

Описание USB Шины

С примерами реализации для USBN9604 и AT90S8515.

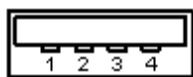
Это техническое описание включает в себя описание стандарта передачи данных по **USB** шине, а также пример реализации USB устройства на базе микроконтроллера AT90S8515 и USB-контроллера фирмы National Semiconductors USBN9604.

Вся информация, представленная в этом документе, может быть использована только для ознакомительных целей. Ни одна из глав этого документа не может быть использована в какой либо форме без предварительного согласия с **KS Labs**.

Глава 1. Знакомство с USB шиной. Аппаратные вопросы.

1.0 Разъемы.

Все устройства имеют исходящий поток данных от устройства в Хост, а Хост, в свою очередь, имеет нисходящий поток данных в устройство. Для подключения USB устройств и Хабов к компьютеру существуют два типа разъемов: А и В. Соответственно для исходящих и нисходящих потоков данных.



USB разъём Тип А



USB разъём тип В

Рисунок 1. Типы USB разъемов.

Разъем А всегда символизирует исходящий поток данных. Типичное месторасположение А разъема это Хосты и Концентраторы. Их вы всегда можете увидеть на материнских платах. Разъем В всегда символизирует нисходящий поток данных. Их вы всегда можете увидеть на любом периферийном оборудовании подключаемом к USB шине. Если вы разрабатываете USB устройство не являющееся Концентратором и оно не имеет штатного (запаянного в плату) USB кабеля, то на плате должен стоять именно USB В разъем.

Для USB 2.0 позиционируется новый тип разъема Mini-USB В. Аргументация применения нового мини разъема исходила из того, что все миниатюрные электронных устройства например такие как: мобильные телефоны и органайзеры имеют очень маленький размер. Размеры же стандартного В разъема слишком велики для простой интеграции его в эти устройства. Ниже в таблице показана распиновка USB разъема.

Номер контакта	Цвет проводника	Функция
1	Красный	+5В
2	Белый	D-
3	Зеленый	D+
4	Черный	GND

Стандартная цветовая гамма проводников внутри USB кабеля облегчает идентификацию проводов при применении кабелей от разных производителей.

Стандарт определяет различные электрические параметры для кабелей. Существует два типа кабелей: для низкоскоростных устройств (со скоростью передачи данных 1.5МБит) и для полноскоростных устройств (со скоростью передачи данных 12МБит). Кабель для низкоскоростных устройств не требует наличия витой пары для сигнальных проводников или общего экрана. Кабель для полноскоростных устройств имеет витую пару из сигнальных проводников и защитный экран. Он может использоваться как для низкоскоростных так и для полноскоростных устройств.

1.1 Электрические характеристики шины USB.

Если Вы не являетесь производителем микросхем трансиверов и контроллеров шины USB, а также микросхем ХОСТ-контроллеров и Концентраторов, то Вам не обязателен курс электрических характеристик USB шины, изложенный в Главе 7 USB спецификации. Ниже будут приведены основные позиции этой главы. Для передачи данных по шине используется дифференциальная пара. Сигналы синхронизации и данные кодируются по методу БВН (NRZI). В этой кодировке логическая “1” представлена неизменным уровнем на протяжении битового интервала, а логический “0” – представляет собой смену уровня на противоположный (фронт) на протяжении битового интервала. Для низкоскоростных и полноскоростных устройств дифференциальная “1” передается путем подтяжки линии D+ к напряжению более 2.8 Вольта, а линии D- к напряжению менее 0.3В. При этом линии D+ и D- терминированы на стороне Хоста (нисходящего потока) резисторами 15кОм подключенными к земле). Дифференциальный “0” передается путем подтяжки линии D+ к напряжению менее 0.3 Вольта, а линии D- к напряжению более 2.8 Вольта. Приемник определяет дифференциальную единицу только в том случае, когда напряжение на линии D+ больше на 200мВ чем на линии D-, а на дифференциальный “0” когда напряжение на линии D+ меньше на 200мВ чем на линии D-. Полярность сигналов на шине инвертируется в зависимости от скорости подключенного устройства (низкоскоростное и полноскоростное). Следовательно состояния “J” и “K” используются для определения логических уровней. Для низкоскоростных устройств состояние “J” представляет собой дифференциальную “1”, а для полноскоростных устройств дифференциальный “0”.

