



# Телекоммуникации

**Андрей Масленников**

**Москва, 2019**

# Темы лекций

## **1. Предмет телекоммуникаций.**

- 1.1. Сигналы и их характеристики.
- 1.2. Основы телекоммуникационного обмена.

## **2. Кодирование информации и данных.**

- 2.1. Кодирование информации и данных на физическом уровне.
- 2.2. Кодирование данных на логическом уровне.

## **3. Телекоммуникационные системы.**

- 3.1. Технологии межсетевое взаимодействия.
- 3.2. Протоколы межсетевое взаимодействия.
- 3.3. Стандарты цифровой передачи данных.
- 3.4. Стандарты беспроводной передачи данных.

## **4. Разработка приложений телекоммуникационного обмена.**

- 4.1. Аппаратная часть.
- 4.2. Программная часть.

---

# **Предмет телекоммуникаций**

---

# Сигналы

## **Аналоговый сигнал**

Изменение физической величины (например, напряжения).

## **Непрерывный сигнал**

Представление аналогового сигнала как непрерывной функции, имеющей непрерывное множество возможных значений.

## **Цифровой сигнал**

Последовательность цифровых значений в вычислителе.

## **Импульсный сигнал**

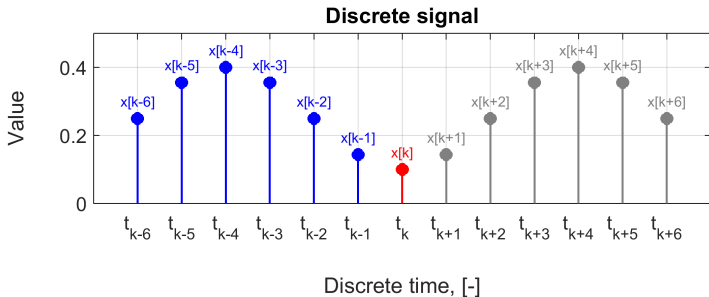
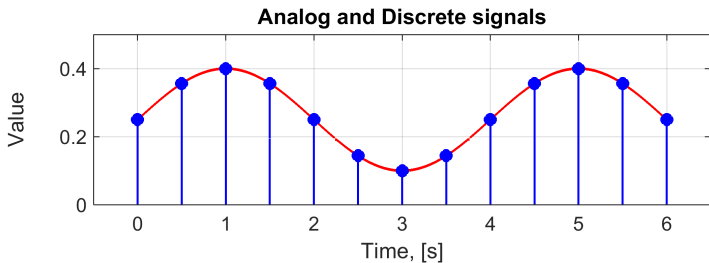
Аналоговый сигнал, являющийся последовательностью импульсов.

## **Дискретный сигнал**

Представление цифрового и импульсного сигналов, как дискретной функции, имеющей дискретное множество возможных значений.

# Сигналы

## Аналоговый и цифровой сигналы



**“Математическая” дискретизация:**

$$\begin{aligned} s(kT) &= s(0)\delta(t) + s(T)\delta(t-T) + \dots + s(nT)\delta(t-nT) \\ &= \sum_{k=0}^n s(kT)\delta(t-kT) = \sum_{k=0}^n s[k]\delta(t-kT) \end{aligned}$$

где:

- $T$  - период дискретизации [с];
- $\delta(t - kT)$  - дельта-функция Дирака со сдвигом на  $kT$ ;
- $f_d = 1/T$  - частота дискретизации [Гц].

**“Математическая” дискретизация - некорректный термин, его суть в математической имитации процесса физической дискретизации сигнала**

# Преобразования Фурье

## Преобразования Фурье

Интегральное преобразование функции одной вещественной переменной (в контексте ЦОС - времени) в функцию другой вещественной переменной (в контексте ЦОС - частоты).

## Виды преобразования Фурье

- непрерывное преобразование Фурье;
- преобразование Фурье для дискретного времени;
- дискретное преобразование Фурье;
- быстрое преобразование Фурье;  
*(алгоритмы вычисления ДПФ)*
- оконное преобразование Фурье.  
*(как дискретное так и непрерывное)*

# Преобразования Фурье

## Непрерывное преобразование Фурье

Прямое преобразование:

$$S(\omega) = \mathcal{F} \left\{ s(t) \right\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} s(t) e^{-j\omega t} dt$$

Обратное преобразование:

$$s(t) = \mathcal{F}^{-1} \left\{ S(\omega) \right\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} S(\omega) e^{j\omega t} d\omega$$

где:

- $\omega$  и  $S(\omega)$  - непрерывные величина и функция;
- $t$  и  $s(t)$  - непрерывные величина и функция;
- $e^{-j\omega t}$  и  $e^{j\omega t}$  - непрерывные функции.

$S(\omega)$ , по факту, аналитическая функция



# Преобразования Фурье

## Синус и косинус преобразования Фурье

**Общее преобразование Фурье:**

$$S(\omega) = \Re \{ S(\omega) \} + j \Im \{ S(\omega) \}$$

**Косинус преобразование Фурье:**

$$\Re \{ S(\omega) \} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} s(t) \cos \omega t dt$$

**Синус преобразование Фурье:**

$$\Im \{ S(\omega) \} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} s(t) \sin \omega t dt$$

Может быть удобнее вычислять аналитически

# Преобразования Фурье

Непрерывное преобразование Фурье для дискретного времени (Discrete-time)

**Прямое преобразование:**

$$S(\omega) = \mathcal{F} \left\{ s(k) \right\} = \sum_{-\infty}^{+\infty} s[k] e^{-j\omega k}$$

**Обратное преобразование:**

$$s(k) = \mathcal{F}^{-1} \left\{ S(\omega) \right\} = \frac{1}{2\pi} \int_{2\pi} S(\omega) e^{j\omega k} d\omega$$

где:

- $\omega$  и  $S(\omega)$  - непрерывные величина и функция;
- $k$  и  $s(k)$  - дискретные величина и функция;
- $e^{-j\omega k}$  и  $e^{j\omega k}$  - непрерывные функции.

