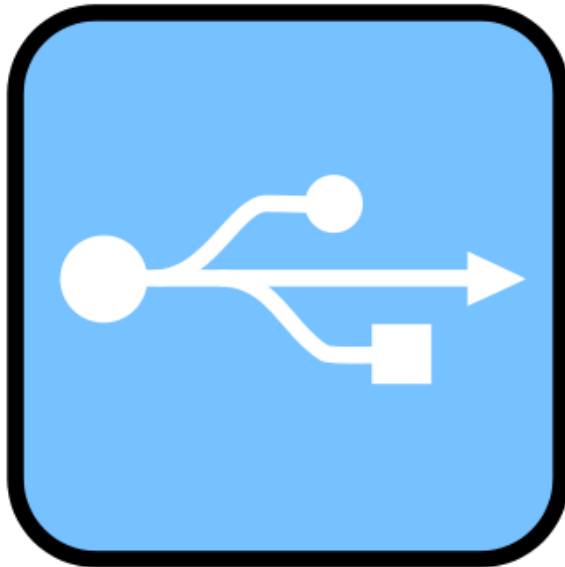


USB (*ю-эс-би*, *англ. Universal Serial Bus* — «универсальная последовательная шина») — последовательный интерфейс передачи данных для среднескоростных и низкоскоростных периферийных устройств в вычислительной технике. Символом USB являются четыре геометрические фигуры: большой круг, малый круг, треугольник и квадрат, расположенные на концах древовидной блок-схемы.



Последовательные и параллельные порты не могут обеспечить подключение к компьютеру всех необходимых устройств ввода/вывода. В 1993 году представители 7 крупнейших компьютерных компаний (Intel, IBM, DEC, Microsoft, Compaq, NEC, Northern Telecom) собрались вместе, чтобы разработать шину, оптимальную для низкоскоростных устройств. Результатом их работы стало шина USB. Назовем требования, изначально составляющая основу проекта:

- Пользователи не должны устанавливать переключатели и перемычки на платах и устройствах.
- Пользователи не должны снимать крышку компьютера, чтобы установить устройства ввода/вывода.
- Должен существовать только один тип кабеля для подсоединения всех устройств.
- Устройство ввода/вывода должны получать питание через кабель.
- Должна быть возможность для подключения к одному компьютеру до 127 устройств
- Система должны поддерживать устройства реального времени (звуковые устройства, телефон).
- Должна быть возможность устанавливать устройства во время работы компьютера.
- Должна отсутствовать необходимость перезагрузки компьютера, после установки устройства.
- Производство новой шины и устройств ввода-вывода для нее не должны требовать больших затрат.

Шина USB удовлетворяет всем этим условиям.

Другими словами, USB стал «общим знаменателем» под тремя не связанными друг с другом стремлениями разных компаний:

- Расширение функциональности компьютера. На тот момент для подключения внешних периферийных устройств к персональному компьютеру использовалось несколько «традиционных» (англ. legacy) интерфейсов (*PS/2*, *последовательный порт*, *параллельный порт*, *порт для подключения джойстика*, *SCSI*), и с появлением новых внешних устройств разрабатывали и новый разъём. Предполагалось, что USB заменит их все и заодно подхлестнёт разработку нетрадиционных устройств.

- Подключить к компьютеру *мобильный телефон*. В то время мобильные сети переходили на цифровую передачу голоса, и ни один из имеющихся интерфейсов не годился для передачи с телефона на компьютер как речи, так и данных.

- Простота для пользователя. Старые интерфейсы (например, последовательный (COM) и параллельный (LPT) порты) были крайне просты для разработчика, но не давали настоящего «*подключи и работай*». Требовались новые механизмы взаимодействия компьютера с низко- и среднескоростными *внешними устройствами* — возможно, более сложные для конструкторов, но надёжные, дружелюбные и пригодные к «горячему» подключению.