USB трансиверы имеют как двуполярный так и однополярный выход. Определенные состояния шины указываются путем выставления на линию D+ или D-, или одновременно на обе линии однополярных сигналов. Например, состояние шины SE0, должно обозначать сброс подключенного устройства при удержании этого состояния на шине более 10 мс.

Состояние SE0 генерируется путем удержания сигнальных линий D+ и D- в низком уровне (менее 0.3В). Низкоскоростные и полноскоростные устройства имеют сопротивление линии 90Ом \pm 15%, поэтому рекомендуется тщательно читать описание применяемых микросхем USB трансиверов для установки резисторов последовательно в линии D+ и D-.

1.2 Идентификация подключенных к USB шине устройств.

USB устройство должно указывать свою скорость путем подтяжки линии D+ или D- посредством сопротивления 1.5кОм \pm 5% к напряжению 3.3 вольта. Полноскоростное устройство, изображенное ниже, использует подтягивающий резистор подключенный к линии D+, чтобы ХОСТ смог определить его как полноскоростное устройство. Это подтягивающее сопротивление ХОСТ или Концентратор используют для обнаружения подключения устройства к своему порту. Без этого резистора ни ХОСТ, ни Концентратор не поймут, что к USB шине подключено новое устройство. Некоторые микросхемы контроллеров USB шины имеют встроенный подтягивающий резистор, который может подключаться/отключаться к напряжению 3.3В под управлением программы, другие контроллеры требуют наличие внешнего подтягивающего резистора.

Например, в микросхемах фирмы National Semiconductors USBN9603/4, имеется возможность программного управления подключением/отключением устройства от USB шины. Когда устройство подключается к USB шине, сначала выполняется процедура инициализации контроллера, лишь затем программа устанавливает в регистре контроля подключение подтягивающего резистора.

Лишь со времени подключения резистора к напряжению 3.3В, ХОСТ считает что на шине появилось новое устройство. Теперь ХОСТ может инициировать сброс устройства и запрос дескрипторов, которые описывают его функциональные особенности и возможности.

Другие производители микросхем например, такие как, Cypress Semiconductors также используют программируемый резистор, только уже для процесса ре-эnumерации Re-NumerationTM. На EzUSB устройствах выполнен очень интересный принцип загрузки программы в устройство. При первом включении микроконтроллер эnumерируется в системе с VID и PID принадлежащими компании Cypress Semiconductors и выполняет загрузку в внутреннее ОЗУ программы пользователя. После этого происходит отключение подтягивающего резистора от шины питания (отсоединение устройства) и управление передается загруженной программе пользователя, которая, в свою очередь, снова подключает подтягивающий резистор иницируя подключение нового устройства.

