

Программисту USB-устройств

Часть 2. Стандартные требования USB

Дмитрий Чекунов (г. Ижевск)

При разработке ПО для устройств с интерфейсом USB необходимо в первую очередь обеспечить поддержку стандартных требований в соответствии со спецификацией USB. Стандартные требования USB – это базовая система команд хоста USB. Они предназначены для универсальной настройки устройств и управления ими. В этой части статьи подробно рассмотрены способ доставки требования устройству, логика анализа и обработки требования и форматы всех стандартных требований. Помимо стандартных требований, устройства могут поддерживать дополнительные требования.

Подробности контрольной транзакции

Передача требований выполняется с помощью контрольных транзакций. Одна транзакция состоит из двух или трёх фаз передачи данных. Типичный пример передачи требования с целью получения дополнительных данных от устройства представлен на рисунке 1.

Первая фаза называется Setup, она показана в верхней части рисунка. Во время этой фазы хост передаёт пакет запроса Setup точке 0 устройства ADDR и пакет данных, содержащий требование. Последний всегда имеет маркер DATA0 и размер 8 байт. Устройство подтверждает приём данных маркером ACK и начинает декодировать и выполнять полученное требование. Хочется обратить внимание на то, что в фазе Setup маркером подтверждения всегда является ACK, появление прочих маркеров запрещено протоколом.

Обработка требования устройством занимает неопределённое время,

поэтому в следующей фазе реализован механизм ожидания готовности устройства. В случае если требуется передача дополнительных данных, в контрольной транзакции начинается фаза данных, показанная в средней части рис. 1. В противном случае сразу начинается фаза статуса, описанная ниже.

В фазе данных хост передаёт пакет запроса IN, а устройство до тех пор, пока не подготовлены данные, отвечает маркером подтверждения NAK. Как можно заметить, обмен данными происходит по типу BULK. При готовности устройство передаёт пакеты данных с маркерами DATA1 (всегда первый в этой фазе) и DATA0, чередуя нумерацию чётных и нечётных пакетов. Хост подтверждает приём маркером ACK. После получения требуемого объёма информации фаза данных заканчивается и начинается фаза статуса, показанная в нижней части рис. 1.

Фаза статуса, в отличие от фазы данных, является обязательной и

предназначена для ожидания готовности устройства. На рис. 1 показано, как хост посылает запрос OUT и пакет данных размером 0 байт с целью получить маркер подтверждения. NAK говорит о том, что устройство ещё не закончило выполнять требование и хост должен повторить запрос позднее. Подтверждение ACK обозначает успешное завершение выполнения требования.

Направление запроса в фазе статуса всегда противоположно направлению запроса в фазе данных, а при отсутствии последней запрос имеет направление IN.

В случае если устройство получило не поддерживаемое требование или требование с нарушенным форматом, в фазе данных или статуса необходимо вернуть маркер STALL – признак ошибки.

СТРУКТУРА ПАКЕТА SETUP

Рассмотрим подробнее пакет данных Setup, передаваемый в одноимённой фазе. Как уже было сказано ранее, он содержит описание требования в определённом формате.

В пакете имеется пять информационных полей, распределение и назначение которых показаны в таблице 1.

Поле bmRequestType позволяет устройству определить тип требования. Для стандартных требований биты D6, D5 имеют значение 00b. Если программист добавляет новое требова-

ние, присущее только одному устройству, то биты должны иметь значение 10b, что является признаком требования производителя (продавца). Если добавляемое требование будет поддерживать класс устройств, то имеет смысл описать новое требование как присущее классу, то есть биты D6, D5 имеют значение 01b.