Спецификация	Скорость	Стандарт USB	Год появления
Low-Speed	до 1,5 Мбит/с	USB 1.0	1996
Full-Speed	до 12 Мбит/с	USB 1.1	1998
High-speed	до 480 Мбит/с	USB 2.0	2000
SuperSpeed	до 5 Гбит/с	USB 3.0 / USB 3.1 Gen 1 / USB 3.2 Gen 1	2008
SuperSpeed+ 10Gbps	до 10 Гбит/с	USB 3.1 Gen 2 / USB 3.2 Gen 2	2013
SuperSpeed++ 20Gbps	до 20 Гбит/с	USB 3.2 Gen 2x2	2017
	До 40 Гбит/с	USB4	2019

USB 1.0

Спецификация выпущена 15 января 1996 года.

Технические характеристики:

- два режима работы:
 - режим с низкой пропускной способностью (Low-Speed) — 1,5 Мбит/с;
 - режим с высокой пропускной способностью (Full-Speed) — 12 Мбит/с;
- максимальная длина кабеля (без экрана) для режима Low-Speed — 3 м;
- максимальная длина кабеля (в экране) для режима Full-Speed — 5 м;
- максимальное количество подключённых устройств (включая размножители) — 127;
- возможно подключение «разноскоростных» периферийных устройств к одному контроллеру USB;
 - напряжение питания для периферийных устройств — 5 В;
 - максимальный ток, потребляемый периферийным устройством, — 500 мА.

Допустимые расстояния:

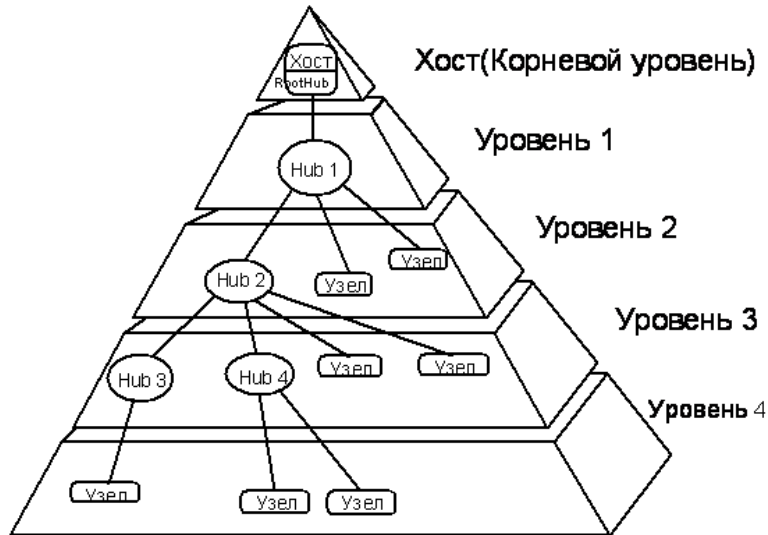
RS232 – десятки метров;

RS485 – 1200 м;

USB – 5 м.

Топология шины

В составе чипсета южного моста одной из микросхем является программно управляемый USB-контроллер. Частью этого контроллера является корневой концентратор (хаб). Этот хаб содержит разъемы для кабелей, которые могут подсоединяться к устройствам ввода-вывода (узел) или к дополнительным хабам, чтобы увеличить количество разъемов.



Таким образом, топология шины USB представляет собой дерево с корнем в центральном хабе, который находится внутри компьютера.

Кабель состоит из четырех проводов: два для передачи данных, питания (+5 В) и общего провода. В стандарте 3.0 (2008 г.) добавилось еще 4 линии связи и один контакт сигнальной земли.

Когда подсоединяется новое устройство ввода-вывода, корневой хаб обнаруживает это и сообщает операционной системе. Операционная система запрашивает новое устройство, выясняя какая пропускная способность шины для него требуется. Если пропускной системы достаточно, то операционная система приписывает ему уникальный адрес (1-127), загружает этот адрес и другую информацию в конфигурационные регистры устройства. Таким образом, новые устройства подсоединяются на лету.

Многие устройства снабжены встроенными сетевыми концентраторами. Например, монитор может содержать два хаба для правой и левой колонок.

