

Универсальная последовательная шина и PIC-контроллеры с модулем USB

Виктор Ульрих

microchip@gamma.spb.ru

1. Краткий обзор USB

Универсальная последовательная шина (USB) была разработана в 1996 году для расширения архитектуры персональных компьютеров. Шина позволяет обеспечить такие основные требования, как высокая скорость передачи, одновременное подключение большого количества устройств, удобство операций, надежная связь, а также возможность горячего подключения устройств и низкая цена.

Шина USB имеет топологию многоярусной звезды, где в центре находится главная ЭВМ, в средних узлах — хабы, а в конечных точках — индивидуальные устройства (функции). USB позволяет подключать к шине до 127 функций, а дерево может быть глубиной до 6 уровней. Главная ЭВМ всегда является ведущей, и обмен данными осуществляется под ее непосредственным управлением.

Передача осуществляется в одном из двух режимов: с полным быстродействием (12 Мбит/с) и низкой скоростью (1,5 Мбит/с). Полное быстродействие позволяет передавать данные в приложениях, работающих с интенсивными звуковыми сигналами или сжатым видео, а низкоскоростной режим поддерживают приложения с небольшой интенсивностью передачи данных.

Для передачи данных определены четыре типа обмена данными (транзакции):

- Изохронные транзакции обеспечивают пересылку данных в реальном масштабе времени, когда фиксированное количество данных передается за равные интервалы времени с фиксированной скоростью. Этот тип транзакций обеспечивает своевременность доставки данных, но их точность не гарантируется, так как нет времени для повторной передачи пакетов с ошибками, которые просто игнорируются.
- Объемные транзакции обеспечивают точность данных, но не своевременность.
- Транзакции прерывания предназначены для связи с устройствами, которые имеют умеренные требования по скорости передачи данных. Главная ЭВМ периодически, через фиксированные интервалы времени, опрашивает эти устройства для того, чтобы определить, имеются ли данные для передачи. В таких устройствах появление данных имеет спонтанный характер, но передача их должна осуществляться немедленно.
- Транзакции управления используются для конфигурирования при подключении и для управления в процессе работы.

Все транзакции по USB состоят из трех пакетов, каждый из которых начинается по инициативе главной ЭВМ:

- Пакет-маркер описывает тип и направление передачи, адрес устройства USB и номер конечной точки.
- Пакет данных, который передает источник, если у него есть данные.
- Пакет взаимодействия передается после успешного приема данных или источником, если у него нет данных для передачи. При обнаружении ошибки пакет взаимодействия не передается.

Для обнаружения ошибок каждый пакет имеет контрольные поля CRC-кодов, позволяющие обнаруживать все одиночные и двойные ошибки.

Все информационные сообщения на шине сгруппированы в кадры. Каждый кадр имеет длительность 1 мс и состоит из множественных передач. Каждый кадр начинается с передачи главной ЭВМ пакета-маркера начала кадра, содержащего номер кадра. Внутри кадра транзакции могут быть повторены несколько раз.

Устройства, подключенные к USB, могут иметь собственный источник питания или питаться непосредственно от шины. Поэтому, кроме сигнальной пары, кабель USB имеет линии напряжения питания 5 В. При питании непосредственно от шины устройство может потреблять до 100 мА. Если для нормальной работы устройству требуется больший ток (до 500 мА), то оно должно запросить его у главной ЭВМ. Если главная ЭВМ определяет, что в настоящее время шина загружена и не может выделить больше тока, то устройство не может переключиться в режим большого потребления и должно остаться в конфигурации малого потребления.

Самым низким уровнем каждого устройства является функция. Устройства могут иметь в своем составе одну или несколько функций. Конечная точка подразумевается как виртуальный порт для управления функцией. Каждая конечная точка может быть источником или приемником данных.

Устройства с полным быстродействием могут иметь до 15 доступных для использования конечных точек, а с низкой скоростью — только до 6. В каждом устройстве для операций конфигурирования и общего управления должна быть реализована конечная точка с номером 0.