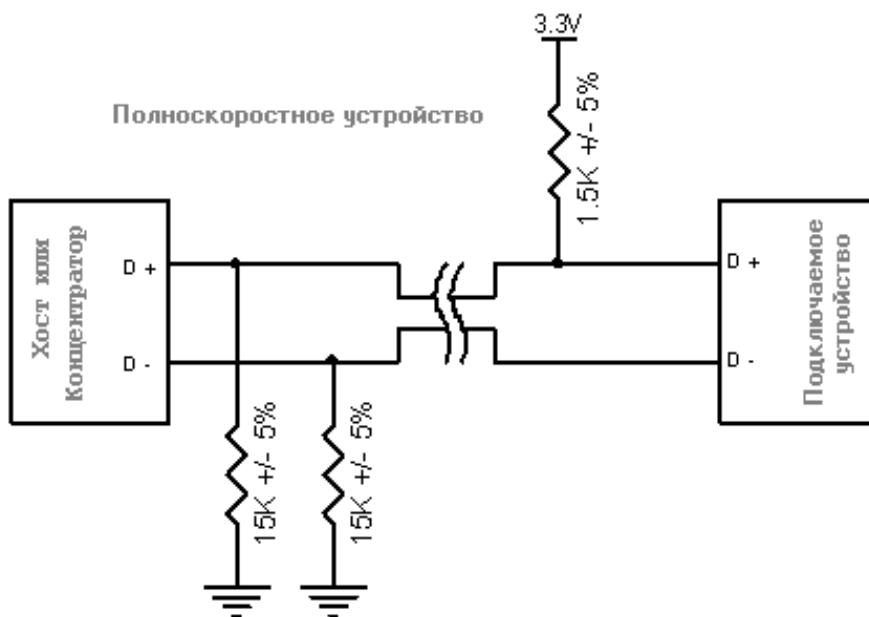


Рисунок 2. Полноскоростное устройство с подтягивающим резистором подключенным к линии D+.

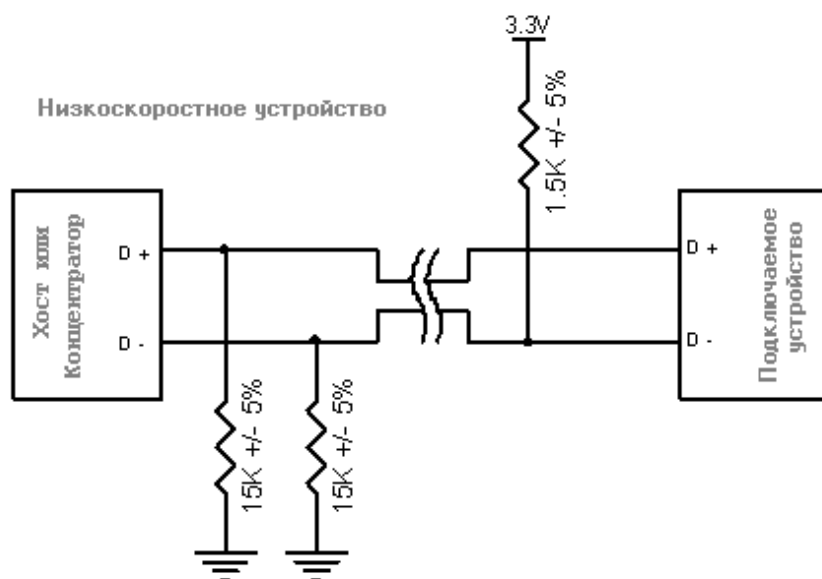


Рисунок 3. Низкоскоростное устройство с подтягивающим резистором подключенным к линии D-.

Высокоскоростные устройства подключаются к USB шине также как и полноскоростные (1.5кОм на 3.3В). Как только устройство подсоединено к шине оно будет осуществлять высокоскоростной обмен в фазе сброса, и будет использовать высокоскоростной обмен если Концентратор его поддерживает.

Если устройство работает в высокоскоростном режиме, то подтягивающий резистор затем отключается, чтобы сбалансировать линию.

1.3 Питание USB устройств.

Одним из преимуществ USB шины являются так называемые “Bus-powered devices” – устройства которые берут питание непосредственно с шины, и не требуют подключения каких либо других разъемов и кабелей питания. USB устройство указывает потребляемую мощность выраженную в мА в дескрипторе конфигурации, который мы рассмотрим немного позже. Причем потребляемый от шины ток указывается с дискретностью 2мА. Например, если устройство потребляет ток от USB шины 100мА, то в дескрипторе конфигурации должна фигурировать цифра 50 ($50 \times 2 = 100\text{мА}$). Потребление тока устройством не должно превышать значения, указанного в процессе эnumерации. Потребляемая мощность определяется для разных классов устройств и выражается в блоках. По USB спецификации величина блока соответствует 100 мА. Существует три класса USB функций (устройств):

