Для каждой конечной точки определены следующие параметры:

* Номер конечной точки.
* Направление передачи данных (IN, OUT)
* Используемый тип передачи данных (control, interrupt, bulk, iso).
* Максимальный размер пакета, который конечная точка может принимать или отправлять.
* Требуемая ширина полосы пропускания канала.
* Допустимая величина задержки обслуживания.
* Частота доступа к шине.
* Способ обработки ошибок.

В соответствии с установленным типом передачи для конечной точки к ней применимы следующие типы операций:

|  |  |
| --- | --- |
| Тип передачи конечной точки | Операции |
| Control | ControlRead, ControlWrite, NoDataControl |
| Bulk | BulkRead, BulkWrite |
| Interrupt | передача по прерыванию/приём по прерыванию |
| Isochronous | изохронная передача/изохронный приём |

Запросы хоста выполняются с помощью управляющих посылок типа control. Для выполнения запроса хост посылает устройству конфигурационный пакет (Setup Packet), который обрабатывается на Setup Stage. Если хост запрашивает что-то – выполняется операция ControlRead. Если что-то устанавливает, выполняется операция ControlWrite. Пакеты данных, которые при этом отправляются или принимаются обслуживаются, соответственно, на Data IN Stage и Data OUT Stage для нулевой конечной точки.

**Пример обработки class-specific запроса.**

USBD\_DCD\_INT\_fops->SetupStage(pdev)

DCD\_EP\_Tx(pdev,0,

pbuf,len);

Низкоуровневый драйвер

usb\_dcd\_int.c

usb\_dcd.c

USBD\_CtlSendData(pdev,CmdBuff,req->wLength)

APP\_FOPS.pIf\_Ctrl(req->bRequest,CmdBuff,req->wLength);

pdev->dev.class\_cb->Setup(pdev, req)

Приложение

usbd\_cdc\_if.c

USBD\_StdEPReq(pdev, &req)

Библиотека USB Device

Драйвер USBD

usbd\_core.c

usbd\_ioreq.c

usbd\_req.c

Класс (CDC)

usbd\_cdc\_core.c