анализирует восьмой бит. Если это нуль (01111110), он считает, что получена флаговая комбинация. Если это единица, выполняется анализ последующих бит. Возможна ситуация приема либо сигнала покоя, либо сигнала аварийного завершения, на которые станция реагирует соответствующим образом. Таким образом, в протоколе HDLC обеспечиваются кодовая прозрачность по данным. Протоколу безразлично, какие кодовые комбинации находятся в потоке данных. Единственное, что требуется, – это поддерживать уникальность флагов.

HDLC используется также два других сигнала: **сигнал аварийного завершения** (**АЗ**) состоит из последовательности единиц, число которых не меньше семи и не больше четырнадцати; **состояние покоя** представляется последовательностью пятнадцати или большего числа единиц.

**Сигнал аварийного завершения** (abort) помещается в конце кадра. Передающая станция посылает этот сигнал, когда возникает исключительная ситуация, требующая восстановления. Вслед за сообщением об аварийном завершении могут посылаться флаги для того, чтобы поддерживать канал в активном состоянии, и передача могла продолжаться.

**Сигнал покоя** означает, что канал находится в состоянии покоя. Одно из применений состояния покоя находит в полудуплексном сеансе, когда при обнаружении сигнала покоя производится изменение направление передачи на противоположное.

**Межкадровое временное заполнение** сопровождается передачей между кадрами непрерывной последовательности флагов. Флаги могут быть восьмибитовыми комбинациями, или же может иметь место совмещение последнего 0 предыдущего флага с первым 0 следующего флага. Например, 01111110011111100111111001111110… или 011111101111111011111110…

### **Управляющее поле HDLC**

Управляющее поле (Рисунок 1.1) определяет тип кадра и используется для реализации механизма управления потоком между передающей и принимающей станциями. На (Рисунок 1.3) представлены команды и ответы, используемые в случае сбалансированной и несбалансированной конфигураций канала. Отметим, что в каждом верхнем прямоугольнике содержатся три команды: SNRM, SARM, SABM.

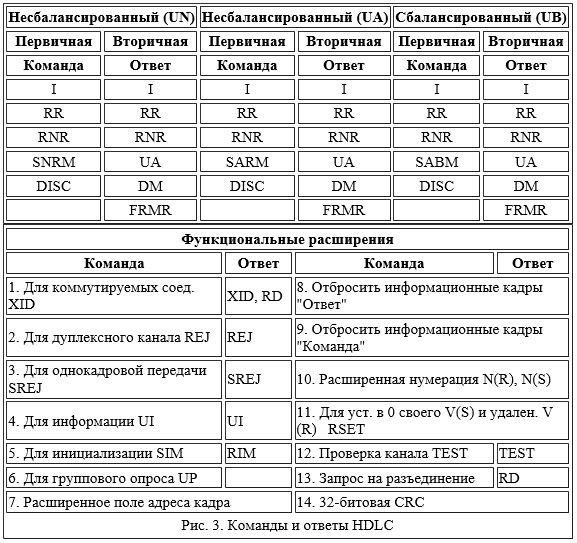


Рисунок 1.3 Команды и ответы HDLC

Эти команды являются командами установки режима. HDLC требует, чтобы в одном из трех режимов была установлена сбалансированная или несбалансированная конфигурация. На рисунке показаны также функциональные расширения (опции) основной структуры. Здесь представлен полный набор команд и ответов. Некоторые подмножества протокола HDLC используют только часть этого набора команд/ответов.

Действительный формат управляющего поля (информационный, супервизорный или ненумерованный) определяет то, как это поле кодируется или используется. Самым простым форматом является информационный формат. Содержимое управляющего поля для этого формата показано на (Рисунок 1.1). Управляющее поле информационного кадра содержит два порядковых номера Номер N(S) (Порядковый номер посылки) связан с порядковым номером передаваемого кадра. N(R) (Порядковый номер приема) означает порядковый номер следующего кадра, который ожидается принимающей станцией. N(R) выступает в качестве подтверждения предыдущих кадров. Например, если поле N(R) установлено в 4, станция, получив N(R)=4, знает, что передача кадров 0, 1, 2 и 3 завершилась успешно и что станция, с которой производится обмен данными, ожидает, что следующий кадр будет иметь порядковый номер посылки N(S)=4. Поле N(R) обеспечивает включающее подтверждение (квитирование), то есть N(R)=4 включает подтверждение не только одного предшествующего сообщения. Переменные состояния посылки V(S) и состояния приема V(R), рассмотренные нами ранее, используются для формирования полей N(S) и N(R) протокола HDLC.