$S(\omega)$ , по факту, аналитическая функция - сумма непрерывных экспонент

# Преобразования Фурье

## Дискретное преобразование Фурье (Discrete)

**Прямое преобразование:**

$$S(n) = \mathcal{F} \left\{ s(k) \right\} = \sum_{k=0}^K s[k] e^{-2\pi jkn/N}$$

**Обратное преобразование:**

$$s(k) = \mathcal{F}^{-1} \left\{ S(n) \right\} = \frac{1}{N} \sum_{n=0}^N S[n] e^{2\pi jkn/N}$$

где:

- $K$  - количество точек в сигнале;
- $N$  - количество частот в спектре;
- $n, S(n), k, s(k)$  - дискретные величины и функции.

Считаем, что дискретный сигнал  $K$ -периодичный

# Преобразования Фурье

Переход от непрерывного преобразования Фурье для дискретного времени

Непрерывное преобразование Фурье для дискретного времени:

$$S(\omega) = \mathcal{F} \left\{ s(k) \right\} = \sum_{-\infty}^{+\infty} s[k] e^{-j\omega k}$$

“Квантуем” частоту  $\omega$ :

$$\omega = \frac{2\pi n}{N}$$

Получим дискретное преобразование Фурье:

$$S(n) = S(\omega) \Big|_{\omega = \frac{2\pi n}{N}} = \sum_{-\infty}^{+\infty} s[k] e^{-j\omega k} \Big|_{\omega = \frac{2\pi n}{N}} = \sum_{k=0}^K s[k] e^{-2\pi jkn/N}$$

Пределы суммы меняются ввиду конечной длительности сигнала

# Преобразования Фурье

## Оконная функция

### Оконная функция

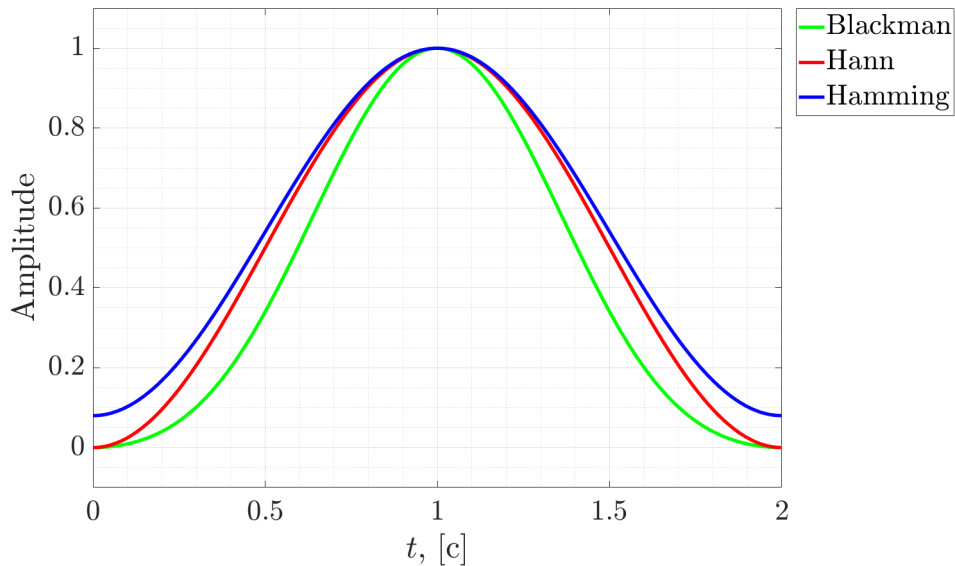
Это весовая функция (определенная на всем временном интервале от  $-\infty$  до  $+\infty$ ), которая используется для управления эффектами, обусловленными наличием боковых лепестков в спектральных оценках (растеканием спектра). С практической точки зрения используется для сведения начальных и конечных участков (значений) сигнала к одному уровню, приводящее к удалению вычислительной ошибки при определении спектров.

### Основные виды оконных функций:

- прямоугольная оконная функция;
- оконная функция Ханна;
- оконная функция Хамминга;
- оконная функция Блэкмана;
- оконная функция Кули-Тьюки.