Ввод и вывод данных на шине осуществляется относительно главной ЭВМ, а не устройства.

USB поддерживает подключение и отключение устройств в процессе работы. Нумерация устройств на

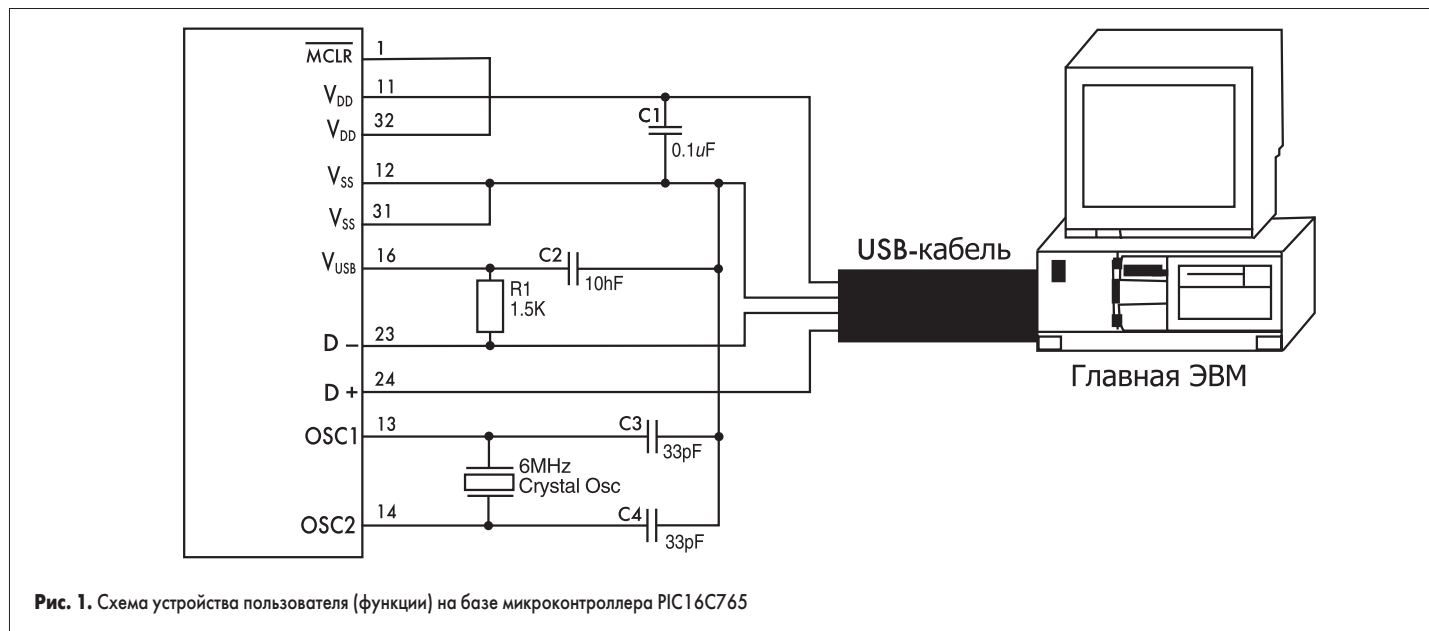


Рис. 1. Схема устройства пользователя (функции) на базе микроконтроллера PIC16C765

шине является постоянным процессом, отслеживающим динамические изменения физической топологии. Процедура нумерации осуществляется без вмешательства пользователя или прикладного программного обеспечения. Процедура нумерации осуществляется следующим образом:

- Когда устройство подключается к шине USB, главная ЭВМ определяет изменение состояния своего порта, а устройство переходит в режим «подключено».
- Главная ЭВМ формирует сигнал сброса на шине, после чего устройство переходит в состояние «включено», все его регистры устанавливаются в исходное состояние и оно отзывается на обращения по нулевому адресу.
- Главная ЭВМ сообщает устройству уникальный адрес, и оно переходит в состояние «адресовано».
- Главная ЭВМ считывает все конфигурации устройства и, исходя из считанной информации, конфигурирует все имеющиеся оконечные точки данного устройства, устройство переходит в состояние «skonфигурировано» и становится готовым для использования.