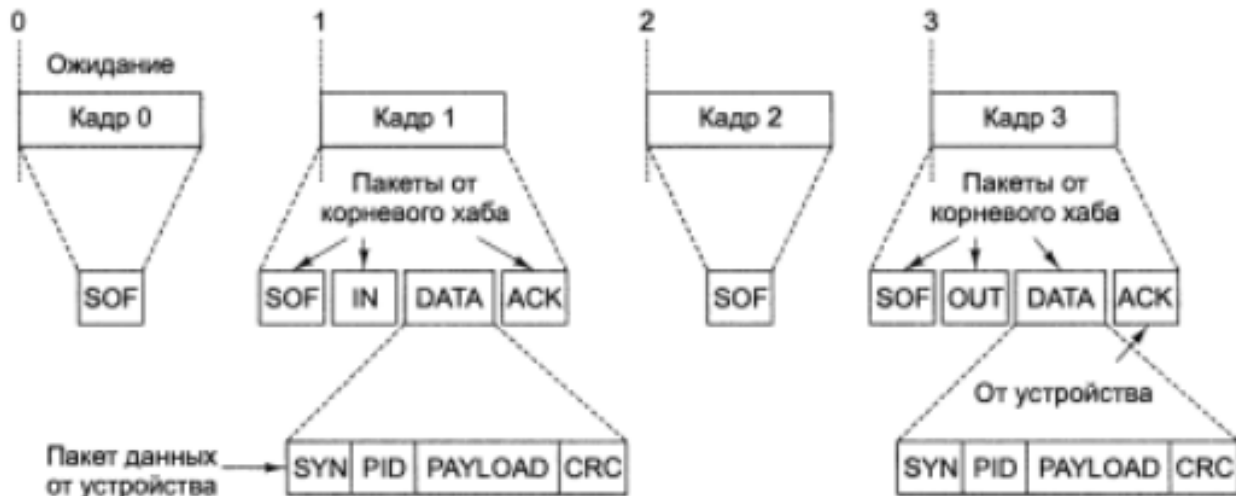
Протокол шины

Шина USB представляет собой ряд каналов между корневым хабом и устройствами ввода-вывода. Каждое устройства может разбить свой канал на 16 подканалов для различных типов данных. В каждом канале/подканале данные перемещаются от центрального концентратора к устройству и обратно. Между двумя устройствами ввода-вывода обмена информацией не происходит.

С частотой 1 кГц (в современных 8кГц) центральный хаб передает новый кадр, чтобы синхронизировать все устройства во времени. Кадр состоит из пакетов, первый из которых передается от хаба к устройству. Следующие кадры могут передаваться и от хаба к устройству и обратно.

На рисунке показаны четыре последовательных кадра.

Время, мс →



В кадрах 0 и 2 не происходит никаких действий, поэтому в них содержится только пакет SOF (Start of Frame – начало кадра). Этот пакет всегда посылается всем устройствам. Кадр 1 – запрос устройству (например, сканеру), чтобы оно передало данные. Кадр 3 состоит из данных, передаваемых устройству (например, принтеру).

Шина USB поддерживает 4 типа кадров: кадры управления, изохронные кадры, кадры передачи больших массивов данных и кадры прерывания.

Кадры управления используются для конфигурирования устройства, передачи устройствам команд и запросов об их состоянии.

Изохронные кадры предназначены для устройств реального времени, которые должны передавать и принимать данные через равные промежутки времени (микрофоны, телефоны, акустические системы). Если при передаче данных произошла ошибка, то изохронное устройство повторную передачу данных не производит.

Кадры передачи больших массивов данных используются для передач большого объема данных устройству или от устройства (например, принтеру), без требований работы в реальном времени.

Кадры прерывания нужны, так как USB не поддерживает прерываний. Например, клавиатура может опрашиваться каждые 50 мс, передавая все собранные нажатия и отпускания клавиш за этот период.

Типы пакетов

Кадр состоит из одного или нескольких пакетов. Пакеты могут посылаются в обоих направлениях. Существуют четыре типа пакетов: маркеры, пакеты данных, пакеты квитирования и специальные пакеты.