- Устройства с малым потреблением и питанием от USB шины
- Устройства с большим потреблением и питанием от USB шины
- Устройства с собственным источником питания

Устройства с большим потреблением (**High-power bus-powered functions**) не должны потреблять более 1 блока мощности (100 мА) при включении, и могут потреблять до 5 (500 мА) блоков после конфигурирования. Эти устройства должны эnumерироваться и при проседании напряжения питания до 4.4В. Они также должны быть полностью работоспособны и при полной токовой нагрузке (вплоть до пяти блоков – 500 мА) с напряжением питания на шине $V_{bus} = 4.75\text{В}$, причем измерение напряжения производится на нисходящем конце кабеля.

Устройства с собственным источником питания могут потреблять ток от USB шины не превышающий размер одного блока (100 мА), а остальную мощность потреблять от собственного блока питания. При пропадании напряжения от собственного источника питания эти устройства должны обеспечить потребление тока от USB шины не превышающее размер одного блока.

Ток потребления обычного устройства с питанием от USB шины не должен превышать 500мА. Также устройство должно поддерживать Режим низкого энергопотребления, так называемый **Suspend Mode**.

1.4 Режим низкого энергопотребления.

Режим низкого энергопотребления обязателен для всех USB устройств. Ток потребления в энергосберегающем режиме пропорционален указанному потреблению в блоках. Для устройства с потреблением тока в 1 блок (100мА) ток в энергосберегающем режиме не должен превышать 500 мкА.

Это с учетом тока через подтягивающий резистор подключенный к одной из сигнальных линий. На концентраторах обе сигнальные линии D+ и D- имеют 15кОм резисторы притягивающие их к земле. Поэтому в целях энергосбережения суммарное сопротивление подключенное к шине питания +3.3В составляет 16.5кОм. При этом ток потребления через эту цепочку резисторов составляет 200мкА. Также большой ток потребляют регуляторы напряжения на 3.3В встроенные в микросхемы контроллеров USB шины. А для полной уверенности, что разрабатываемое вами устройство не выйдет за недопустимую черту потребления тока в режиме энергосбережения вам необходимо погрузить и управляющий микроконтроллер в режим 'сна' (Idle). Многие разработчики задают вопросы в форуме на официальном сайте www.usb.org к какого рода осложнениям может привести превышение этого предела? На данный момент множество ХОСТов и Концентраторов не имеют возможности детектирования такого рода превышений, для них потребление тока в энергосберегающем режиме размером в 5 или 10 мА является нормой. Тем не менее при нормальной работе, если вы пытаетесь превысить лимит в 100 мА или указанную вами допустимую нагрузку, Концентратор отключит ваше устройство от USB шины.

Конечно эти вопросы разработки могут отпасть сами собой если Вы остановились на устройстве с собственным источником питания. Превышающий ток потребления в энергосберегающем режиме не является большим препятствием для настольных компьютеров, но с введением спецификации USB On-The-Go, мы увидим, что USB ХОСТ будет уже встроен в мобильные телефоны, мобильные органайзеры, видео камеры. И лишнее потребление мощности не продлит рабочий срок жизни батарей мобильных устройств.

1.4 Вход в режим низкого энергопотребления.

USB устройство входит в режим пониженного энергопотребления, когда на шине нет никакой активности более 3 мс. У USB устройства есть еще 7 мс для того, чтобы погрузить себя в состояние “сна” и установить ток потребления от шины не более заявленного. Чтобы поддерживать связь с ХОСТом в состоянии пониженного энергопотребления, устройство должно обеспечить протекание тока через подтягивающий резистор выбора скорости, наличие которого будет сигнализировать о том, что устройство все еще подключено к шине. USB ХОСТ-контроллер периодически может выдавать на шину сигнал начала пакета (SOF), для того, чтобы предотвратить погружение подключенных устройств в энергосберегающий режим.