При конфигурировании устройство передает главной ЭВМ свои параметры, позволяющие идентифицировать устройство, определить оконечные точки и функции для каждой оконечной точки. Для этого существует пять основных категорий описателей.

- Описатель устройства содержит общую информацию: фирма-производитель, номер изделия, серийный номер, число поддерживаемых конфигураций.
- Описатель конфигурации содержит информацию о потребляемой мощности устройства и количестве интерфейсов, поддерживаемых в этой конфигурации. Устройство может иметь более одной конфигурации (то есть устройство с большим потреблением может также поддерживать конфигурацию с малым потреблением).
- Описатель интерфейса содержит число оконечных точек, используемых в этом интерфейсе, а также класс драйвера для поддержки устройства.

- Описатель оконечной точки детализирует фактические регистры устройства. Содержит информацию о поддерживаемых типах передачи, направлении обмена (ввод или вывод), требуемой полосе пропускания канала и об интервалах обслуживания. В устройстве может быть более одной оконечной точки, и оконечные точки могут быть разделены между различными интерфейсами.
- Строковые описатели используются для того, чтобы обеспечить пользователя специфической или специальной для приложения информацией. Они могут быть закодированы в формате Unicode. Перечисленные выше четыре описателя могут быть ссылкой или индексом к строковым описателям. Строковые описатели необязательны.

Операционные системы главной ЭВМ обычно имеют драйверы, которые группируют функции по общим признакам устройств, называемых классами. Например, класс может включать устройства хранения, звуковые, связи и пользовательского интерфейса, но не ограничиваться перечисленным. Ссылки на драйверы класса для данного устройства могут быть в двух описателях: устройства и интерфейса.

Для большинства устройств может быть подобран стандартный драйвер для данного класса, который поддерживает большинство функций или команд. Если устройство не имеет стандартного драйвера для данного класса, то оно должно поставляться вместе со специфическим файлом .inf или драйвером для его поддержки.

Краткий обзор USB не может полностью раскрыть все тонкости интерфейса, поэтому дополнительно хотелось бы отметить, что передачи с низкой скоростью разработаны для устройств, которые для связи с главной ЭВМ в прошлом использовали прерывания. В системах с USB, если эти устройства имеют данные для передачи, то не прерывают процессор непосредственно, а главная ЭВМ периодически опрашивает каждое устройство для того, чтобы определить наличие у него данных для передачи. Скорость опроса между устройством и главной ЭВМ задается при конфигури-

ровании и обеспечивает допустимое время ожидания обслуживания.

Подробную информацию по шине USB можно получить на сайте www.usb.org.

2. PIC — микроконтроллеры с модулем USB

Фирма Microchip Technology выпускает микроконтроллеры PIC16C745/765, которые имеют встроенный периферийный модуль USB, соответствующий спецификации USB v1.1. Модуль USB поддерживает передачи управления и прерывания (ввод-вывод) для низкой скорости 1,5 Мбит/с. При реализации функций могут поддерживаться 3 номера оконечной точки (0, 1, 2) при общем количестве 6 оконечных точек.

На кристалле микроконтроллера интегрирован приемопередатчик USB, включающий линейные драйверы D+/D- и обеспечивающий физический стык с линией. Также имеется стабилизатор, который обеспечивает напряжением питания 3,3 В линейные драйверы D+/D- и внешний вывод VUSB. Этот вывод предназначен для подтягивания линии D- к высокому уровню через резистор 1,5 кОм 5%, что в соответствии со стандартом USB определяет устройство как низкоскоростное. Для стабильности на выводе VUSB требуется конденсатор 10 нФ 20%. На рис. 1 представлена схема функции на базе микроконтроллера PIC16C765 с питанием от шины USB.

Управление модулем USB осуществляется при помощи 9 регистров управления и состояния, которые размещены в банке 3 памяти данных микроконтроллера.