Пятый двоичный разряд, бит P/F или бит опроса/окончания принимается во внимание только тогда, когда он установлен в 1. Бит P/F называется битом P, когда он используется первичной станцией, и битом F, когда он используется вторичной станцией. Он используется первичной и вторичной станциями для выполнения следующих функций:

* Первичная станция использует бит P для санкционирования передачи кадра статуса от вторичной станции. P также может означать опроc.
* Вторичная станция отвечает на бит P кадром данных или состояния с битом F. Бит F может также означать окончание передачи вторичной станцией в режиме нормального ответа (NRM).

Только один бит P (ожидающий ответа в виде F бита) может быть активным в канале в любой момент времени. Если некоторый бит P установлен в 1, он может быть использован в качестве контрольной точки. То есть P=1 как бы говорит: ответьте мне, потому что я хочу знать ваш статус. Контрольные точки играют большую роль в реализации механизма управления трафиком. Это также способ устранения неопределенностей и отмены накопленных транзакции. Бит P/F может использоваться и интерпретироваться следующим образом:

* В режиме NRM вторичная станция не может вести передачу, пока не будет получена команда с установленным в 1 битом P. Первичная станция может запросить информационные (I) кадры путем посылки кадра с установленным в 1 битом P или путем посылки некоторых супервизорных (S) кадров (RR, REJ или SREJ) с установленным в 1 битом P.
* В режимах ARM и ABM информационные кадры могут передаваться без получения полномочий на передачу с помощью команды, имеющей бит P. Поэтому бит P используется в этом случае для запроса ответа с установленным в 1 битом F так быстро, насколько это возможно. Например, в случае двунаправленной одновременной (полнодуплексной) передачи, когда по получении команды с установленным в 1 битом Р передачу ведет вторичная станция, бит F устанавливается в 1 в самом первом очередном ответе. Передача кадра с установленным в 1 битом F не требует, чтобы вторичная станция прекратила передачу. Вслед за кадром с установленным в 1 битом F могут быть еще переданы кадры. В режимах ARM и ABM не следует интерпретировать бит F как окончание передачи вторичной станцией; его следует просто считать индикатором ответа на предыдущий кадр.

### **Описание команд и ответов**

**Супервизорный формат** показан на (Рисунок 1.1) и предусматривает четыре команды и ответа (RR, RNR, REJ, SREJ), которые представлены на рис.4.5. (Обобщенная сводка всех команд и ответов приведена в таблице). Назначение этого формата состоит в выполнении нумерованных [т.е. использующих порядковые номера кадров N(R)] супервизорных функций, таких, как подтверждение (квитирование), опрос, временная задержка передачи данных и восстановление после ошибок. Кадры супервизорного формата не содержат информационного поля, следовательно, как показано на (Рисунок 1.1), в них располагается только порядковый номер приема N(R). Супервизорный формат может быть использован для подтверждения приема кадров от передающей станции.

Функции команд и ответов, используемых супервизорным форматом:

* **RR** (**Receive ready - Готов к приему**) используется первичной или вторичной станцией для индикации того, что станция готова принять информационный кадр и/или подтвердить (квитировать) ранее принятые кадры с помощью поля N(R). Если станция до этого, используя команду "Не готов к приему", посылала уведомление о том, что она занята, теперь она использует команду "Готов к приему" для индикации того, что она свободна и готова принять данные. Первичная станция может также использовать команду "Готов к приему" для опроса вторичной станции.
* **RNR** (**Receive not ready - Не готов к приему**) используется станцией для индикации состояния занятости. Эта команда уведомляет передающую станцию о том, что принимающая станция не способна принять дополнительные поступающие данные. Кадр RNR, используя поле N(R), может подтвердить прием ранее переданных кадров. Состояние занятости может быть сброшено посылкой кадра RR, а также некоторых других кадров, которые будут рассмотрены позднее.
* **REJ** (**Reject - Неприем**) используется для запроса передачи кадров, начиная с кадра, указанного в поле N(R). Подтверждаются все кадры с номерами до N(R) - 1. Кадр REJ может использоваться для реализации метода "Возвращение-на-N" (Go-Back-N).
* **SREJ** (**Selective reject - Выборочный неприем**) используется станцией для запроса повторной передачи единственного кадра, который определен в поле N(R). Как и в случае включающего подтверждения, подтверждение распространяется на все информационные кадры с номерами до N(R) - 1 включительно. Выборочный неприем позволяет реализовать режим выборочного повторения. Как только передан кадр SREJ, следующие кадры принимаются и сохраняются для повторно передаваемого кадра.