- Для высокоскоростной шины длительность между двумя микро-кадрами составляет $125.0 \mu s \pm 62.5 ns$
- Для полноскоростной шины длительность между двумя микро-кадрами составляет $1.000 ms \pm 500 ns$.

Термин “Глобальное энергосбережение” используется в тех случаях, когда полностью вся шина погружается в режим пониженного энергопотребления.

Каждое устройство в отдельности тоже может быть погружено в состояние “сна”, для этого ХОСТ инициирует команду Концентратору погружения в сон устройства, подключенного к его нисходящему порту. Этот момент называется “Селективный режим пониженного энергопотребления”.

Устройство возобновит свою работу, как только на шине будет замечена любая активность. Если USB устройство имеет особенность “Remote wakeup”, то оно может сообщить в ХОСТ о своем желании возобновить с ним связь.

1.5 Скорость передачи данных по USB шине.

Другая область, которая часто игнорируется при прочтении спецификации, это – допуски на частоту передачи данных по USB шине. Более подробно об этом рассматривается в главе 7.1.11 USB спецификации.

- Высокоскоростные данные синхронизируются частотой 480 МГц, с допуском $\pm 500\text{ppm}$
- Полноскоростные данные синхронизируются частотой 12.000 МГц, с допуском 2500ppm.
- Низкоскоростные данные синхронизируются частотой 1.5 МГц, с допуском 15000ppm.

Глава 2. Описание протоколов используемых при передаче данных.

2.0 Структура USB пакета.

В отличие от RS-232 и аналогичных последовательных интерфейсов, где формат посылаемых данных не имеет четко организованной структуры, для USB шины разграничены различные слои протоколов. Пусть эти слова не пугают Вас пока Вы не ознакомились полностью со всей структурой пакета. Как только Вы поймете что происходит на шине, то вы разберетесь, что Вас затронут протоколы лишь верхнего уровня, всё канальное кодирование, детектирование событий и ошибок возьмет на себя USB контроллер.

Каждая USB транзакция состоит из следующих частей:

- Признак пакета (Заголовок определяющий что далее будет следовать)
- Пакет данных (Опционально)
- Пакет статуса транзакции (Используется для подтверждения нормального завершения транзакции и обеспечения коррекции ошибок при передаче)

Как было указано выше мастером на USB шине является ХОСТ. Он и является инициатором всех транзакций. Первым передаваемым пакетом является Признак (**Token Packet**). Он генерируется ХОСТом для определения что будет следовать далее, какая это будет транзакция: чтение или запись, к какому устройству и конечной точке будет обращение. Следующим пакетом, как обычно, следует пакет данных (**Data Packet**). И в завершении транзакции следует пакет статуса (**Status Packet**), который либо подтверждает успешное принятие данных, либо указывает что конечная точка приостановлена (**STALL**) или не готова принимать данные.

2.1 Поля USB пакета.

Данные по USB шине всегда передаются младшими битами вперед. USB пакет состоит из следующих полей:

- **Поле синхронизации (Sync Field)**

Все пакеты должны начинаться с поля синхронизации. Поле синхронизации имеет размер 8 бит для низкоскоростных и полноскоростных устройств или 32 бита для высокоскоростных устройств и используется для подсинхронизации тактового генератора, встроенного в USB контроллер. Последние два бита поля синхронизации являются маркером, который используется для идентификации конца области синхронизации и начала PID поля.

- **Поле идентификатора пакета (PID Field)**

Идентификатор пакета следует непосредственно после поля синхронизации в каждом передаваемом USB пакете. PID состоит из четырехбитного типа пакета, следующего за четырехбитным проверочным полем как показано на рисунке 4.