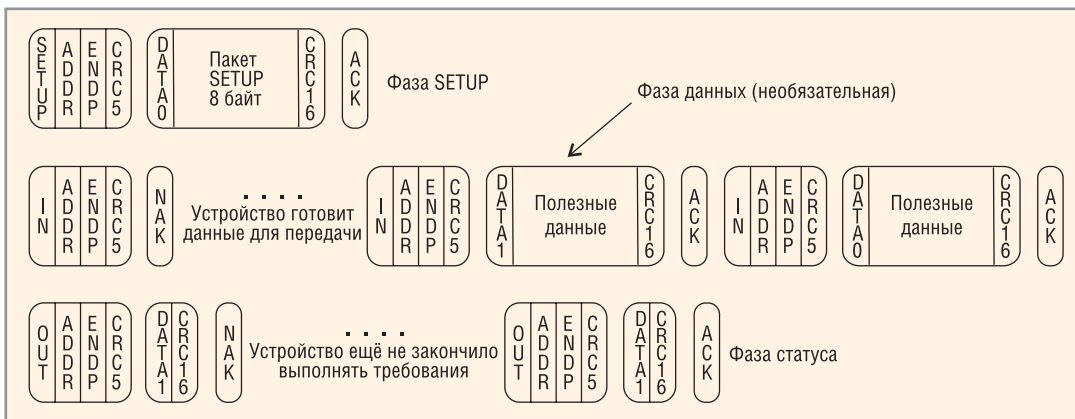


Рис. 1. Фазы контрольной транзакции

Для стандартных требований допустимые значения поля bRequest представлены в таблице 2. Для дополнительных требований значение данного поля определяется программистом.

Назначение полей wValue и wIndex изменяется в зависимости от требования.

Поле wLength показывает, какое количество байт должно быть передано в фазе данных. Когда бит D7 поля bmRequestType установлен в 1, хост ожидает от устройства данные объёмом wLength. При D7, равном 0, хост передаёт указанный объём данных устройству.

При разработке собственных требований программист должен строго соблюдать структуру поля bmRequestType и правильно использовать поле wLength, ограничение на назначение прочих полей отсутствует, перед передачей хост их не проверяет.

СОСТОЯНИЯ УСТРОЙСТВА НА ШИНЕ

В процессе подключения к шине устройство до обретения полноценного работоспособного состояния проходит через несколько промежуточных состояний. На рис. 2 показана диаграмма всех состояний. Стрелки обозначают события, переводящие устройство в следующее состояние, для обозначения состояния использованы круги.

При подключении к порту хаба устройство принимает своё первое состояние – подключённое. Если хаб не сконфигурирован, то на его портах отсутствует внутреннее напряжение питания шины. В некоторый момент хост выполняет конфигурирование хаба, что приводит к появлению питания на портах и переходу подключённого устройства в запитанное состояние. Хаб обнаруживает новое устройство и устанавливает соответствующий признак. Хост периодически контролирует статус хаба и при появлении признака наличия нового устройства разрешает соответствующий порт хаба. В этот момент для устройства формируется сигнал Reset и производится аппаратная идентификация его рабочей скорости (low, full, high). Устройство переходит в исходное состояние и принимает адрес 0 по умолчанию. Хост командой SET_ADDRESS назначает устройству

уникальный адрес на шине, и оно переходит в состояние «адресованное».

Далее хост считывает описание устройства и конфигурирует его командой SET_CONFIGURATION. После этого устройство переходит в сконфигурированное состояние и становится способным выполнять заложенные в него функции.

Передача требований устройству допустима только в трёх последних состояниях, на рисунке эти состояния выделены жирными кружками. В каждом состоянии имеются некоторые ограничения на выполнение требований, что будет рассмотрено ниже.

При появлении сигнала Reset устройство переходит в исходное со-

стояние, а в случае прерывания питания на шине USB возвращается в запитанное состояние.

Если хост не обращается к устройству в течение некоторого времени, то оно может перейти в состояние «Спящее». Выход из этого состояния происходит при первом же обращении хоста к устройству или, если разрешено хостом, при возникновении внешнего воздействия на устройство, о чём оно может сообщить хосту.