**Ненумерованные команды и ответы** используются для посылки большинства индикаторов команд и ответов. Ненумерованные команды можно разбить на группы в соответствии с выполняемыми функциями:

* команды установки режима: SNRM, SARM, SABM, (SNRME, SARME, SABME - для расширенной адресации), SIM, RIM, DISC;
* команды передачи информации: UI, UP;
* команды восстановления: RSET;
* другие команды: XID, TEST, DM, UA, FRMR, RD.

Функции команд/ответов для ненумерованного формата:

**UI** (**Unnumbered information - Ненумерованная информация**). Эта команда позволяет производить передачу данных пользователя в ненумерованном кадре (т. е. без порядкового номера).

**RIM** (**Request Initialization Mode - Режим инициализации запроса**). Кадр RIM является запросом на команду SIM от вторичной к первичной станции.

**SIM** (**Set Initialization Mode - Установить режим инициализации**). Эта команда используется для инициализации сеанса между первичной и вторичной станциями. Ожидаемым ответом является UА.

**SNRM** (**Set Normal Response Mode - Установить режим нормального ответа**). Эта команда переводит вторичную станцию в NRM (режим нормального ответа). NRM предотвращает посылку вторичной станцией несанкционированных (unsolicited) кадров. Это означает, что первичная станция управляет всем потоком сообщений в канале.

**DM** (**Disconnect Mode - Режим разъединения**). Этот кадр ответа передается вторичной станцией для индикации того, что она находится в режиме логического разъединения.

**DISC** (**Disconnect - Разъединить**). Эта команда, передаваемая первичной станцией, переводит вторичную станцию в режим разъединения аналогично нажатию рычага телефонного аппарата.

**UA** (**Unnumbered Acknowledgment - Ненумерованное подтверждение**). Это положительное подтверждение АСК для установки режима команд (SIM, DISC, RESET). UA также используется для уведомления об окончании состояния занятости станции.

**FRMR** (**Frame Rejekt - Неприем кадра**). Вторичная станция посылает этот кадр, когда она встречается с недействительным кадром. Причина указывается в информационном поле этого кадра. Кадр ответа FRMR используется при выполнении следующих условий:

* Прием недействительного управляющего поля команды или ответа.
* Прием слишком длинного информационного поля.
* Прием недействительного поля N(R).
* Прием недопустимого информационного поля или супервизорного/ненумерованного кадра неправильной длины.

В информационном поле кадра FRMR содержится информация о состоянии (status) отвергнутого кадра:

* управляющее поле отвергнутого кадра;
* текущее значение переменных состояния принимающей станции – посылки V(S) и приема V(R);
* отвергнутый кадр был командой или ответом;
* управляющее поле является недействительным;
* кадр был передан с недопустимым информационным полем;
* информационное поле является слишком длинным;
* порядковые номера являются недействительными.

**RD** (**Request Disconnect - Запрос разъединения**). Это запрос от вторичной станции на логическое разъединение и установление состояния логического разъединения.

**XID** (**Exchage State Identification - Идентификация станции при коммутации**). Эта команда запрашивает идентификацию вторичной станции. В системах с коммутацией эта команда используется для идентификации вызывающей станции.

**TEST** (**Test-проверка**). Этот кадр (команда/ответ) используется для санкционирования тестовых ответов от вторичной станции. В ответе желательно сформировать информационную область, содержащуюся в команде.

**SARM** (**Set Asynchronous Response Mode - Установить режим асинхронных ответов**). Устанавливает режим, позволяющий вторичной станции вести передачу без опроса со стороны первичной станции. Он переводит вторичную станцию в состояние передачи информации (IS) режима ARM. Поскольку команда SARM устанавливает две несбалансированные станции, SARM должна выдаваться по обоим направлениям передачи:

* А посылает: B, DISC
* В посылает: B, UA A, DISC
* А посылает: A, UA
* В посылает: A, SARM
* А посылает: A, UA B, SARM
* В посылает: B, UA

Команды DISC посылаются, чтобы гарантировать полную реинициализацию канала.