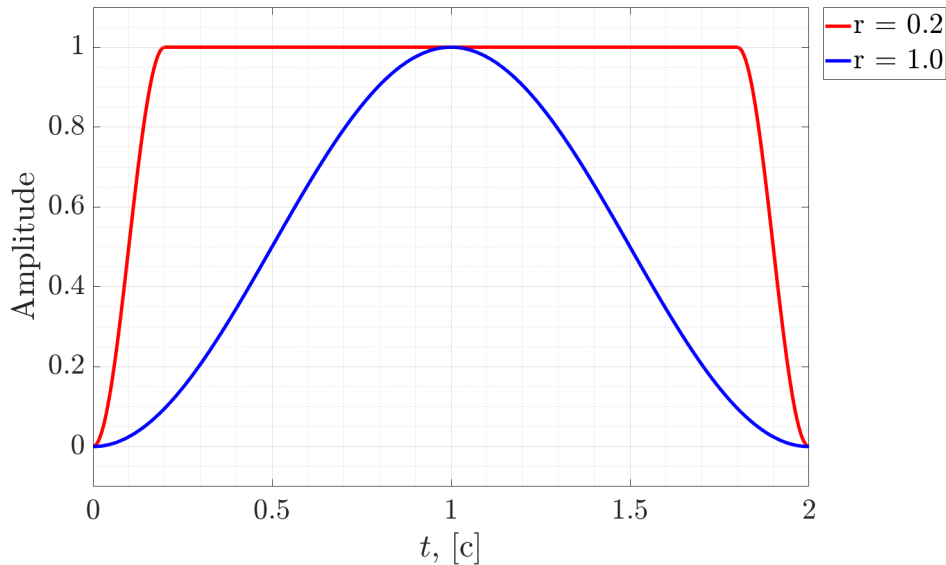
# Преобразования Фурье

Оконные функции Ханна, Хэмминга и Блэкмана



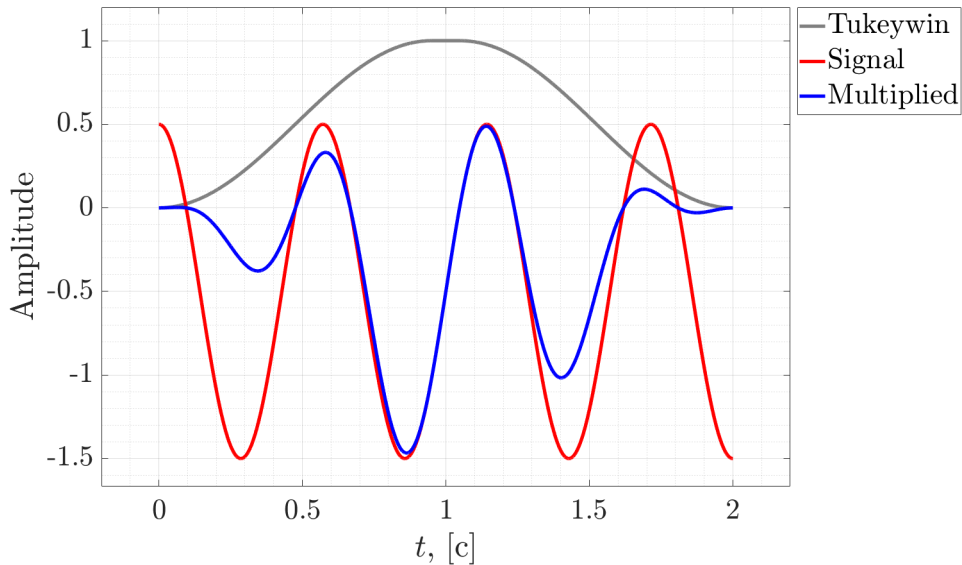
# Преобразования Фурье

Оконная функция Кули-Тьюки с различными значениями параметра  $r$



# Преобразования Фурье

Пример использования оконной функции





# Преобразования Фурье

## Оконное преобразование Фурье (Short-time)

Непрерывное преобразование Фурье:

$$S(\omega, \tau) = \mathcal{F} \left\{ s(t) \right\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} s(t) W(t - \tau) e^{-j\omega t} dt$$

Непрерывное преобразование Фурье для дискретного времени:

$$S(\omega, m) = \mathcal{F} \left\{ s(k) \right\} = \sum_{k=-\infty}^{+\infty} s[k] W(k - m) e^{-j\omega k}$$

Дискретное преобразование Фурье:

$$S(n, m) = \mathcal{F} \left\{ s(k) \right\} = \sum_{k=0}^K s[k] W(k - m) e^{-2\pi jkn/N}$$

# Преобразования Фурье

## Оконное преобразование Фурье

Используя формулу Эйлера:

$$e^{jx} = \cos(x) + j \sin(x)$$

Дискретное преобразование Фурье:

$$S(n) = \sum_{k=0}^K s[k] e^{-2\pi jkn/N} = \sum_{k=0}^K s[k] \left( \cos(-2\pi kn/N) + j \sin(-2\pi kn/N) \right)$$

Непрерывное преобразование Фурье для дискретного времени:

$$S(\omega) = \sum_{k=-\infty}^{+\infty} s[k] e^{-j\omega k} = \sum_{k=-\infty}^{+\infty} s[k] \left( \cos(-\omega k) + j \sin(-\omega k) \right)$$

Непрерывное преобразование Фурье:

$$S(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} s(t) e^{-j\omega t} dt = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} s(t) \left( \cos(-\omega t) + j \sin(-\omega t) \right) dt$$

# Преобразования Фурье

## Связь с рядами Фурье

Дискретное преобразование Фурье:

$$S(n) = \sum_{k=0}^K s[k] e^{-2\pi jkn/N} = \sum_{k=0}^K s[k] \left( \cos(-2\pi kn/N) + j \sin(-2\pi kn/N) \right)$$

Ряд Фурье:

$$s(t) = \frac{c_0}{2} + \sum_{k=1}^{+\infty} c_k \cos \left( 2\pi kt/T + \varphi_k \right)$$

Для вектора гармоник  $n = \{0, 1, \dots, N\}$  получаем вектор значений:

$$S[n] = a_n + j b_n$$

для непрерывного случая вычисляется аналогично с вектором частот  $\omega_j$

# Спектр сигнала

## Спектр сигнала

В широком терминологическом смысле спектр - это разложение сигнала (являющегося некоторой сложной функцией) на более простые в базисе ортогональных функций, например:

- $\sin$  и  $\cos$ ;  
(осуществляется преобразованием Фурье)
- функции Уолша;  
(осуществляется преобразованием Уолша-Адамара)
- вейвлет функции.  
(осуществляется вейвлет-преобразованием)

## Спектр сигнала в контексте ЦОС

Представление функции (сигнала)  $s(t)$ , заданной во временной области, в функцию, заданную в частотной области  $S(\omega)$ .