ЛОГИКА АНАЛИЗА ТРЕБОВАНИЯ

Логика анализа требования показана на рисунке 3. При получении устройством пакета Setup необходимо проанализировать поле bmRequestType, чтобы

Таблица 1. Структура пакета SETUP

Номер поля	Название поля	Размер, байт	Назначение
1	bmRequestType	1	Битовое поле, характеризующее тип требования. Распределение бит: D7 – направление передачи данных в фазе данных (0 – OUT, 1 – IN); D6, D5 – тип требования: 0 – стандартное 1 – класса 2 – продавца (производителя) 3 – зарезервировано; D4...D0 – получатель данного требования: 0 – устройство 1 – интерфейс 2 – точка 3 – остальные 4...31 – зарезервировано
2	bRequest	1	Номер требования. Допустимые номера для стандартных требований представлены в таблице 2
3	wValue	2	Значение поля зависит от требования, обычно используется для передачи некоторого значения
4	wIndex	2	Значение поля зависит от требования, обычно используется для передачи индекса или смещения
5	wLength	2	Количество байт, передаваемых в фазе данных

Таблица 2. Допустимые номера стандартных требований

Значение bRequest	Название требования	Назначение
0	GET_STATUS	Получить сведения о состоянии заданного получателя
1	CLEAR_FEATURE	Очистить свойство в окружении заданного получателя
2	Зарезервировано	Не используется
3	SET_FEATURE	Установить свойство в окружении заданного получателя
4	Зарезервировано	Не используется
5	SET_ADDRESS	Установить новый адрес устройства
6	GET_DESCRIPTOR	Получить описание заданного свойства
7	SET_DESCRIPTOR	Установить новое описание заданного свойства
8	GET_CONFIGURATION	Получить номер активной конфигурации
9	SET_CONFIGURATION	Установить новую конфигурацию
10	GET_INTERFACE	Получить номер альтернативной установки для заданного интерфейса
11	SET_INTERFACE	Установить новую альтернативную установку для заданного интерфейса
12	SYNCH_FRAME	Получить номер фрейма