Форматы пакетов приведены на рисунке.

Поле	Sync	PID	Check	Addr	EndP	CRC	EOP
Длина, бит	8	4	4	7	4	5	2

а

Sync	PID	Check	Data	CRC	EOP
8	4	4	8×n	16	2

б

Sync	PID	Check	EOP
8	4	4	2

в

Форматы пакетов USB: а — маркер; б — пакет данных;
в — пакет квитирования

Пакет начинается с синхропоследовательности Sync и завершается признаком конца – EOP. Тип пакета определяется полем PID (указатель типа пакета), Check – его инверсия для контроля. Addr – адрес устройства, EndP – номер конечной точки, CRC – контрольный циклический код.

Маркеры передаются от концентратора к устройству и предназначены для управления системой. Пакеты SOF (Start Of Frame – начало кадра), IN (запрос на передачу данных от устройства), OUT (запрос на передачу данных устройству) – маркеры.

Поле	Sync	PID	Check	Addr	EndP	CRC	EOP
Длина, бит	8	4	4	7	4	5	2

В маркерах вслед за полем синхронизации для каждого пакета передается 4-битный идентификатор PID младшим битом вперед. Биты поля PID задают тип пакета (формат и способ обнаружения ошибок соответствующего пакета), а биты Check являются инверсными значениями PID и служат в качестве поля проверки правильности передачи поля PID.

Для выбора устройства и конечной точки используется 7-битный адрес устройства Addr и 4-битный номер EndP. Поле адреса предназначено для ввода (вывода) данных и установочных опознавателей. При сбросе или отключении питания адрес устройства принимает значение 0 и затем программируется хостом. Низкоскоростные устройства содержат до двух точек, а высокоскоростные – до 16 конечных точек. Поле адреса и номера конечной точки защищены 5-битовым контрольным циклическим кодом CRC.

Пакет данных состоит из полей PID, Check, поля данных (0-1023 байтов) и 16-битного поля CRC. Существует два пакета данных (Данные (0) и Данные (1)) с различными идентификаторами, необходимые для поддержания соответствующей синхронизации. Данные в пакете представлены в виде последовательности байтов.

Поле	Sync	PID	Check	Data	CRC	EOP
Длина, бит	8	4	4	8×n	16	2

Пакет квитирования предназначен для проверки успешности передачи данных. Различают три типа этого пакета: АСК (подтверждение) – пакет данных получен без ошибок и пакет PID верен (пакет применяется при передаче данных); NAK (неподтверждение) – пакет, показывающий на невозможность устройством принять данные от хоста (временный отказ) или устройство не имеет данных для передачи хосту (кроме того, пакет используется для сообщения о временной паузе в передаче или приеме данных устройством); STALL – ответный пакет, говорящий о постоянном отказе и необходимости вмешательства программы хоста.

Поле	Sync	PID	Check	EOP
Длина, бит	8	4	4	2

Контроль и обработка ошибок

Все принимаемые пакеты проверяются на наличие ошибок, что позволяют принятые форматы пакета и некоторые соглашения:

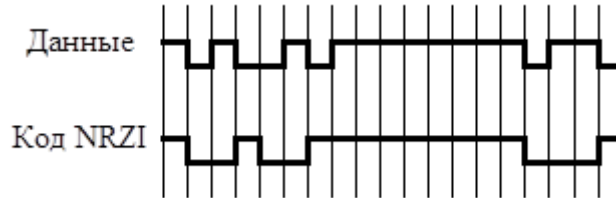
- пакет начинается с синхронизирующей последовательности, за которой следует его идентификатор PID (Packet Identifier). За идентификатором следует его инверсная копия — Check. Несовпадение двух копий считается признаком ошибки;
- тело пакета (все поля пакета, исключая PID и признак EOP) защищается CRC-кодом: 5-битным для пакетов-маркеров, 16-битным — для пакетов данных. Несовпадение CRC с ожидаемым значением считается признаком ошибки;
- пакет завершается специальным сигналом EOP; если в пакете оказывается не целое число байт, он считается ошибочным. Ложный EOP, даже на границе байта, не позволит принять пакет из-за практически неизбежной в данной ситуации ошибки по CRC-контролю;
- на физический уровень (в шину) данные пакета передаются с использованием вставки бит (bit stuffing, после шести единичных бит вставляется нолик), что предотвращает потерю битовой синхронизации при монотонном сигнале. Прием более шести единичных бит подряд считается ошибкой (на HS — признаком конца кадра).