- UIR — регистр флажков прерываний модуля USB.
- UIE — регистр управления прерываниями модуля USB.
- UEIR — регистр флажков ошибок модуля USB.
- UEIE — регистр разрешения прерываний по ошибкам модуля USB.
- USTAT — регистр состояния модуля USB.
- UCTRL — регистр управления.

- UADDR — регистр адреса USB.
- USWSTAT — регистр состояния программного обеспечения USB.
- UEPn — регистры управления оконечными точками.

Для эффективного управления связью с оконечной точкой модуль USB имеет двухпортовую память емкостью 64 байта, которая расположена в банке 3 памяти данных микроконтроллера, где размещены таблицы дескрипторов буферов (BDT) и буферы данных. Для каждой оконечной точки существует дескриптор буфера (BD), который состоит из 4 регистров и указывает на буфер данных.

- BDndST — регистр состояния буфера, доступный ядру процессора только для записи.
- BDndST — регистр состояния буфера, доступный ядру процессора только для чтения.
- BDndBC — счетчик байтов буфера.
- BDndAL — начальный адрес буфера.

Где n — симплексное соединение номера оконечной точки и направления в регистре USTAT.

Поскольку буферы доступны и ядру процессора и модулю USB, то для того чтобы различить, кто может модифицировать BD и данные в буфере, используется простой механизм семафора — бит UOWN в регистре BDndST.

Если бит UOWN = 0, то доступ к BD и буферу имеет ядро процессора. Если бит UOWN = 1, то доступ к BD и буферу принадлежит модулю USB, при этом ядро процессора не должно изменять BD или данные в соответствующем буфере.

Управление BD и буфером данных в двухпортовой памяти обычно осуществляется следующим образом:

- Ядро процессора проверяет бит UOWN в регистре BDndST. Если он равен 0, то устанавливает адрес начала буфера в регистре BDndAL, в случае необходимости заполняет буфер и устанавливает необходимое значение в BDndST с UOWN = 1.
- После завершения транзакции с главной ЭВМ модуль USB выполняет следующее:
 - получает адрес буфера;
 - читает или записывает буфер;
 - модифицирует регистр USTAT;
 - модифицирует дескриптор буфера значением ID пакета (PID);
 - модифицирует счетчик байтов;
 - сбрасывает бит UOWN.
- По прерыванию ядро процессора читает регистр USTAT, определяет номер оконечной

точки, читает соответствующий BD, где проверяет биты UOWN, PID и значение счетчика байтов.

3. Программное обеспечение USB

Для помощи разработчикам в реализации проектов фирма Microchip Technology предоставляет набор библиотечных функций, поддерживающих USB-интерфейс в соответствии с главой 9 спецификации USB. Эти библиотеки позволяют освободить основное программное обеспечение от необходимости обрабатывать сложный протокол USB. Выполнение большинства функций USB обеспечивается просто вызовом интерфейсных функций Put/Get (передать/принять), обработка которых осуществляется на втором плане программой обслуживания прерывания модуля USB. Библиотеки также обеспечивают приложения операциями при нумерации и конфигурировании. Однако установка реальных описателей функции должна выполняться пользователем.

Последние версии библиотек подпрограмм можно получить на сайте Microchip Technology (www.microchip.com)

Библиотечные подпрограммы разработаны для использования с компоновщиком. Нет необходимости создавать дополнительные включаемые файлы. Подпрограммы предлагаются упакованными в несколько отдельных файлов:

- USB_CH9.ASM — обработка всех команд в соответствии с главой 9 спецификации USB.
- USB_INTF.ASM — содержит функции интерфейса PutUSB, GetUSB.
- USBMACRO.INC — содержит макрокоманды.
- USB_DEFS.INC — содержит определения.
- USB_INT.ASM — типовая программа обработки прерывания.
- I6C765.LKR — сценарий для компоновщика (для MPLAB).

