Физический уровень

Физический уровень является первым уровнем модели OSI.

Он описывает способы передачи бит через физические среды, служащими линиями, соединяющие устройства.

Наша программа использует интерфейс RS-232-C для соединения компьютеров.

Последовательная передача данных означает, что данные передаются по единственной линии. При этом биты байта данных передаются по очереди с использованием одного провода. Для синхронизации группе битов используется ***стартовый бит***, после группы битов следуют ***бит проверки на четность*** и один или два ***стоповых бита****, но может и отсутствовать.*

*Исходное состояние линии последовательной передачи данных - уровень логической 1. Это состояние линии называют отмеченным —* ***MARK****. Когда начинается передача данных, уровень линии переходит в 0. Это состояние линии называют пустым —* ***SPACE****. Если линия находится в таком состоянии больше определенного времени, считается, что линия перешла в состояние разрыва связи —* ***BREAK****.*

**Контрольный бит** формируется на основе правила, которое создается при настройке передающего и принимающего устройства. Контрольный бит может быть установлен с контролем на четность, нечетность, иметь постоянное значение 1 либо отсутствовать совсем.

Если используется **бит четности**, то передается и он. Бит четности имеет такое значение, чтобы в пакете битов общее количество единиц (или нулей) было четно или нечетно, в зависимости от установки регистров порта. Этот бит служит для обнаружения ошибок, которые могут возникнуть при передаче данных из-за помех на линии. Приемное устройство заново вычисляет четность данных и сравнивает результат с принятым битом четности. Если четность не совпала, то считается, что данные переданы с ошибкой. Конечно, такой алгоритм не дает стопроцентной гарантии обнаружения ошибок. Так, если при передаче данных изменилось четное число битов, то четность сохраняется, и ошибка не будет обнаружена. Поэтому на практике применяют более сложные методы обнаружения ошибок.

*В самом конце передаются один или два* ***стоповых*** *бита завершающих передачу байта.*

*Передатчик и приемник должны использовать один и тот же формат данных, иначе обмен будет невозможен.*

*Используется один общий возвратный провод — сигнальная земля (нессимметричный интерфейс).*

*Данные передаются по единственной линии (последовательный интерфейс).*

*Биполярный потенциальный код +3В для нуля и -3В для единицы.*

# Обмен сигналами между сторонами интерфейса **RS-232C** выглядит так:

1. компьютер после включения выставляет сигнал **DTR**, который удерживается активным. Если второй компьютер включен, он отвечает компьютеру сигналом **DSR**.
2. Если первый компьютер получил сигнал **DSR** и хочет передать данные, он выставляет сигнал **RTS**;
3. когда второй компьютер готов принимать данные, он отвечает сигналом **CTS**. С этого момента адаптер может бит за битом передавать информацию по линии **TD**;
4. *получив байт данных, принимающий компьютер может сбросить свой сигнал* ***CTS****, информируя компьютер о необходимости “притормозить”*
5. *Когда принимающему компьютеру необходимо передать данные в отсылающий компьютер, он выставляет сигнал* ***DCD****. После этого начинается передача данных по* ***RD****.*

**Физический уровень выполняет следующие функции:**

1. Задание параметров COM-порта.
2. Установление физического канала.
3. Поддержание соединения.
4. Разъединение физического канала.
5. Передача информации из буфера в интерфейс.
6. Прием информации и ее накопление в буфере.

Данный функционал реализован в модулях **ft\_serial\_1.py** и **ft\_serial.py.** Многие функции из этих пакетов используют достаточно низкоуровневую библиотеку win32, относящуюся по сути к Си библиотеке WinApi.

Рассмотрим каждую из функций уровня.

**Задание параметров COM-порта.**

Начальная установка параметров COM-порта выполняется во время инициализации класса Serial. При этом происходит проверка валидности устанавливаемых значений параметров с помощью property. От пользователей требуется задать одни и те же настройки на обоих машинах. После этого вызывается функция конфигурации порта, которая заполняет структуру **DCB** новыми параметрами и настраивает COM-порт в соответствии с ней.

**Установление физического канала.**

Данная функция реализована с помощью функции open() класса Serial. После этого создается процесс чтения из буфера и процесс записи в буфер.

**Поддержание соединения.**

Инкапсулировано в winapi.

**Разъединение физического канала.**

Разъединение физического канала происходит с помощью функции close(). Она проверяет, состояние канала и, если он открыт, прекращает процессы чтения и записи и закрывает канал.

**Передача информации из буфера в интерфейс.**

Передача информации из буфера в интерфейс реализована посредством использования функции ft\_write(). Данная функция проверяет, открыт ли порт. Затем шифрует данные с помощью кодирования Хемминга и отправляет результат в порт интерфейса.

**Прием информации и ее накопление в буфере.**

Чтение информации реализовано посредством функции ft\_write(). Эта функция принимает количество байт для чтения. Получить количество доступных байт можно с помощью property in\_waiting.