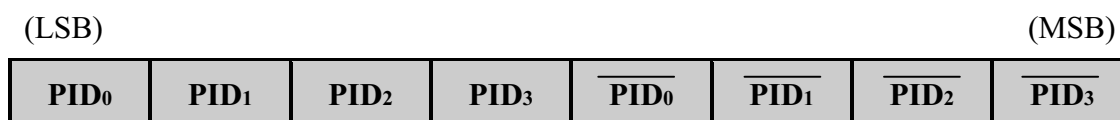


Рисунок 4. Формат поля PID.

Контрольная область PID генерируется как инверсия четырехбитного типа пакета и необходима для устранения ложного декодирования следующих за этим полем данных. Ошибка PID детектируется в том случае, когда инвертированная контрольная область не совпадает с соответствующими битами идентификатора пакета. Любой PID полученный с ошибкой или неуказанный в нижеследующей таблице, должен быть проигнорирован получателем пакета.

PID Тип	Имя PID	PID[3...0]	Описание
Признак (Token)	Out	0001	Адрес + номер конечной точки для транзакции ХОСТ → Функция
	In	1001	Адрес + номер конечной точки для транзакции ХОСТ ← Функция
	SOF	0101	Маркер начала пакета и его номер
	SETUP	1101	Адрес + номер конечной точки для SETUP транзакции ХОСТ → Функция в канале управления
Данные	DATA0	0011	Четный PID пакета данных
	DATA1	1011	Нечетный PID пакета данных
Статус	ACK	0010	Приемник принял пакет данных без ошибок
	NAK	1010	Принимающее устройство не может принять данные, или передающее не может их отправить
	STALL	1110	Конечна точка приостановлена (HALT) или запрос по управляющему каналу не поддерживается
Спец.	PRE	1100	ХОСТ инициирует преамбулу, разрешая трафик для низкоскоростных устройств

Рисунок 5. Типы PID.

PID биты приведены в таблице в MSB порядке. Когда они передаются по шине, правый бит (бит 0) выдается первым.

- **Поле адреса устройства (Addr Field)**

Поле адреса используется для идентификации к какому устройству направлен текущий пакет. Размерность поля - 7 бит, что позволяет адресовать 127 уникальных USB устройств. После сброса или включения питания, адрес устройства устанавливается по умолчанию в 0 и должен быть запрограммирован ХОСТом в ходе процесса энумерации. Адрес 0 (заданный по умолчанию) зарезервирован для вновь подключаемых устройств и не может быть назначен для нормальной работы.

- **Поле адреса конечной точки (Endpoint Field)**

Поле адреса конечной точки имеет размерность 4 бита и позволяет размещать в устройстве до 16 конечных точек. Все функции должны поддерживать одну управляющую '0' конечную точку. Низкоскоростные устройства поддерживают только три канала передачи: управляющий канал, связанный с нулевой конечной точкой и два дополнительных канала (две Interrupt конечные точки). Полноскоростные устройства поддерживают максимум до 16 конечных точек.

- **Поле номера кадра (Frame Number Field)**

Поле номера кадра представляет собой 11-битное поле, которое инкрементируется ХОСТом при инициировании нового кадра. Поле номера кадра начинается заново с нуля при достижении максимального значения 0x7FF, и используется только для SOF маркеров в каждом начале кадра.

- **Поле циклического контроля по избыточности (CRC)**

Циклический контроль по избыточности (CRC) используются для защиты всех полей кроме PID в маркерах и пакетах данных. Защита Маркера и пакета данных обеспечивает 100% нахождение всех одиночных и двойных битовых ошибок. Для маркеров предусмотрено пяти-битное поле CRC, которое используется для защиты полей ADDR и ENDP пакетов IN, SETUP, OUT или поле отметки времени маркера SOF. Для пакета данных используется 16-битный полином, кодирующий всё поле данных пакета.

- **Поле конца пакета (EOP)**

Поле конца пакета представляет собой сигнал окончания пакета и устанавливается на шине путем выставления сигнала "Single Ended Zero" SE0 в течении двух битовых интервалов следующего за J состоянием, установленным на время одного битового интервала.