Основные функции и подпрограммы

InitUSB должна вызываться основной программой после включения питания. Она устанавливает таблицу дескрипторов буферов, приводит модуль в состояние «включено» готовит устройство к процессу нумерации. При этом разрешается прерывание USB только по сбросу USB, что предотвращает ответы на какие-либо операции на шине, пока модуль не сброшен. Прерывание RST_USB переводит модуль в заданное по умолчанию состояние, когда он отвечает на команды по нулевому ад-

ресу. Когда получена команда SET ADDRESS, то модуль переходит к состоянию «адресовано» и может отвечать на команды по новому адресу.

PutUSB посылает данные к главной ЭВМ. Указатель на блок данных передается в FSR/IRP, размер блока и номер оконечной точки передается в регистре W. Если выходной буфер доступен для оконечной точки, то блок данных копируется в буфер, переворачивается бит DATA 0/1 и устанавливается бит обладания буфером. Если буфер недоступен, то есть он был предварительно загружен, но главная ЭВМ не запросила его передачу, то в этом случае возвращается код отказа, для того чтобы основная программа могла повторить передачу этого буфера позже.

GetUSB возвращает данные, посланные из главной ЭВМ. Если имеется готовый буфер, (то есть данные были получены от главной ЭВМ), то он копируется по адресу, указанному в FSR/IRP, номер оконечной точки передается в регистре W. Если данные не получены, то возвращается код отказа. Таким образом, функции опроса буфера и копирования готовых данных объединены в одну.

ServiceUSBInt обрабатывает все прерывания, сгенерированные модулем USB. В первую очередь копируется активный буфер в общую память, что обеспечивает быстрый оборот буферов в двухпортовой памяти, а также позволяет избежать необходимости переключать банки в процессе обработки буфера.

StallUSBEP/UnstallUSBEP устанавливает или очищает бит останова BSTALL в регистре управления оконечной точки. Бит останова указывает главной ЭВМ, что требуется вмешательство пользователя и пока такого вмешательства не будет, попытки связи с оконечной точкой не будут успешными. После вмешательства пользователя, функция UnstallUSBEP очистит бит, что позволит возобновить связь. Эти функции необходимы для сообщения главной ЭВМ, что требуется вмешательство пользователя. Например, ситуация, когда в принтере отсутствует бумага.

SoftDetachUSB очищает бит DEV_ATT приблизительно на 50 мс, электрически отключая устройство от шины, а затем повторно подключает его. Это позволяет главной ЭВМ обнаружить отключение устройства и выполнить процесс нумерации заново.

CheckSleep проверяет бит UIDLE регистра UCTRL. Если бит установлен, что указывает на отсутствие в шине каких-либо действий в течение 3 мс, то переводит устройство в режим останова. Режим останова будет продолжаться до пробуждения устройства действиями на шине. Эта функция должна выполняться вне программы обработки прерывания, потому что для пробуждения необходимо прерывание, а также потому, что основная программа может быть не готова к останову, когда происходит прерывание. Основная программа должна периодически вызывать эту функцию, когда устройство может перейти в режим останова.

Перед переходом в режим останова функция разрешает прерывания, так что устройство будет пробуждено при активизации шины.

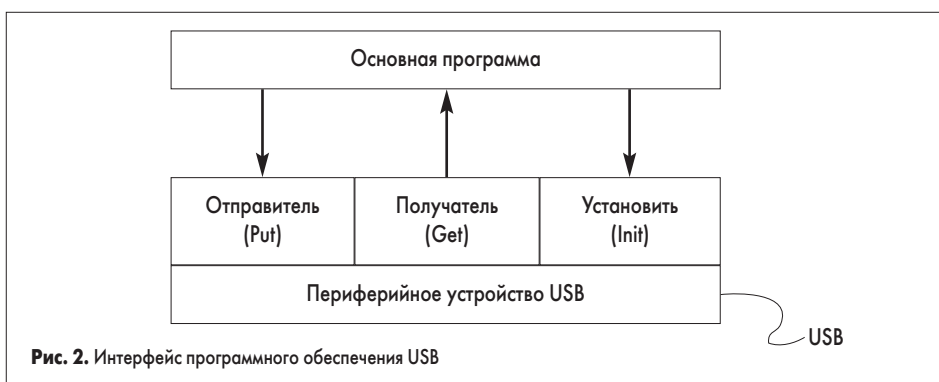


Рис. 2. Интерфейс программного обеспечения USB

После обработки прерывания программа будет продолжена с команды после вызова CheckSleep.