**SABM** (**Set Asynchronous Balanced Mode - Установить асинхронный сбалансированный режим**). Устанавливает режим в ARM, в котором станции являются равноправными. Для передачи не требуется опроса, поскольку каждая станция является станцией комбинированного типа.

**SNRME** (**Set Normal Response Extended - Установить расширенный режим нормального ответа**). Устанавливает SNRM с двумя дополнительными байтами в управляющем поле.

**SABME** (**Set Asynchronous Balance Mode Extended - Установить расширенный асинхронный сбалансированный режим**). Устанавливает SABM с двумя дополнительными байтами в управляющем поле.

**UP** (**Unnumbered Poll - Ненумерованный опрос**). Команда опрашивает станцию безотносительно к нумерации кадров и квитированию. Если бит опроса установлен в 0, ответ является необязательной возможностью, предоставляемой только для одного ответа. В качестве ответа могут быть переданы неподтвержденные или еще непереданные I-кадры.

**RSET** (**Reset - Сброс переменных**). Передающая станция сбрасывает свой V(S), а принимающая станция свой V(R) с целью возобновления упорядоченной передачи в заданном направлении новой последовательности кадров. Эта команда используется для восстановления. Кадры, которые ранее не были квитированы, остаются таковыми.

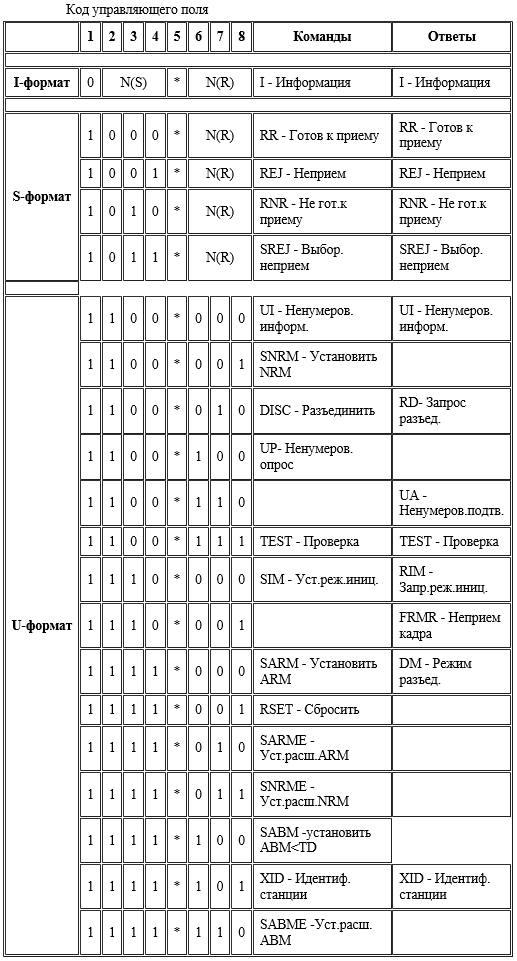
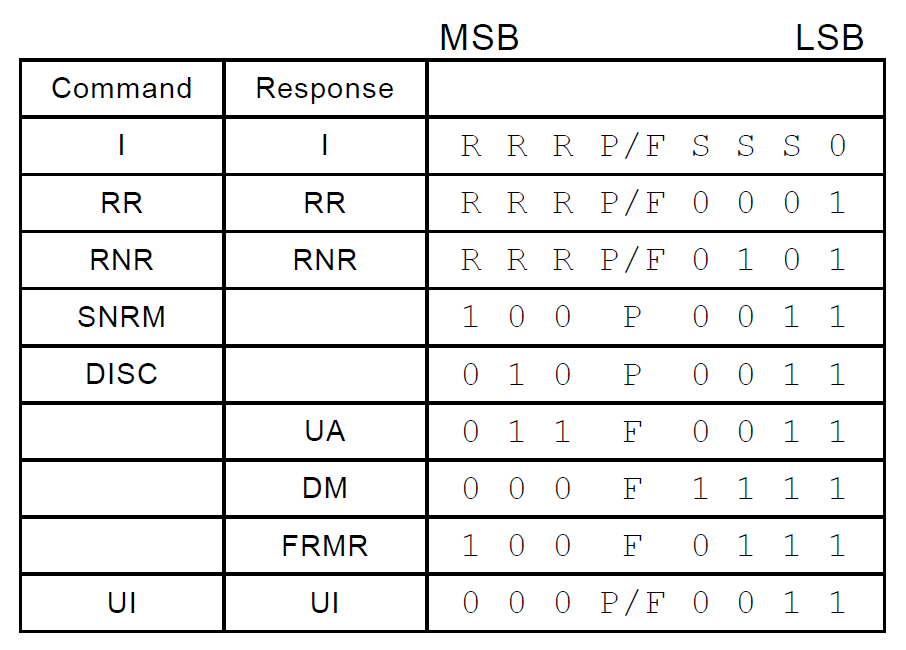