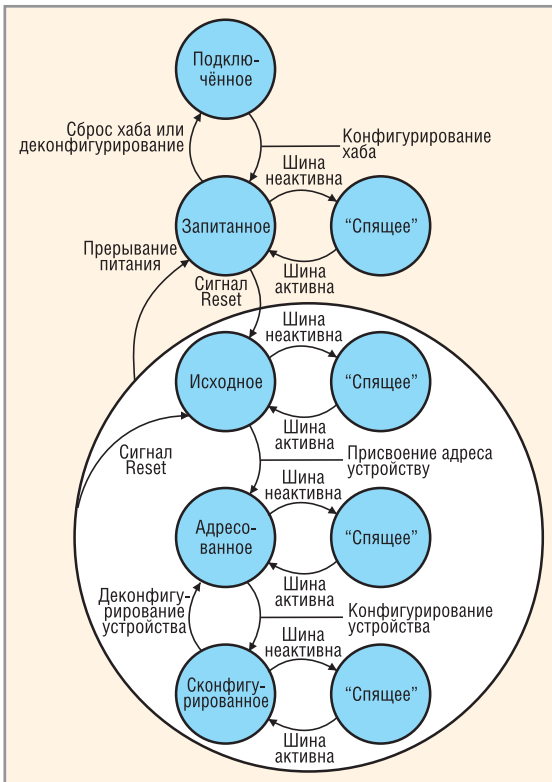


Рис. 2. Состояния устройства на шине USB

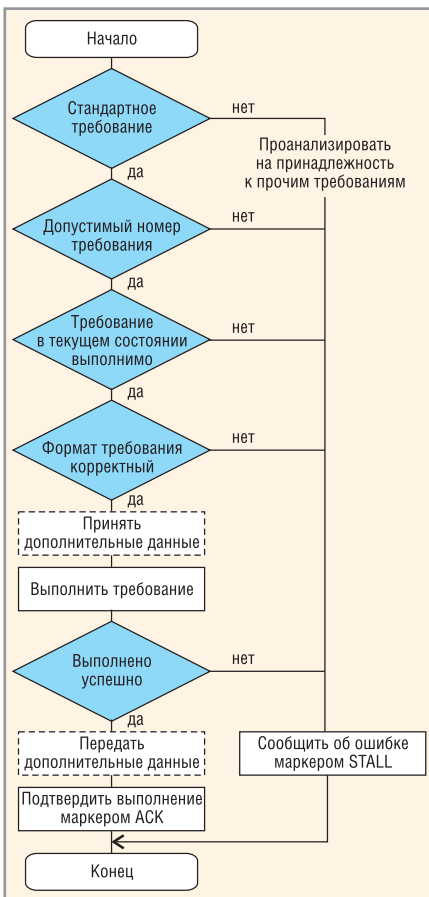


Рис. 3. Логика анализа требования

определить тип, к которому относится требование. Если требование стандартное, то номер требования в поле bRequest должен быть одним из представленных в табл. 2.

Так как устройство, подключённое к шине, может находиться в одном из нескольких описанных ранее состояний, необходимо определить, возможно ли выполнение требования в текущем состоянии.

Анализ пакета заканчивается проверкой значений полей wValue и wIndex в соответствии с форматом требования.

В случае, когда поле wLength имеет ненулевое значение, а в поле bRequestType бит D7 равен 0, будет присутствовать фаза данных OUT. Далее хост начинает передачу дополнительных данных, а устройство должно обеспечить их приём до начала выполнения требования.

Когда все исходные данные подготовлены, устройство выполняет требование.

Если в процессе выполнения возникает ошибка, то сообщить об этом хосту позволяет маркер подтверждения STALL. При успешном выполнении требования устройство сообщает о готовности маркером ACK в фазе статуса.

Если в требовании поле wLength имеет ненулевое значение, а в поле bmRequestType бит D7 равен 1, то контрольная транзакция включает фазу данных с направлением IN. Устройство должно передать некоторый объём информации хосту, как показано на рис. 1. После завершения фазы данных начинается фаза статуса, в которой устройство подтверждает завершение выполнения требования маркером ACK.

ФОРМАТЫ СТАНДАРТНЫХ ТРЕБОВАНИЙ

Перед подробным рассмотрением требований хочу напомнить, что в случае, когда требование не поддерживается в текущем состоянии устройства или получено с нарушением формата, устройство должно сообщить об ошибке. Нарушением формата требования считаются недопустимые значения и значения, отличающиеся от явно указанных в таблицах стандартных требований.

GET_STATUS (Получить статус)

Требование возвращает сведения о состоянии заданного получателя. По-

лучателем является устройство, интерфейс или точка. Поддержка требования в различных состояниях устройства:

- исходное – требование не поддерживается;
- адресованное – требование допустимо, если адресатами являются устройство, интерфейс 0 или точка 0;
- сконфигурированное – требование допустимо.

В таблице 3 показан формат требования, в котором получателем является устройство. Размер пакета возвращаемых данных составляет 2 байта, из них биты D15...D2 зарезервированы для будущего использования и сброшены в 0. Биты D0 и D1 содержат полезную информацию о статусе устройства. Бит D0 называется Self Powered и показывает текущий источник питания устройства. Признаком питания от внутренней шины USB является равенство этого бита 0. Если бит D0 установлен в 1, то устройство имеет собственный источник питания. Хост на выбор устройством источника питания повлиять не может. Бит D1 называется Remote Wakeup и характеризует возможность устройства сообщить хосту о выходе из состояния «Спящее» (рис. 2). Если бит D1 установлен в 1, то устройству разрешена данная операция. Запретить и разрешить данную функцию устройства хост может соответственно командами CLEAR_FEATURE и SET_FEATURE. Когда устройство переходит в исходное состояние, то значения битов D0, D1 сбрасываются в 0.