# Спектр сигнала

## Амплитудный спектр

Показывает на сколько сильно составляющая каждой частоты (гармонике) представлена в сигнале  $s(t)$ , что определяется как:

$$\left| S(\omega) \right| \quad \text{для непрерывного спектра}$$

$$\left| S(k) \right| \quad \text{для дискретного спектра}$$

## Фазовый спектр (фазовый сдвиг)

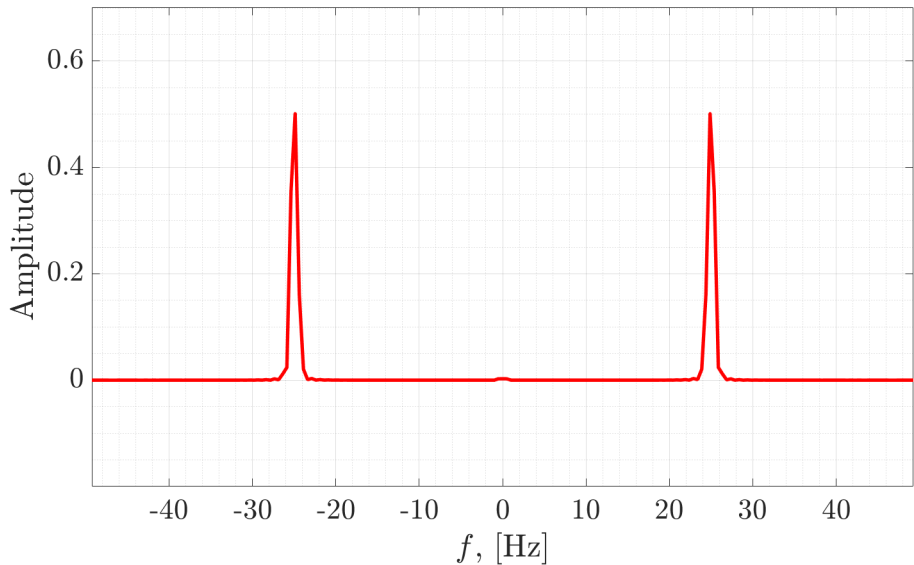
Показывает на сколько сильно сдвинута по фазе составляющая на каждой частоте (гармоники), что определяется как:

$$\angle S(\omega) \quad \text{для непрерывного спектра}$$

$$\angle S(k) \quad \text{для дискретного спектра}$$

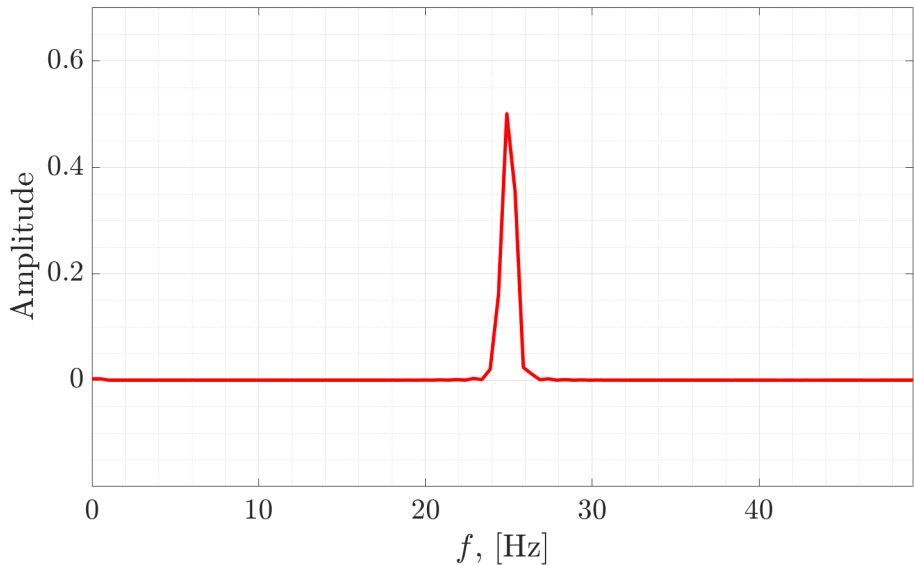
# Спектр сигнала

Пример двустороннего амплитудного спектра



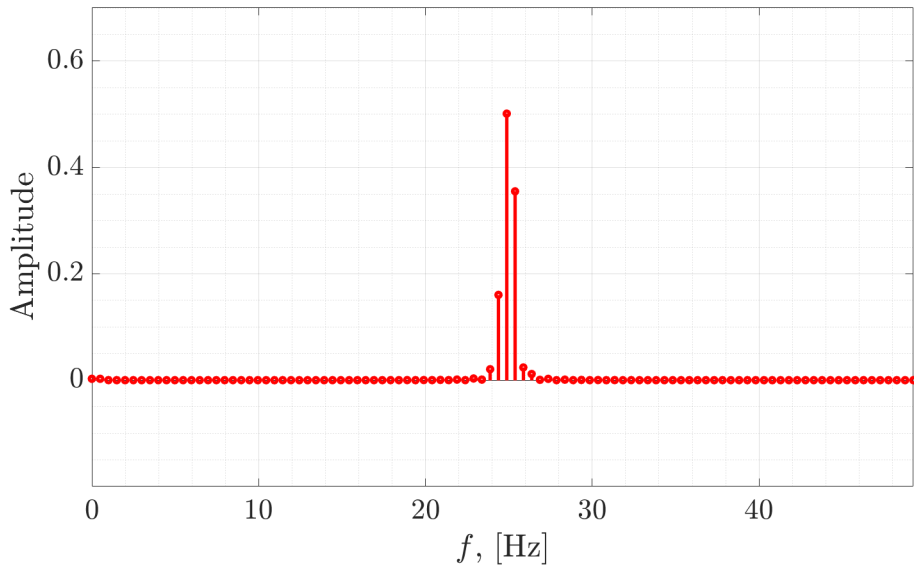
# Спектр сигнала

Пример одностороннего амплитудного спектра



# Спектр сигнала

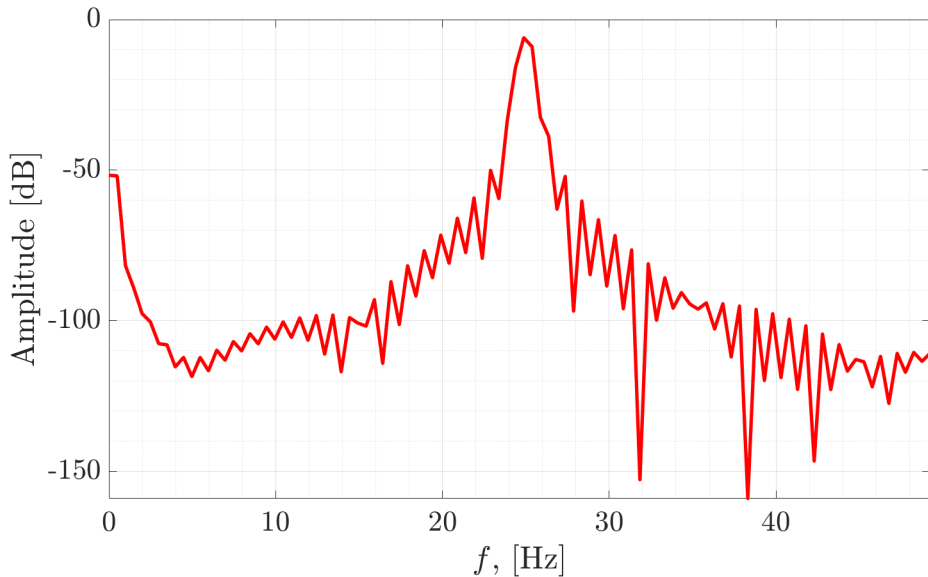
Пример одностороннего амплитудного спектра