Для обнаружения отсутствия ответа партнера на пакет каждое устройство имеет счетчик тайм-аута, который прерывает ожидание ответа по истечении некоторого времени. В USB имеется ограничение на время оборота по шине (roundtrip time): время от конца EOP сформированного пакета до получения начала ответного пакета. Для конечного устройства (и хост-контроллера) нормируется максимальная задержка ответа (response time) от конца увиденного EOP до введения им начала пакета. Для хабов нормируется задержка трансляции

пакетов, для кабелей — задержка распространения сигналов. Счетчик тайм-аута должен учитывать максимальную задержку, возможную для допустимой конфигурации шины: до 5 промежуточных хабов, до 5 метров каждый кабель. Допустимое значение тайм-аута, выражаемое в битовых интервалах (bt), зависит от скорости.

Кодирование 0 и 1

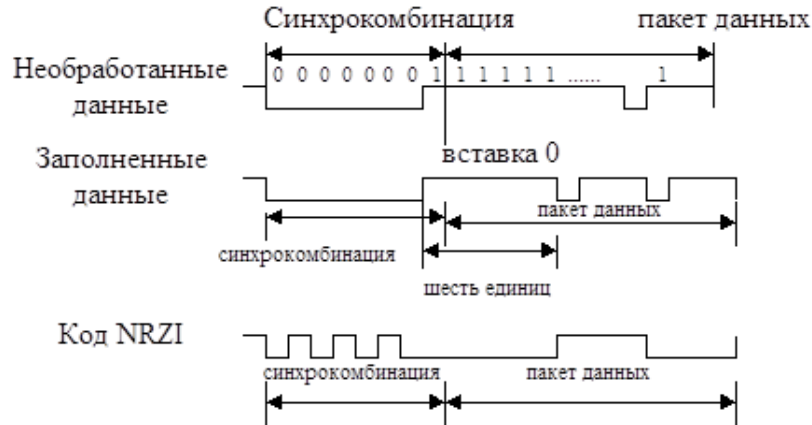
В шине USB используется метод кодирования NRZI (Not Return to Zero Invert – без возвращения к нулю с инверсией). В этом случае метод кодирования NRZI состоит в том, что если бит передаваемых данных равен 0, то происходит изменение уровня напряжения, а если равен 1, то уровень напряжения сохраняется. На рисунке показан пример кодирования данных методом NRZI.



Пример кодирования методом NRZI (LowSpeed)

Таким образом, строка нулей вызывает переключение уровней сигналов, а строка единиц образует длительные отрезки уровней без всяких переходов, что может нарушить условие синхронизации при выделении каждого бита. Поэтому при передаче данных через каждые шесть последовательных единиц вставляется нуль, чтобы гарантировать достоверное определение каждого битового интервала при приеме в наиболее худшем случае, когда передаются единичные значения битов данных. Приемник декодирует код NRZI и отбрасывает вставленные биты нулей.

На рисунке представлена временная диаграмма этапов кодирования данных.



На диаграмме вначале показаны необработанные данные, содержащие поле синхрокомбинаций и пакет данных, причем синхрокомбинация имеет 7 нулей и заканчивается единичным битом, после которого начинается пакет данных. Затем на диаграмме изображены заполненные данные, которые дополнительно содержат после шести единиц вставленный бит 0. В число шести единиц входит и последний единичный бит синхрокомбинации. После этого выполняется кодирование заполненных данных методом NRZI с учетом и поля синхрокомбинации. Правило заполнения требует, чтобы бит 0 был вставлен, даже если этот бит будет последним, перед сигналом EOP (конец пакета).