В табл. 3 показан формат требования, адресованного интерфейсу. Номер интерфейса задаётся в поле wIndex и должен иметь допустимое значение. Данные, возвращаемые хосту, не несут полезной информации, все биты сброшены в 0 и зарезервированы для будущего использования.

Формат требования, адресованного точке, показан в табл. 3. Номер доступной точки задаётся в поле wIndex. Хотелось бы напомнить, что точка 0 доступна всегда. В пакете возвращаемых данных все биты, за исключением D0, зарезервированы для будущего использования и сброшены в 0. Бит D0 называется Halt и характеризует способность обмена данными для точек с типом передачи bulk и interrupt. Когда бит имеет значение 1, то на все требования точка

отвечает маркером подтверждения STALL, что является признаком её не-работоспособности и требует вмешательства хоста. Если бит D0 равен 0, то точка работоспособна. Хост может манипулировать работоспособностью точки командами CLEAR_FEATURE, SET_FEATURE.

**CLEAR_FEATURE
(Очистить окружение)**

Требование позволяет очистить некоторое свойство в окружении заданного получателя. Получателем является устройство, интерфейс или точка. Поддержка требования в различных состояниях устройства:

- исходное – требование не поддерживается;

- адресованное – требование допустимо, если адресатами являются устройство, интерфейс 0 или точка 0;
- сконфигурированное – требование допустимо.

В табл. 3 показан формат требования, в котором получателем является устройство. Значение в поле wValue, равное 1, говорит о необходимости сбросить бит устройства Remote Wakeup в 0, что запрещает в дальнейшем устройству информировать хост о выходе из состояния «Спящее».

В табл. 3 показан формат требования, адресованного интерфейсу. Номер интерфейса задаётся в поле wIndex и должен иметь допустимое значение.

Формат требования, адресованного точке, показан в табл. 3. Номер доступной точки задаётся в поле wIndex. Значение в поле wValue, равное 0, говорит о необходимости сбросить бит Halt соответствующей точки. Данная операция вызывает выполнение некоторых действий устройством, после которых точка возвращается в работоспособное состояние. Для точек с типом передачи bulk и interrupt и направлением IN необходимо установить маркер DATA0 для следующего пакета данных.

**SET_FEATURE
(Установить окружение)**

Требование позволяет установить некоторое свойство в окружении за-

Таблица 3. Формат стандартных требований

Требование	Поле bmRequestType	Поле bRequest	Поле wValue	Поле wIndex	Поле wLength	Фаза данных
GET_STATUS для устройства	1000 0000b	0	0	0	2	Сведения о состоянии устройства
GET_STATUS для интерфейса	1000 0001b	0	0	Номер интерфейса	2	Сведения о состоянии интерфейса
GET_STATUS для точки	1000 0010b	0	0	Номер точки	2	Сведения о состоянии точки
CLEAR_FEATURE для устройства	0000 0000b	1	1	0	0	Отсутствует
CLEAR_FEATURE для интерфейса	0000 0001b	1	0	Номер интерфейса	0	Отсутствует
CLEAR_FEATURE для точки	0000 0010b	1	0	Номер точки	0	Отсутствует
SET_FEATURE для устройства	0000 0000b	3	1	0	0	Отсутствует
SET_FEATURE для устройства	0000 0000b	3	2	Номер теста/0	0	Отсутствует
SET_FEATURE для интерфейса	0000 0001b	3	0	Номер интерфейса	0	Отсутствует
SET_FEATURE для точки	0000 0010b	3	0	Номер точки	0	Отсутствует
SET_ADDRESS	0000 0000b	5	Новый адрес устройства	0	0	Отсутствует
GET_DESCRIPTOR	1000 0000b	6	1/0	0	Предполагаемая длина описания	Описание заданного элемента
GET_DESCRIPTOR	1000 0000b	6	2/индекс элемента	0	Предполагаемая длина описания	Описание заданного элемента
GET_DESCRIPTOR	1000 0000b	6	3/индекс элемента	Идентификатор кодовой страницы	Предполагаемая длина описания	Описание заданного элемента
GET_DESCRIPTOR	1000 0000b	6	6/0	0	Предполагаемая длина описания	Описание заданного элемента
GET_DESCRIPTOR	1000 0000b	6	7/0	0	Предполагаемая длина описания	Описание заданного элемента
SET_DESCRIPTOR	0000 0000b	7	1/0	0	Длина нового описания	Описание заданного элемента
SET_DESCRIPTOR	0000 0000b	7	2/индекс элемента	0	Длина нового описания	Описание заданного элемента
SET_DESCRIPTOR	0000 0000b	7	3/индекс элемента	Идентификатор кодовой страницы	Длина нового описания	Описание заданного элемента
SET_DESCRIPTOR	0000 0000b	7	6/0	0	Длина нового описания	Описание заданного элемента
SET_DESCRIPTOR	0000 0000b	7	7/0	0	Длина нового описания	Описание заданного элемента
GET_CONFIGURATION	1000 0000b	8	0	0	1	Номер активной конфигурации
SET_CONFIGURATION	0000 0000b	9	Номер новой конфигурации	0	0	Отсутствует
GET_INTERFACE	1000 0001b	10	0	Номер интерфейса	1	Номер альтернативной установки
SET_INTERFACE	0000 0001b	11	Номер новой альтернативной установки	Номер нового интерфейса	0	Отсутствует
SYNCH_FRAME	1000 0010b	12	0	Номер точки	2	Номер фрейма