ConfiguredUSB (макрокоманда) непрерывно опрашивает биты состояния и ожидает, пока устройство не будет сконфигурировано главной ЭВМ. Макрокоманда должна использоваться после вызова InitUSB.

SetConfiguration — функция, которая позволяет по команде конфигурирования от главной ЭВМ передавать параметры устройства. Значения параметров устройства хранятся в USB_CURR_CONFIG программного модуля. При вызове этой функции в регистре W передается указатель на новую конфигурацию. Эта функция вызывается из программы обработки прерывания, так что значения параметров устройства должны быть сохранены до того, как прерывания будут разрешены.

Определения

Имеются также два определения (#defines), задаваемые в начале программы и определяющие опции управления при трансляции.

#define ERRORCOUNTERS — это определение включает в программу код, позволяющий считать число ошибок, которые происходят с учетом типа ошибок. Для этого требуется дополнительное пространство памяти программ и ячейки оперативной памяти для организации счетчиков.

#define FUNCTIONIDS — это определение удобно при отладке. Оно кодирует 6 старших битов регистра USWSTAT, которые показывают, какая функция выполняется.

4. Ресурсы процессора

Основная обработка USB происходит в программе обработки прерывания и, таким образом, невидима для основной программы. Однако она использует ресурсы процессора: память программ, память данных, регистры, уровни в стеке и циклы процессора.

Память программ

Программа обработки прерывания USB, включая подпрограммы интерфейса, но без описателей занимает 1 кВ. Таблицы описателей могут занимать дополнительно 256 W. Расположение этих частей не ограничено, для управления размещением каждой части сценарий компоновщика может быть отредактирован.

Память данных

За исключением общедоступной памяти данных, обслуживание прерывания USB занимает примерно 40 байт памяти данных в банке 2. Для использования основной программой остается вся универсальная память данных в банках 0 и 1 и половина банка 2.

Общедоступная память данных

РIS16C745/765 имеет 16 байт общедоступной памяти данных. Они занимают 16 последних адресов в каждом банке, при обращении по этим адресам, независимо от состояния битов выбора банка RP0, RP1 и IPR, осуществляется обращение к одним и тем же 16 байтам памяти.

Эта память особенно полезна при обработке прерываний для стека и регистров специальных функций, поскольку в момент преры-

вания неизвестно, какой банк памяти данных адресован. В устройствах, которые не имеют общедоступной памяти, стек для регистра W должен быть предусмотрен во всех банках.

Буферные регистры

РIS16C745/765 имеет двухпортовую буферную память емкостью 64 байта, 24 из которых используются для таблицы дескрипторов буферов (BDT), а остальные 40 — для буферов.

Оконечной точке 0 при вводе и выводе требуются специализированные буферы, так как транзакции установки не могут быть приостановлены. Остальные три буфера распределяются между четырьмя возможными оконечными точками, так как в соответствии со спецификацией USB-устройства с низкой скоростью могут иметь только 2 активные оконечные точки (USB 1.1 пункт 5.3.1.2).

Уровни стека

Аппаратный стек микроконтроллера имеет глубину всего 8 уровней. Поэтому вложенность подпрограмм и прерываний не может быть более 8. Процесс нумерации USB требует 6 уровней, так что самое лучшее, если основная программа будет удерживаться от выполнения подпрограмм и обработки прерываний, пока процесс нумерации не завершен. Макрокоманда ConfiguredUSB служит для этой цели, она проверяет два младших разряда USWSTAT и ожидает, пока не завершится процесс нумерации.