2.2 Типы USB пакетов.

Для USB шины позиционируется четыре типа пакетов. Пакеты-маркеры обозначает какого типа транзакция начинается на шине, пакет данных включает в себя передаваемую структуру данных, пакеты статуса используются для информирования об успешно выполненной транзакции или ошибках произошедших при передаче, и пакеты начала кадра иницируются при генерации нового кадра на шине.

- **Маркерные пакеты (Token Packets)**

Существуют три типа маркерных пакетов,

- **In** - информируют USB устройство, что ХОСТ хочет читать данные из устройства
- **Out** – информирует USB устройство, что ХОСТ хочет передавать данные в устройство
- **Setup** – используются для обозначения начала управляющего (Control Transfer) типа передачи данных

Все маркерные пакеты имеют следующий формат:

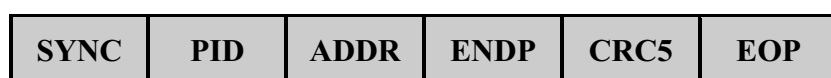


Рисунок 6. Маркерный пакет.

- **Пакеты данных (Token Packets)**

Существуют два типа пакетов данных, каждый из которых способен содержать до 1024 байтов данных.

- **DATA0**

- **DATA1**

У высокоскоростных устройств для пакетов данных определены два других PID-а: DATA2 и MDATA. Пакет данных имеет нижеследующий формат:



Рисунок 7. Пакет данных.

- Максимальная полезная загрузка для низкоскоростных устройств составляет 8 байт.

- Максимальная полезная загрузка для полноскоростных устройств составляет 1023 байта.

- Максимальная полезная загрузка для высокоскоростных устройств составляет 1024 байт.

- Данные всегда посылаются целым числом байт. Данные CRC16 вычисляются только для поля данных в пакете и не включают PID, который имеет собственное поле проверки.

- **Пакеты подтверждения (Handshake Packets)**

Существуют три типа пакетов подтверждения, структура которых имеет только PID поле:

- **АСК** - подтверждение того, что пакет был успешно принят
- **НАК** - информирует, что устройство в данный момент не может принимать либо отправлять данные. Также используется в Interrupt транзакциях для информирования ХОСТА, что устройство не имеет новых данных для передачи. ХОСТ никогда не может выдавать ответ NAK. NAK используется в целях управления потоком данных.
- **STALL** - указывает, что устройство неспособно передавать или получать данные, и требуется вмешательство ХОСТА для снятия состояния останова. Как только конечная точка устройства остановлена, устройство должно продолжить возвращать STALL, пока условие, вызвавшее останов не будет удалено с помощью вмешательства ХОСТА. Хосту запрещено возвращать STALL.

Пакет подтверждения имеет нижеследующий формат:



Рисунок 8. Пакет подтверждения.

- **Пакеты начала кадра (Start of Frame Packets)**

SOF пакет состоит из 11-ти битного номера кадра и генерируется ХОСТом каждую $1\text{ms} \pm 500\text{ns}$ для полноскоростной шины и каждые $125\text{ }\mu\text{s} \pm 0.0625\text{ }\mu\text{s}$ для высокоскоростной шины.

Пакет начала кадра имеет нижеследующий формат:

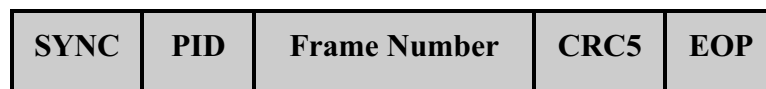


Рисунок 9. Пакет начала кадра.

2.3 USB функция.

Когда мы думаем о USB устройстве то мы представляем периферийное оборудование, но USB устройство само по себе не означает наличие только приемо-передатчика. USB устройство на самом деле представляет собой совокупность функций или законченных устройств, объединенных в одном изделии. Итак теперь мы знаем из каких составных частей состоит USB пакет. К счастью, большинство USB контроллеров, весь физический протокол нижнего уровня поддерживают самостоятельно без вмешательства пользовательской программы.