данного получателя. Получателем является устройство, интерфейс или точка. Поддержка требования в различных состояниях устройства:

- исходное – требование допустимо, если адресатом является устройство, работающее в высокоскоростном режиме, и поле wValue имеет значение 2;
- адресованное – требование допустимо, если адресатами являются устройство, интерфейс 0 или точка 0;
- сконфигурированное – требование допустимо.

В табл. 3 показан формат требования, в котором получателем является устройство. Значение в поле wValue, равное 1, говорит о необходимости установить бит устройства Remote Wakeup в 1, что разрешает в дальнейшем устройству информировать хост о выходе из состояния «Спящее». Значение в поле wValue, равное 2, допустимо только для устройств, работающих в высокоскоростном режиме. Такое требование переводит устройство в тестовый режим, номер теста задаётся в старшем байте поля wIndex.

В табл. 3 показан формат требования, адресованного интерфейсу. Номер интерфейса задаётся в поле wIndex и должен иметь допустимое значение.

Формат требования, адресованного точке, показан в табл. 3. Номер доступной точки задаётся в поле wIndex. Значение в поле wValue, равное 0, говорит о необходимости установить бит Halt соответствующей точки. Точка с типом передачи bulk или interrupt переходит в неработоспособное состояние и на все запросы хоста отвечает подтверждением STALL.

SET_ADDRESS (Установить адрес)

Данное требование используется хостом для установки нового уникального адреса устройству. Допустимый диапазон адресов находится в пределах от 1 до 127 включительно. Поддержка требования в различных состояниях устройства:

- исходное – требование допустимо. Если новый адрес имеет значение от 1 до 127 включительно, то устройство переходит в состояние «адресованное». Если новый адрес равен 0, устройство остаётся в текущем состоянии;

- адресованное – требование допустимо. Если новый адрес имеет значение 0, то устройство переходит в исходное состояние. Если новый адрес имеет допустимое значение, то устройство работает в дальнейшем с новым адресом;

- сконфигурированное – требование не поддерживается.

В табл. 3 показан формат требования. В поле wValue задаётся новый адрес устройства.

GET_DESCRIPTOR (Получить описание)

Требование предназначено для получения описания заданного типа. Получателем требования является только устройство. Поддержка требования в различных состояниях устройства:

- исходное – требование допустимо;
- адресованное – требование допустимо;
- сконфигурированное – требование допустимо.