Рисунок 1.4 Команды и ответы



**RRR** is the receive sequence number N(R), **SSS** is the send sequence number N(S) and **P/F** is the poll/final bit.

### **Системные параметры Т1, N2, N1, K и рекомендации по их установке**

**Таймер Т1** запускается с момента передачи каждого кадра и используется для инициирования повторной передачи, в случае его переполнения. При выборе периода таймера Т1 необходимо учитывать, запускается ли таймер по началу или по концу кадра. Для правильной работы процедуры необходимо, чтобы период таймера Т1 был больше, чем максимальное время между передачей некоторого кадра (SARM, SABM, DM, DISC, FRMR, I или супервизорной команды) и приемом соответствующего кадра, возвращаемого в качестве отклика на этот кадр (UA, DM или подтверждающий кадр).

**Счетчик N2** используется для определения максимального числа повторных передач, выполняемых по переполнении таймера Т1. Переменные Т1 и N2 используются также командами/ответами установления звена, такими, как SABM и UA.

**Счетчик N1** – максимальное число битов в I-кадре. Определяет максимальную длину информационных полей.

**Размер окна К** – максимальное число не подтвержденных I-кадров кадров, т.е. которые можно передать, не ожидая подтверждения. Это максимальное число последовательно пронумерованных I-кадров, которые в любой момент времени станции могут передать без получения подтверждения. Оно не должно быть более 7 при нумерации 23.

Параметры Т1, N2, N1 и K являются системными в стеке протокола X25/2, подлежащими согласованию с администрацией на некоторый период времени.

### **Пример**

**Обмен SNRM/UA кадрами с обсуждением параметров HDLC**

Sent frame:

0x7E A00A 0002 0023 21 93 1871 7E

7EA00A00020023219318717E

Комменнтарий:

**7E**// флаг открытия HDLC кадра

**A00A**// тип кадра и длина

**0002**// адрес получателя (сервер) верхняя часть HDLC адреса 0x0001

**0023**// адрес получателя (сервер) нижняя часть HDLC адреса 0x0011

**21**// адрес отправителя (клиент) 0x01

**93**// тип кадра SNRM (Set Normal Response Mode – установить режим нормального ответа)

// информационное поле с HDLC параметрами отсутствует, предполагается по умолчанию

**1871**// CRC последовательность (код проверки целостности кадра)

**7E**// флаг закрытия HDLC кадра

Received frame:

7EA023210002002373F6C58180140502008006020080070400000001080400000001CE6A7E

Комменнтарий:

**7E**// флаг открытия HDLC кадра

**A023**// тип кадра и длина

**21**// адрес получателя (клиент) 0x01

**0002**// адрес отправителя (сервер) верхняя часть HDLC адреса 0x0001

**0023**// адрес отправителя (сервер) нижняя часть HDLC адреса 0x0011

**73**// тип кадра UA (Ненумерованное подтверждение)

**F6C5**// Код проверки целостности заголовка (Header check sequence)

**818014**// идентификатор формата / идентификатор группы / длина группы

**05020080**// идентификатор параметра/длина/значение – максимальная длина информационного поля для передачи

**06020080**// идентификатор параметра/длина/значение – максимальная длина информационного поля для приема

**070400000001**// идентификатор параметра/длина/значение – размер окна передачи

**080400000001**// идентификатор параметра/длина/значение – размер окна приема

**CE6A** // Код проверки целостности кадра (Frame check sequence)

**7E** // флаг закрытия HDLC кадра

### **Examples for DLMS/COSEM information exchange**

prepared by Gyozo Kmethy

#### **SNRM/UA frame exchange with HDLC parameter negotiation**

Sent frame:

7EA00A00020023219318717E

Comments:  
7E// HDLC frame opening flag  
A00A// frame type and length  
0002// destination address (server) upper HDLC address 0x0001  
0023// destination address (server) lower HDLC address 0x0011  
21// source address (client) 0x01  
93// frame type SNRM (Set Normal Response Mode)   
// information field with HDLC parameters not present, defaults are proposed  
1871// Frame check sequence  
7E// HDLC frame closing flag  
  