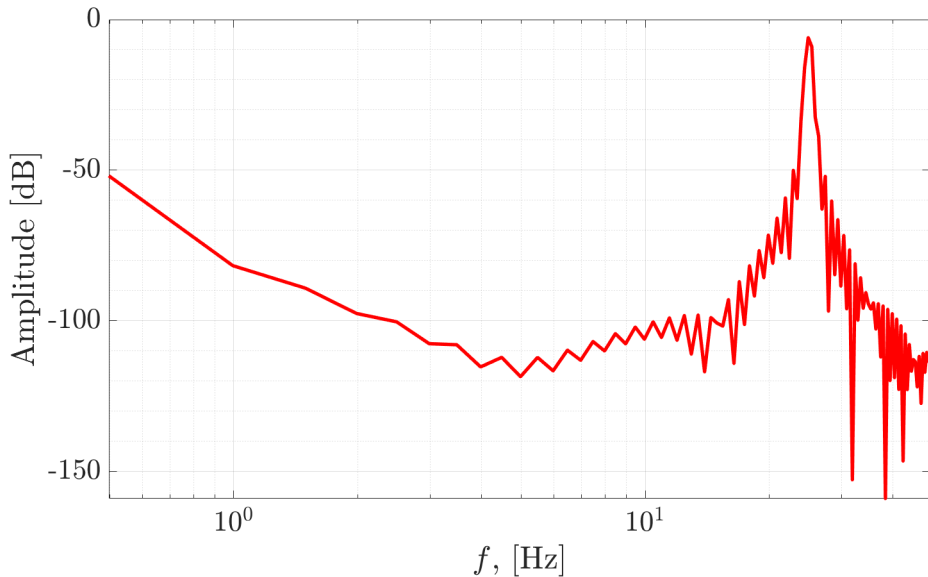
# Спектр сигнала

Пример одностороннего амплитудного спектра с амплитудой в дБ



# Спектр сигнала

Пример одностороннего амплитудного спектра в логарифмическом масштабе



# Спектр сигнала

## Алгоритм построения одностороннего амплитудного спектра

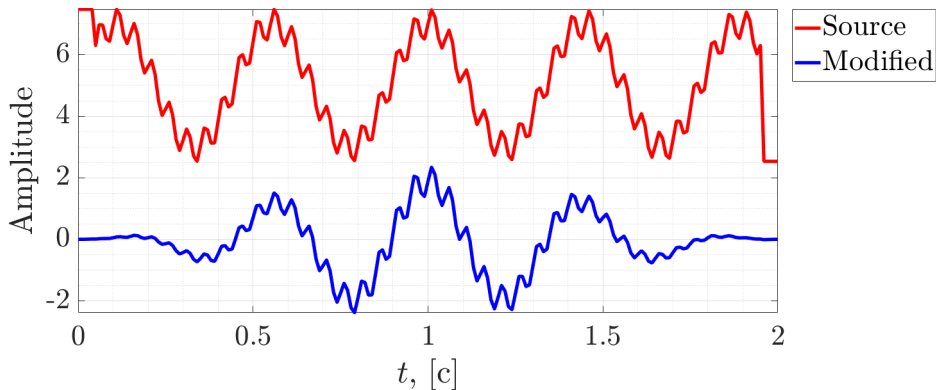
1.  $s = s - \text{mean}(s)$ .  
(удаляем среднее значение из сигнала)
2.  $s = s * W$ .  
(умножаем сигнал на вектор значений оконной функции)
3.  $S = \text{fft}(s, N)$ .  
(вычисляем дискретное преобразование Фурье)
4.  $S = \text{abs}(S)/N$ .  
(вычисляем амплитудный спектр и нормируем на  $N$ )
5.  $S = S(1 : N/2)$ .  
(выделяем половину спектра для области  $[0, +\omega]$ )
6.  $S(2 : N/2) = S(2 : N/2) * 2$ .  
(корректируем нормировку для  $N/2$  точек)
7.  $f = f_d * (0 : 1 : (N/2) - 1)^T / N$ .  
(формируем вектор частот в Гц)

# Спектр сигнала

## Алгоритм построения одностороннего амплитудного спектра

Сигнал:

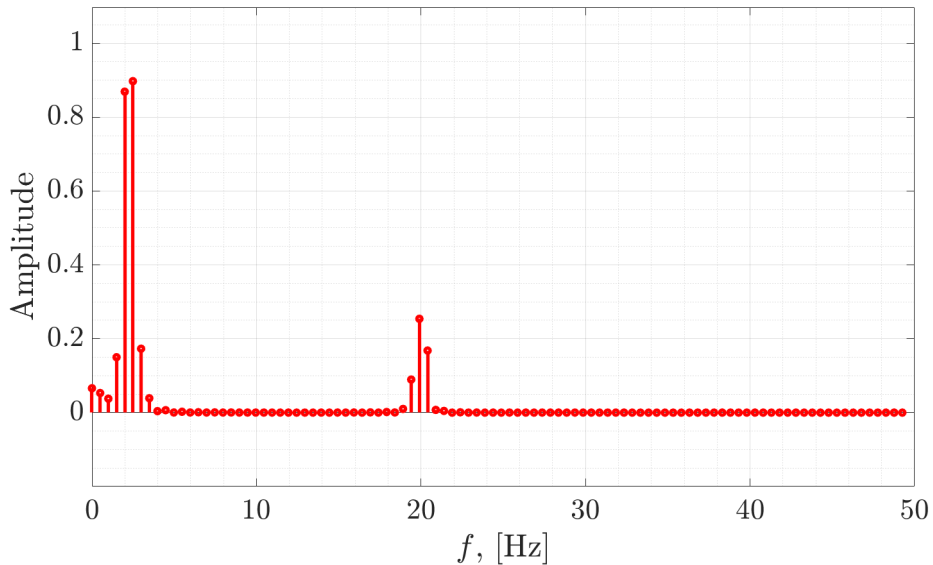
$$s(t) = 2.0 \cos(2\pi 2.25 t) + 0.5 \cos(2\pi 20 t) + 5.0$$



$$f_d = 100 \text{ Гц}, \quad t_{\max} = 2 \text{ с}, \quad r = 0.95$$

# Спектр сигнала

Алгоритм построения одностороннего амплитудного спектра



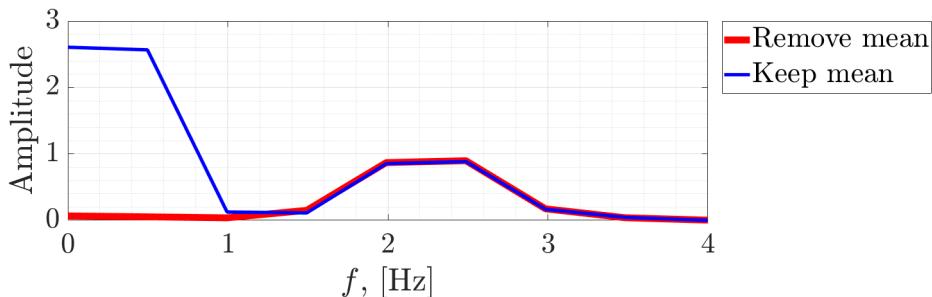
# Спектр сигнала

## Алгоритм построения одностороннего амплитудного спектра

1.  $s = s - \text{mean}(s)$ .

### Удаление среднего значения

Среднее значение это константа - составляющая на нулевой частоте, и ее вклад в спектре может существенно превышать вклад остальных частот, в результате чего график спектра будет визуально нечитаем.



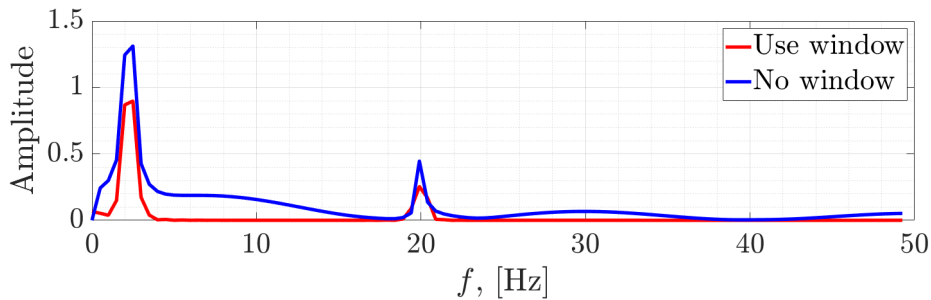
# Спектр сигнала

## Алгоритм построения одностороннего амплитудного спектра

2.  $s = s * W(t - \tau)$ .