В табл. 3 показан формат требования. В старшем байте поля wValue задаётся тип описания, которое необходимо получить. Далее перечислены значения и соответствующие им типы: 1 – устройство, 2 – конфигурация, 3 – строка, 6 – устройство при работе в другом скоростном режиме, 7 – конфигурация в другом скоростном режиме. Значения 6 и 7 применимы только для устройств, работающих в высокоскоростном режиме. В устройстве может быть несколько описаний с типом «конфигурация» и «строка», поэтому в младшем байте поля wValue задаётся индекс элемента для данного типа описания. Для прочих типов младший байт данного поля должен быть сброшен в 0. Поле wIndex для всех типов описания, за исключением строки, имеет значение 0. Для строковых описаний в данном поле задаётся идентификатор кодовой страницы. Если будет задано значение 0, то устройство должно вернуть первое доступное описание для заданного индекса. Длина описания заранее неизвестна, поэтому хост обычно считывает описание в два этапа. Сначала производится запрос описания с минимальным значением в поле wLength, после этого хост способен определить реальную длину описания и повторить запрос с корректным значением в поле wLength.

SET_DESCRIPTOR (Установить описание)

Требование предназначено для установки нового описания заданного типа. Получателем требования является только устройство. Поддержка требования в различных состояниях устройства:

- исходное – требование не поддерживается;
- адресованное – требование допустимо, если устройство его поддерживает;
- сконфигурированное – требование допустимо, если устройство его поддерживает.

В табл. 3 показан формат требования. В старшем байте поля wValue задаётся тип описания, которое необходимо установить. Далее перечислены значения и соответствующие им типы: 1 – устройство, 2 – конфигурация, 3 – строка, 6 – устройство при работе в другом скоростном режиме, 7 – конфигурация в другом скоростном режиме. Значения 6 и 7 применимы только для устройств, работающих в высокоскоростном режиме. В устройстве может быть несколько описаний с типом «конфигурация» и «строка», поэтому в младшем байте поля wValue задаётся индекс элемента для данного типа описания. Для прочих типов младший байт данного поля должен быть сброшен в 0. Поле wIndex для всех типов описания, за исключением «строки», имеет значение 0. Для строковых описаний в данном поле задаётся идентификатор кодовой страницы. Если хост выполняет добавление новых строковых описаний, то в первую очередь он должен корректно сформировать строку с индексом 0, содержащую сведения о поддерживаемых устройствам кодовых страницах.

GET_CONFIGURATION (Получить конфигурацию)

Требование позволяет получить номер активной конфигурации устройства. Поддержка требования в различных состояниях устройства:

- исходное – требование не поддерживается;
- адресованное – требование допустимо, возвращаемое значение 0;
- сконфигурированное – требование допустимо, устройство возвращает номер активной конфигурации.

В табл. 3 показан формат требования. Если возвращаемое значение

равно 0, это говорит о том, что устройство не сконфигурировано.

SET_CONFIGURATION

(Установить конфигурацию)

Требование позволяет установить новую конфигурацию устройства. Поддержка требования в различных состояниях устройства:

- исходное – требование не поддерживается;
- адресованное – требование допустимо. Если значение новой конфигурации равно 0, то устройство остаётся в текущем состоянии. Если задано допустимое значение конфигурации, то устройство переходит в состояние «skonфигурированное»;
- skonфигурированное – требование допустимо. Если значение новой конфигурации равно 0, то устройство переходит в адресованное состояние. Если задано допустимое значение конфигурации, то устройство переходит в новую конфигурацию.

В табл. 3 показан формат требования. В поле wValue задаётся номер новой конфигурации. Хост получает номера доступных конфигураций при считывании описания конфигураций командой GET_DESCRIPTOR. При выборе новой конфигурации, отличной от 0, необходимо также выполнить установку интерфейса и альтернативной установки с номерами 0 (первые доступные в конфигурации). Для всех доступных точек в альтернативной установке с типом передачи bulk и interrupt и направлением передачи IN необходимо установить маркер DATA0 для следующего пакета

данных. Бит Halt всех точек сбрасывается в 0, все точки работоспособны.