Received frame:

7EA023210002002373F6C58180140502008006020080070400000001080400000001CE6A7E

Comments:   
7E// HDLC frame opening flag  
A023// frame type and length  
21// destination address (client) 0x01  
0002// source address (server) upper HDLC address 0x0001  
0023// source address (server) lower HDLC address 0x0011  
73// frame type UA (Unnumbered acknowledge)  
F6C5// Header check sequence  
818014//format identifier / group identifier / group length  
05020080//parameter identifier/length/value – maximum information field length transmit  
06020080//parameter identifier/length /value – maximum information field length receive   
070400000001//parameter identifier/length/value – window size transmit  
080400000001//parameter identifier/length/value – window size receive  
CE6A // Frame check sequence  
7E // HDLC frame closing flag

#### **AARQ/AARE exchange**

Sent frame: SNRM

7EA00A00020023219318717E

Received frame: UA

7EA023210002002373F6C581801405020080060200800704000000010804  
00000001CE6A7E

Sent frame: AARQ

7EA02E0002002321107ECBE6E600601DA109060760857405080101BE10  
040E01000000065F1F040000301DFFFFD4C57E

Comments:  
7E// opening flag  
A02E// frame type and length  
0002002321// destination and source addresses  
10// frame type I frame  
7ECB// Header check sequence  
E6E600// LLC bytes  
601D// AARQ tag and length  
A109060760857405080101//Application context name tag, length and encoded value  
// This application association is established with lowest level security, so authentication components are not present  
BE10// tag and length for AARQ user field  
040E// encoding the choice for user-information (OCTET STRING, universal) and length  
01// tag for xDLMS-Initiate request  
00// usage field for dedicated-key component – not used  
00// usage field for the response allowed component – not used  
00// usage field of the proposed-quality-of-service component – not used  
06// proposed dlms version number 6  
5F1F// tag for conformance block  
04// length of the conformance block  
00// encoding the number of unused bits in the bit string   
00301D//conformance block  
FFFF// client-max-receive-pdu-size  
D4C5// HDLC frame check sequence  
7E// closing flag  
  
Received frame:

7EA03A2100020023309941E6E7006129A109060760857405080101A20302  
0100A305A103020100BE10040E0800065F1F040000301D190000070C527E

Comments:  
7EA03A2100020023309941//HDLC frame header  
E6E700// LLC bytes  
6129//AARE tag and length  
A109060760857405080101//Application context name tag, length and encoded value  
A203// tag and length of the result component  
02//encoding the choice for result (INTEGER, universal)  
0100// length and value of result (accepted)  
A305// tag and length for the result-source-diagnostic component  
A103// tag and length of the acse-service-user choice  
02// encoding the choice for result-source-diagnostic (INTEGER, universal)  
0100// length and value of result-souce-diagnostic  
BE10//tag and length for AARE user-field  
040E// encoding the choice for user-information (OCTET STRING, universal) and length   
08// tag for xDLMS-Initate.response   
00// usage field of the negotiated-quality-of-service component  
06// negotiated dlms version number 6  
5F1F// tag for conformance block  
04// length of the conformance block  
00// encoding the number of unused bits in the bit string   
00301D//negotiated conformance block  
1900// server-max-receive-pdu-size   
0007// VAA name (0x0007 for LN referencing)  
0C52// HDLC frame check sequence  
7E// closing flag

#### **Reading and setting the clock**

Sent frame:

7EA0199575546835E6E600C0018100080000010000FF01000DFD7E

Comments:  
7EA0199575546835// HDLC header  
E6E600// LLC bytes  
C001// GET.request.normal  
81// invoke-id and priority  
0008// interface class 8, clock  
0000010000FF// logical name, OBIS code of the clock  
0100// asking for 1st attribute, logical name  
0DFD7E// HDLC frame closing  
  
Received frame:

7EA018759574E9E8E6E700C401810009060000010000FFFD497E

Comments:  
7EA018759574E9E8E6E700  
C401//GET.response.normal  
81// invoke-id and priority  
00// Get-Data-Result choice data  
0906// octet string (6)  
0000010000FF// logical name, OBIS code of the clock  
FD497E// HDLC closing  
  
Sent frame:

7EA0199575767837E6E600C0018100080000010000FF020065D77E

Comments:  
7EA0199575767837E6E600  
C0018100080000010000FF0200// GET.request normal, 2nd attribute, time  
65D77E  
  