### Умножение сигнала на оконную функцию

Если начало и конец сигнала не находятся в одной точке, то через них можно провести несколько низкочастотных составляющих, которых в сигнале нет.



# Спектр сигнала

## Алгоритм построения одностороннего амплитудного спектра

3.  $S = \text{fft}(s, N)$ .

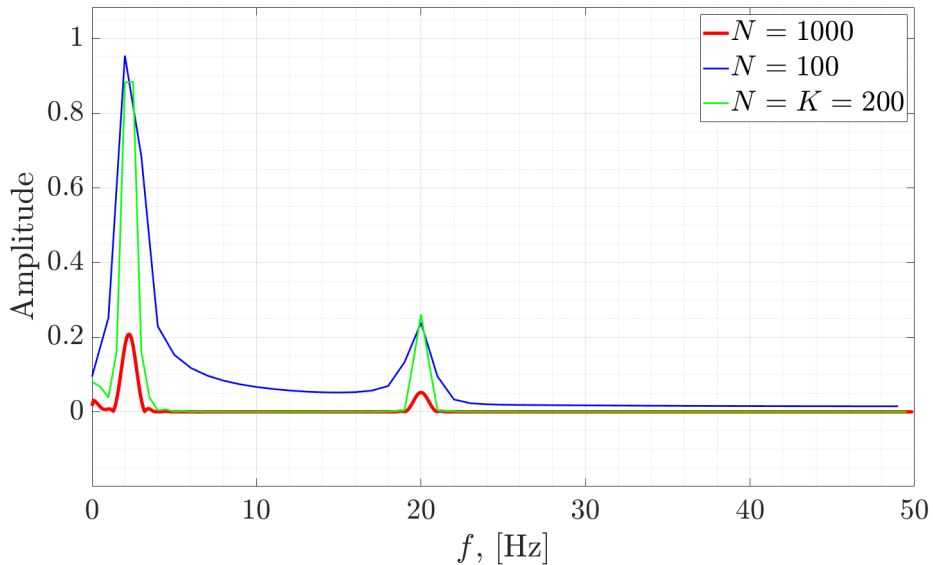
### Вычисление преобразования Фурье

В общем случае, наиболее точный спектр для дискретного сигнала определяется преобразованием Фурье при совпадении количества точек  $N$  преобразования Фурье и длительности сигнала  $K$ .



# Спектр сигнала

Количество точек преобразования Фурье



# Спектр сигнала

## Алгоритм построения одностороннего амплитудного спектра

4.  $S = \text{abs}(S)/N.$

### Нормировка на количество точек в сигнале

Проводится для того, чтобы спектры сигнала, полученного для  $K_1$  и  $K_2$  точек, были сравнимы между собой, как спектры одного и того же сигнала.

6.  $S(2 : N/2) = S(2 : N/2) * 2.$

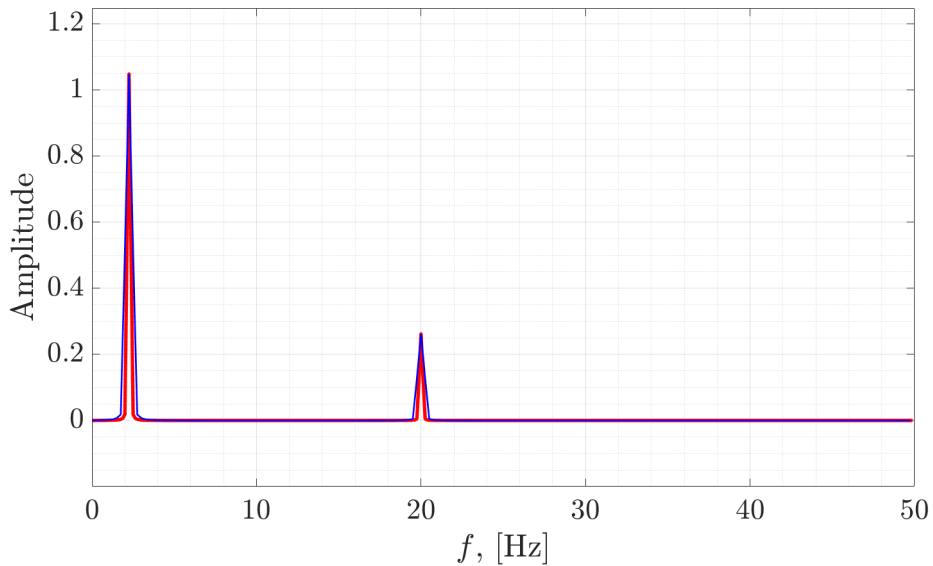
### Умножение на двойку

Вырезав половину, получим спектр с  $K/2$  точками, пронормированный ранее для  $K$  точек, поэтому домножаем на двойку. Первый элемент соответствует  $\omega = 0$  и в умножении на два не нуждается, т.к. он не имеет зеркального отображения.