GET_INTERFACE

(Получить интерфейс)

Требование позволяет получить номер активной альтернативной установки в заданном интерфейсе текущей конфигурации устройства. Получателем требования является только интерфейс. Поддержка требования в различных состояниях устройства:

- исходное – требование не поддерживается;
- адресованное – требование не поддерживается;
- skonфигурированное – требование допустимо, устройство возвращает номер активной альтернативной установки.

В табл. 3 показан формат требования. В поле wIndex задаётся номер интерфейса, для которого необходимо получить сведения об активной альтернативной установке.

SET_INTERFACE

(Установить интерфейс)

Требование позволяет выбрать новую альтернативную установку в заданном интерфейсе текущей конфигурации. Получателем требования является только интерфейс. Поддержка требования в различных состояниях устройства:

- исходное – требование не поддерживается;
- адресованное – требование не поддерживается;
- skonфигурированное – требование допустимо.

В табл. 3 показан формат требования. В поле wValue задаётся номер но-

вой альтернативной установки, а в поле wIndex – номер нового интерфейса. Хост получает номера доступных интерфейсов и альтернативных установок при считывании описания конфигураций командой GET_DESCRIPTOR. При выборе новой альтернативной установки необходимо для всех доступных точек с типом передачи bulk и interrupt и направлением передачи IN установить маркер DATA0 для следующего пакета данных. Бит Halt всех точек сбрасывается в 0, все точки работоспособны.

SYNCH_FRAME

(Синхронизировать фрейм)

Требование используется для установки и контроля фрейма синхронизации. Получателем требования является только точка. Поддержка требования в различных состояниях устройства:

- исходное – требование не поддерживается;
- адресованное – требование не поддерживается;
- skonфигурированное – требование допустимо.

В табл. 3 показан формат требования. В поле wIndex задаётся номер доступной точки, работающей с изохронным типом передачи данных и использующей неявную синхронизацию при обмене данными.

ЛИТЕРАТУРА

1. Universal Serial Bus Specification Revision 2.0. www.usb.org.
2. EZ-USB FX2 Technical Reference Manual. www.cypress.com.

Продолжение следует.



Новости мира News of the World Новости мира

Intel позаботится о безопасности беспроводного USB-интерфейса

Разрабатываемый при поддержке компании Intel беспроводный стандарт Wireless USB обзавелся усиленной защитой. Intel представила более совершенный алгоритм шифрования Advanced Encryption Standard (AES) для сетей Wireless USB (WUSB). Он обеспечивает дополнительную безопасность данных за счёт использования 128-разрядных ключей.

Ожидается, что первые чипы с поддержкой окончательной спецификации 1.0 технологии WUSB появятся к середи-

не 2005 года. В дальнейшем Intel намерена также обеспечить совместимость WUSB со стандартом Ultra Wide Band (UWB).

NEC вновь возглавит гонку суперкомпьютеров?

Разработанный компанией NEC суперкомпьютер способен вдвое обойти по производительности своего «визави» от IBM. Месяц назад суперкомпьютер Blue Gene/L компании IBM возглавил рейтинг наиболее производительных систем, отобрав пальму первенства у NEC. Вторая не заставила себя долго ждать с ответом. Как заявляет NEC, в перспективе её новая вы-

числительная система SX-8 сможет выполнять 65 трлн. вычислений в секунду. Такой показатель обеспечит ей первое место в списке самых производительных суперкомпьютеров. NEC готовит три модификации SX-8 – две одноузловые и одну многоузловую, которая будет состоять максимум из 512 узлов, несущих 4096 процессоров. Именно последней модификации разработчики и прочат первенство. Начало поставок SX-8 было намечено на декабрь 2004 года. Цена на эти устройства начинается с \$10 780 (одноузловая версия). В течение трёх лет NEC планирует продать около 700 таких систем.

<http://itware.com.ua/>