Received frame:

7EA01E7595966F67E6E700C4018100090C07D20C04030A060BFF007800F3307E

Comments:  
7EA01E7595966F67E6E700// HDLC header  
C401// GET.response.normal  
8100  
090C// octet string (12)  
07D2// year 2002  
0C// month December  
04// day 4th  
03// day of the week, Wednesday  
0A060B// time 10:06:12  
FF// hundredths not specified  
0078// deviation 120 minutes  
00// status OK  
F3307E  
  
Sent frame:

7EA027957598B8DBE6E600C1018100080000010000FF0200090C07D20C04030A060BFF00780062FB7E

Comments:  
7EA027957598B8DBE6E600  
C101// SET.normal request  
81  
0008  
0000010000FF  
0200// set the 2nd attribute, time  
090C07D20C04030A060BFF007800// just SET the value read before  
62FB7E  
  
Received frame:

7EA0107595B85101E6E700C501810036CF7E

Comments:  
7EA0107595B85101E6E700  
C5018100// SET. response normal, success  
36CF7E  
  
Sent frame:

7EA0199575BA183BE6E600C0018100080000010000FF0300BDCE7E

Comments:  
7EA0199575BA183BE6E600  
C0018100080000010000FF0300// GET.request normal, 3d attribute,   
//time\_zone  
BDCE7E  
  
Received frame:

7EA0137595DA8864E6E700C4018100100078563A7E

Comments:  
7EA0137595DA8864E6E700  
C4018100// GET.response normal  
10// integer 16  
0078// 120 minutes  
563A7E  
  
Sent frame:

7EA01C9575DC7F53E6E600C1018100080000010000FF030010007801177E

Comments:  
7EA01C9575DC7F53E6E600  
C1018100080000010000FF// SET.request normal  
0300// third attribute  
100078 //integer 16, 120 minutes   
01177E  
  
Received frame:

7EA0107595FC7105E6E700C501810036CF7E

Comments:  
7EA0107595FC7105E6E700  
C5018100// SET.response normal, success  
36CF7E  
  
Sent frame:

7EA0199575FE383FE6E600C0018100080000010000FF0400B5837E

Comments:  
7EA0199575FE383FE6E600  
C0018100080000010000FF0400// GET.request normal, 4th attribute, status  
B5837E  
  
Received frame:

7EA01275951E1BF8E6E700C40181001100AC5B7E

Comments:  
7EA01275951E1BF8E6E700  
C40181  
001100// Data, unsigned8, OK  
AC5B7E  
  
Sent frame:

7EA0199575104831E6E600C0018100080000010000FF05006D9A7E

Comments:  
7EA0199575104831E6E600C0018100080000010000FF0500// GET.request normal, 5th attribute,   
// daylight savings begin6D9A7E  
  
Received frame:

7EA01E75953053A7E6E700C4018100090CFF4F19F3007000000000000F090A7E

Comments:  
7EA01E75953053A7E6E700  
C40181  
00090C// data, octet string(12)  
FFFFFFFFFFFFFFFFFF800000// insignificant value  
090A7E  
  
Sent frame:

7EA0199575325833E6E600C0018100080000010000FF060005B07E

Comments:  
7EA0199575325833E6E600C0018100080000010000FF0600// GET.request normal, 6th attribute,   
// daylight\_savings\_end05B07E  
  
Received frame:

7EA01E75955247E7E6E700C4018100090CFF4F19F3007000000000000F090A7E

Comments:  
7EA01E75955247E7E6E700  
C40181  
00090C// data, octet string (12)  
FFFFFFFFFFFFFFFFFF800000// insignificant value  
090A7E  
  
Sent frame:

7EA0199575546835E6E600C0018100080000010000FF0700DDA97E

Comments:  
7EA0199575546835E6E600C0018100080000010000FF0700// GET.request normal, 7th attribute, //daylight savings deviationDDA97E  
  
Received frame:

7EA0127595744734E6E700C40181000F002D547E

Comment:  
7EA0127595744734E6E700  
C40181  
000F00//data, integer8, 0  
2D547E  
  
Sent frame:

7EA0199575767837E6E600C0018100080000010000FF0800152A7E

Comments:  
7EA0199575767837E6E600  
C0018100080000010000FF0800// GET.request normal, 8th attribute,  
//daylight\_savings\_enabled  
152A7E  
  