Вмешательство пользователя необходимо лишь тогда, когда на шине произошло одно из событий: прием или выдача пакета данных в/из FIFO, событие NAK или STALL, процесс эnumерации или переход в режим пониженного энергопотребления.

Большинство функций имеет буфера для приема/передачи данных. Обычно их размер составляет 8 байт. Каждый буфер закреплён за определенной конечной точкой, например за EP0 Out и EP0 In. Например ХОСТ инициирует запрос дескриптора устройства. USB контроллер функции считывает Setup пакет и определяет по адресному полю был ли он направлен именно к ней. Если пакет был направлен именно по этому адресу, то USB контроллер считывает данные в соответствующий буфер (FIFO) указанный в поле конечной точки (Endpoint Field). После этого функция генерирует пакет подтверждения, в котором указывает статус принятия пакета и устанавливает сигнал прерывания для внешнего управляющего контроллера, указывающий, что произошло событие по приему данных в RXFIFO. Все эти действия происходят без внешнего вмешательства ПО пользователя, что разгружает микроконтроллер реализующий функцию управления. ПО микроконтроллера обрабатывает прерывание – считывает из RXFIFO запрос дескриптора и отвечает на него выдачей структуры соответствующей запрошенному дескриптору.

2.4 Конечные точки.

Конечные точки функции могут принимать или передавать данные. Конечные точки USB функции являются самым последним звеном в коммуникационном канале. Например, в программном слое, ваш драйвер устройства посылает пакет данных в конечную точку EP1 функции. Данные будут последовательно поступать в функцию и складываться USB контроллером в буфер RXFIFO1. Как только у управляющего микроконтроллера появится свободное время он может считать из RXFIFO1 полученные данные. Если он хочет ответить пакетом данных, USB контроллер не может просто выставить их шину, т.к. мастером и инициатором запросов на шине является ХОСТ. Следовательно, управляющий микроконтроллер записывает пакет данных в IN конечную точку, например в EP1. Данные будут находится в буфере первой конечной точки TXFIFO1 до тех пор, пока ХОСТ не выставит на шину IN транзакцию к этой конечной точке. Конечные точки - это интерфейс связи между аппаратными средствами и микропрограммой работающей на функциональном устройстве. Все USB устройства должны поддерживать нулевую конечную точку (EP0). Она получает все управляющие и статус запросы во время процесса эnumерации и всего цикла работы, пока устройство присоединено к USB шине.

2.4 Каналы.

Пока устройство посылает и принимает данные в конечные точки, программное обеспечение пользователя передает данные через каналы. Канал является логическим соединением между ХОСТом и конечной точкой.

Каналы также имеют набор параметров, закреплённых за ними, например: ширина полосы выделенная каналу для приема/передачи данных; тип передачи данных (Control, Bulk, Iso, Interrupt); направление передачи и размер передаваемого блока данных. Например, для всех USB устройств, каналом по умолчанию (Default pipe) является двунаправленный канал с IN и OUT нулевой конечной точкой и управляющим типом передачи данных.

В спецификации USB шины определено 2 типа каналов:

- **Потоковые каналы (Stream Pipes)** – не имеют структуры определенной USB стандартом, это значит, что вы можете посылать по каналу данные любого типа. Данные по каналу передаются последовательно и имеют определенное направление: только в устройство или только из устройства. Потоковые каналы поддерживают bulk, isochronous и interrupt типы передачи данных. Потоковые каналы могут управляться либо ХОСТом, либо USB устройством.
- **Каналы сообщений (Message Pipes)** – имеют строго определенную USB стандартом структуру. Каналы сообщений всегда контролирует ХОСТ, и появляются они при генерации ХОСТом запроса в устройство. Данные в канале передаются по направлению указанному в поле запроса. Каналы сообщений поддерживают изменение направления передачи данных, но используют только управляющий тип передачи данных.