1. Че это такое: **USB(Universal Serial Bus**) – интерфейс для передачи данных с периферийного устройства на хост(компьютер). Стандарт **USB** ввел единообразие в работу с широким спектром периферийных устройств. До его появления использовалось множество разных интерфейсов для подключения периферийных устройств (последовательный порт RS-232C для модема, мыши, параллельный порт для принтера и дисководов IOMEGA ZIP, PS/2 для мыши и клавиатуры, специализированные интерфейсы для подключения сканеров и т. д.).
2. Про библиотеку:  
   Она реализует универсальный низкоуровневый интерфейс, который позволяет взаимодействовать со всеми возможными USB-устройствами. С помощью библиотеки libusb можно получить список доступных USB-устройств, узнать их параметры, обмениваться данными с устройствами в синхронном и асинхронном режимах, реагировать на подключение и отключение устройств. Зачем нужна: Программирование взаимодействия с USB-устройствами достаточно трудоемко. Для упрощения работы с USB-устройствами из прикладных программ была создана многоплатформенная библиотека libusb.
3. Дескриптор устройства (libusb\_device\_descriptor) содержит основные характеристики устройства, такие как класс устройства, идентификаторы производителя и продукта. Эти данные хранятся в USB-устройстве и могут быть получены с помощью функции libusb\_get\_device\_descriptor.

**libusb\_get\_device\_descriptor:**

* **Назначение:** Получение дескриптора устройства (Device Descriptor).
* **Информация:** Дескриптор устройства содержит основные характеристики устройства, такие как:
  + **Идентификатор производителя (Vendor ID):** Уникальный идентификатор производителя.
  + **Идентификатор продукта (Product ID):** Уникальный идентификатор конкретного продукта.
  + **Класс устройства (Device Class):** Класс устройства, который определяет его функциональность.
  + **Количество конфигураций (Number of Configurations):** Количество возможных конфигураций устройства.
  + **Серийный номер (Serial Number):** Серийный номер устройства (если он доступен).
  + **Версия устройства (Device Version):** Версия устройства.

**libusb\_get\_config\_descriptor:**

* **Назначение:** Получение дескриптора конфигурации устройства (Configuration Descriptor).
* **Информация:** Дескриптор конфигурации содержит информацию о конкретной конфигурации устройства, включая:
  + **Количество интерфейсов (Number of Interfaces):** Количество интерфейсов, доступных в данной конфигурации.
  + **Описание интерфейсов (Interface Descriptors):** Информация о каждом интерфейсе, включая класс интерфейса, количество конечных точек и их типы.
  + **Описание конечных точек (Endpoint Descriptors):** Информация о каждой конечной точке, включая ее адрес, тип (например, IN или OUT), максимальный размер пакета и т.д.

Функция libusb\_get\_device\_descriptor используется для получения общей информации о устройстве, в то время как libusb\_get\_config\_descriptor предоставляет более детальную информацию о конкретной конфигурации устройства, включая интерфейсы и конечные точки. Эти функции дополняют друг друга, обеспечивая полный обзор устройства и его возможностей.

1. **Конечная точка (Endpoint):**
   1. **Логический объект:** Конечная точка — это логический объект, который используется для передачи данных между устройством и хостом.
   2. **Адрес:** Каждая конечная точка имеет уникальный адрес, который используется для идентификации ее в рамках интерфейса устройства.
2. **Типы конечных точек:**
   1. **IN Endpoint:** Конечная точка, через которую данные передаются от устройства к хосту. Например, данные с датчика или камеры.
   2. **OUT Endpoint:** Конечная точка, через которую данные передаются от хоста к устройству. Например, команды управления или данные для записи на устройство.
3. **Тип передачи данных:**
   1. **Control Endpoint:** Используется для управляющих команд и настройки устройства. Каждое устройство должно иметь хотя бы одну конечную точку управления (обычно это Endpoint 0).
   2. **Bulk Endpoint:** Используется для передачи больших объемов данных, например, файлов или изображений.
   3. **Interrupt Endpoint:** Используется для передачи данных с фиксированной частотой, например, данных с мыши или клавиатуры.
   4. **Isochronous Endpoint:** Используется для передачи данных в реальном времени, например, аудио или видео.
4. В контексте USB-устройств, термин "конфигурация" (Configuration) относится к набору параметров и настроек, которые определяют, как устройство будет работать в системе. Каждое USB-устройство может иметь несколько конфигураций, и пользователь или операционная система может выбрать одну из них для использования.

Конфигурация USB-устройства включает в себя:

1. **Интерфейсы (Interfaces):**
   * Интерфейс — это набор функциональных возможностей устройства, которые могут быть использованы одновременно. Например, для веб-камеры один интерфейс может быть предназначен для захвата видео, а другой — для захвата аудио.
2. **Конечные точки (Endpoints):**
   * Каждый интерфейс может иметь несколько конечных точек, через которые происходит обмен данными между устройством и хостом.
3. **Атрибуты и настройки:**
   * Конфигурация также включает в себя различные атрибуты и настройки, такие как потребляемая мощность, режимы работы и другие параметры.

**Сколько конфигураций может быть?**

USB-устройство может иметь одну или несколько конфигураций. Количество конфигураций определяется разработчиком устройства и может варьироваться от одной до нескольких десятков.

1. **Драйверы** — это программные модули, которые обеспечивают взаимодействие между операционной системой и конкретным USB-устройством. Они отвечают за интерпретацию данных, передаваемых по USB-шине, и управление устройством.
2. В контексте USB-устройств, термин "альтернативные настройки" (Alternate Settings) относится к различным режимам работы интерфейса устройства. Каждый интерфейс может иметь несколько альтернативных настроек, которые определяют, как устройство будет работать в разных режимах. Альтернативные настройки позволяют устройству переключаться между различными режимами работы без необходимости изменения конфигурации устройства. Это полезно, например, когда устройство может работать в разных режимах с разной скоростью передачи данных или с разными наборами конечных точек.

**Примеры альтернативных настроек:**

1. **Веб-камера:**
   * **Альтернативная настройка 1:** Захват видео с разрешением 640x480 и частотой 30 кадров в секунду.
   * **Альтернативная настройка 2:** Захват видео с разрешением 1280x720 и частотой 60 кадров в секунду.
2. **Аудиоустройство:**
   * **Альтернативная настройка 1:** Запись аудио с частотой дискретизации 44.1 кГц.
   * **Альтернативная настройка 2:** Запись аудио с частотой дискретизации 48 кГц.
3. Контекст (ctx) в библиотеке libusb — это структура данных, которая содержит информацию о текущем состоянии сессии работы с USB-устройствами. Он необходим для управления всеми операциями, связанными с USB.