Received frame:

7EA0127595965BF0E6E700C401810003008DFD7E

Comments:  
7EA0127595965BF0E6E700  
C40181  
000300// data, boolean, FALSE  
8DFD7E  
  
Sent frame:

7EA0199575980839E6E600C0018100080000010000FF0900CD337E

Comments:  
7EA0199575980839E6E600  
C0018100080000010000FF0900// GET.request normal, 9th attribute,  
//clock\_base  
CD337E  
  
Received frame:

7EA0127595B82738E6E700C401810016012D077E

Comments:  
7EA0127595B82738E6E700  
C40181  
001601// data, enumerated, 1, internal crystal  
2D077E

## **Прикладной уровень**

Последним из трех уровневого профиля DLMS является **прикладной уровень** (**Application layer**). После того, как установлена связь на физическом и HDLC уровнях, нам необходимо соединиться на прикладном уровне. В OSI терминологии этот тип соединения называется **ассоциация** (**Association**). Поэтому, чтобы создать ассоциацию, мы посылаем запрос на ассоциацию (соединение) и ожидаем подтверждения ассоциации. Потом, если ассоциация была корректно установлена, мы продолжаем посылать запросы и принимать ответы пока не завершим. Наконец, мы освобождаем ассоциацию отсоединением на HDLC уровне.

Все запросы и соответствующие ответы определены в стандарте использующим ASN.1 (Abstract Syntax Notation One) нотацию. Однако, в наших инструментах и библиотеках, конкретные примеры запросов и ответов представлены в XML-формате, как показано ниже:

### 4.2.1 Example of Logical Device Name readout

В этом примере считывается логическое имя устройства (Logical Device Name).

Client → Server: (I frame) READ REQUEST

7E A0 14 00 02 4A 19 41 32 EB AC E6 E6 00 05 01 02 FD 08 10 D3 7E

7E // HDLC frame opening flag

A0 14 // frame type [bit 12…15] = 0xA, segmentation [bit 11] =0, length =0x14=20;

00 02 // destination address (server) upper HDLC address = 0x0001

4A 19 // destination address (server) lower HDLC address = 0x128C = 4748

41 // source address (client) = 0x20=32

32 // FrameType I [bit 0]=0; SendNr: [bit 1..3]=1; PF: [bit 4]=1; RecNr: [bit 5..7]=1

|  |  |  |  |
| --- | --- | --- | --- |
|  | Company:  PZIP Ltd. | Departament:  Development and Design | Data:  2015-03-24 |

EB AC // Header Check Sequence (HCS)

E6 E6 00 // LLC bytes

05 // COSEM APDU: Read Request

1. // Number of elements in sequence=1;
2. // VariableAccessSpecification = CHOICE[2] variable-name;

FD 08 // Short Name = 0xFD00 (object: Logical Device Name; attribute: value);

10 D3 // HDLC frame check sequence (FCS) 7E // HDLC frame closing flag

Server → Client: (I frame) READ REASPONSE

7E A0 1F 41 00 02 4A 19 52 C2 2B E6 E7 00 0C 01 00 0A 0B 45 47 4D 37 35 33 37 34 38 00 00 AD 9B 7E 7E // HDLC frame opening flag

A0 1F // frame type [bit 12…15] = 0xA, segmentation [bit 11] =0, length =0x1F=31;

41 // destination address (client) = 0x20

00 02 // source address (server) upper HDLC address = 0x0001

4A 19 // source address (server) lower HDLC address = 0x128C = 4748

52 // FrameType I [bit 0]=0; SendNr: [bit 1..3]=1; PF: [bit 4]=1; RecNr: [bit 5..7]=2

C2 2B // Header Check Sequence (HCS)

E6 E7 00 // LLC bytes

0C // COSEM APDU: Read Response 01 // Number of elements in sequence=1; 00 // success (not error):

0A 0B // visible string (size(11)):

45 47 4D 37 35 33 37 34 38 00 00 // EGM753748

AD 9B // HDLC frame check sequence (FCS)

7E // HDLC frame closing flag

*Remark*. Value of Logical Device Name “EGM753748” contains manufacturer identifier “EGM” and meter serial number “753748”.

Максимальный размер данных в LoRa сообщении на 0-7 датарейте соотественно 59-230(250) байт.

Максимальное значение длины информационного поля в DLMS 2030 байт.

Ограничение на вся длину сообщения (с заголовками) по оперативной памяти модуля 512 байт.