Выбор страниц и выбор банков

Микропрограммные библиотеки выполнены с осторожным использованием битов переключения банков и страниц. При переключении банков должен обеспечиваться вызов правильных регистров. Регистр PCLATH изменяется использованием команд PAGESEL. Он необходим, когда выполняются операции CALL или GOTO за пределы текущего сегмента памяти программ. ServiceUSBInt изменяет W, STATUS, FSR и PCLATH, следовательно, они должны быть сохранены при входе в программу обработки и восстановлены при выходе.

5. Интегрирование USB с основной программой

При написании подпрограммы обслуживания прерывания необходимо принять во внимание, что регистры W, STATUS, FSR и PCLATH могут быть разрушены при обслуживании прерывания USB, поэтому они должны быть сохранены. Библиотека предлагает скелет подпрограммы обслуживания прерывания (см. файл USB_INT.ASM), которая сохраняет и восстанавливает эти регистры, а также выполняет проверку каждого из возможных битов прерывания. Эта подпрограмма может быть включена в начало основной программы.

Интерфейс между основной программой и уровнем протокола обеспечивают три основные функции: InitUSB, PutUSB и GetUSB.

Работа USB зависит от описателей, представляющих собой параметры устройства, которые передаются главной ЭВМ и содержат информацию об устройстве и способе связи с ним.

```

;*****
; Демонстрационная программа, которая инициализирует модуль USB,
; позволяет главной ЭВМ выполнить процесс нумерации устройства,
; затем копирует буфер EP1OUT в EP1IN.
;*****
main
    call InitUSB      ; Инициализация.
    ConfiguredUSB    ; Ожидание окончания нумерации.
idleloop
    call CheckSleep  ; Останов, если шина неактивна.
    CheckEP1        ; Проверить, транзакция EP1 OUT
    bcf STATUS,IRP  ; Указатель на младшие банки
movlw buffer
    movwf FSR        ; Указать на буфер в FSR
    movlw 1          ; номер оконечной точки 1
    call GetUSB      ; Если данные готовы, они будут скопированы.
    btfss STATUS,C   ; Данные получены?
    goto idleloop   ; Нет, проверить снова.
PutBuffer
    bcf STATUS,IRP  ; Указатель на младшие банки
    movwf bufferlen ; Сохранить длину буфера
movlw buffer
    movwf FSR        ; Указать на буфер в FSR
    swapf bufferlen,W ; Старший нибл W — длина буфера
    iorlw 1          ; Младший нибл W — номер оконечной точки.
call PutUSB
    btfss STATUS,C   ; Передача успешна?
    goto PutBuffer   ; Нет, повторить попытку.
    goto idleloop   ; Да: переход к началу цикла

end

```

Программа должна быть дополнена установкой значений для команды SetConfiguration. Команды главы 9 спецификации USB вызывают SetConfiguration, когда получена соответствующая команда. Описатели и SetConfiguration находятся в файле descript.asm.

Функция InitUSB инициализирует модуль USB, таблицу дескрипторов буферов и разрешает прерывание USB, так что процесс нумерации может быть начат.

Фактический процесс нумерации происходит автоматически под управлением главной ЭВМ в программе обработки прерывания. Макрокоманда ConfiguredUSB ожидает, пока устройство не будет сконфигурировано и готово к продолжению. Время, требуемое для процесса нумерации, полностью зависит от загрузки шины и главной ЭВМ.

Для передачи данных используется функция PutUSB, посылающая данные на главную ЭВМ, и функция GetUSB для получения данных от главной ЭВМ.

Далее представлен демонстрационный пример использования функций USB. Программа сначала инициализирует модуль USB, что позволяет главной ЭВМ выполнить процесс нумерации устройства. Процесс нумерации происходит в программе обработки прерывания. Затем ожидается завершение процесса нумерации. После того как устройство готово, выполняется опрос оконечной точки EP1 OUT для того, чтобы определить, имеются ли данные. Если данные получены, то они копируются в буфер оконечной точки EP1 IN для передачи.