# Спектр сигнала

Нормировка на количество точек в сигнале

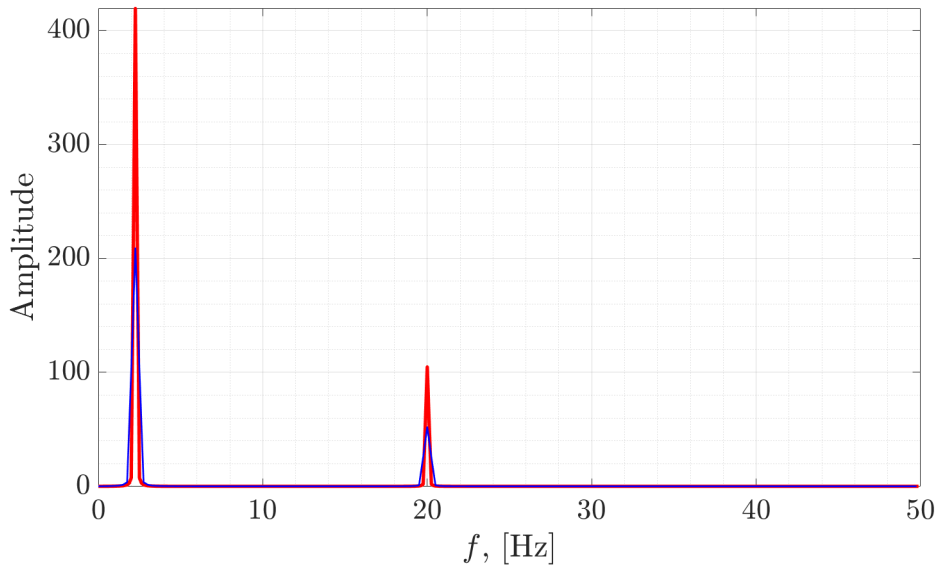
Do normalization



# Спектр сигнала

Нормировка на количество точек в сигнале

No normalization



# Спектр сигнала

## Алгоритм построения одностороннего амплитудного спектра

5.  $s = S(1 : N/2)$ .

### Выделение половины спектра

В общем случае преобразование Фурье определено на интервале  $[-\omega, +\omega]$  и зеркально относительно  $\omega = 0$ , поэтому выделяем спектр на интервале  $[0, +\omega]$ . Результат `fft` - вектор, где сначала идут значения преобразования Фурье для  $[0, +\omega]$  и затем  $[-\omega, 0]$ .

# Спектр сигнала

## Алгоритм построения одностороннего амплитудного спектра

$$7. \mathbf{f} = \mathbf{f}_d * (0 : 1 : (N/2) - 1)^T / N.$$

### Формирование вектора частот

При построении спектра для сигнала из  $K$  точек, получаем при  $N = K$  преобразование Фурье из  $K$  точек, т.е. спектр из  $K$  частот, где каждая половина спектра состоит из  $K/2$  точек, т.е.  $K/2$  гармоник (дискретных частот) от нуля до частоты Найквиста.

Рассмотренные методы построения спектра (через преобразования Фурье) предполагают, что период дискретизации  $T$  является константой. На практике, такое не всегда соблюдается. В зависимости от уровня отклонения  $T$  от постоянного значения может быть целесообразно переходить к дискретному преобразованию Фурье для неравномерно распределенных данных (Non-uniform discrete Fourier transform).

# Спектр сигнала

## Разрешение по частоте

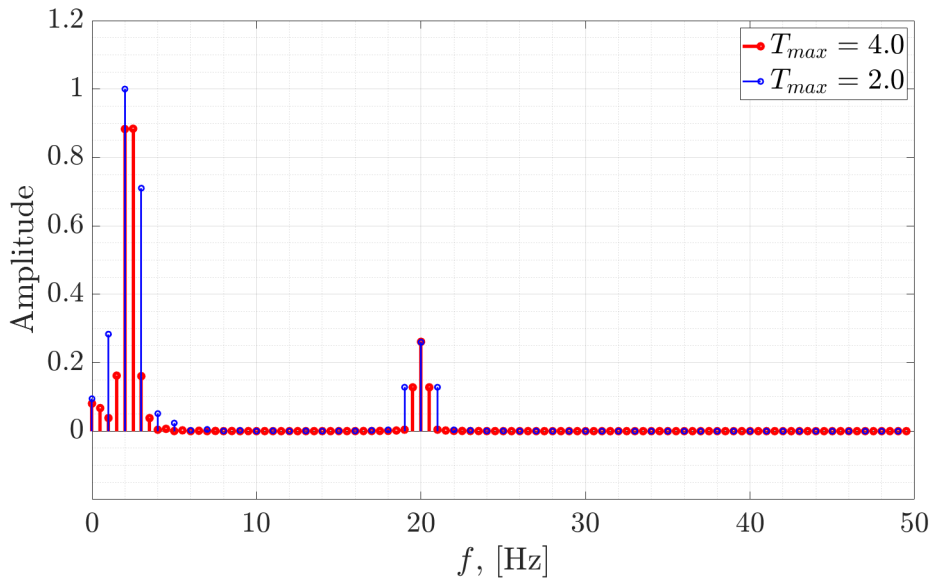
### Разрешение по частоте

Количество точек спектра, приходящихся на единицу частоты (Гц).

	Пр. 1	Пр. 2	Пр. 3
$f_d$ , Гц	100	100	100
$f_N$ , Гц	50	50	50
$T_{\max}$ , секунд	1	2	4
Точек в сигнале $K$	100	200	400
Точек преобразования Фурье $N$	100	200	400
Точек в спектре	100	200	400
Точек в одностороннем спектре	50	100	200
Разрешение по частоте, точек/Гц	1	2	4
$\Delta f$ между соседними частотами, Гц	1	0.5	0.25

# Спектр сигнала

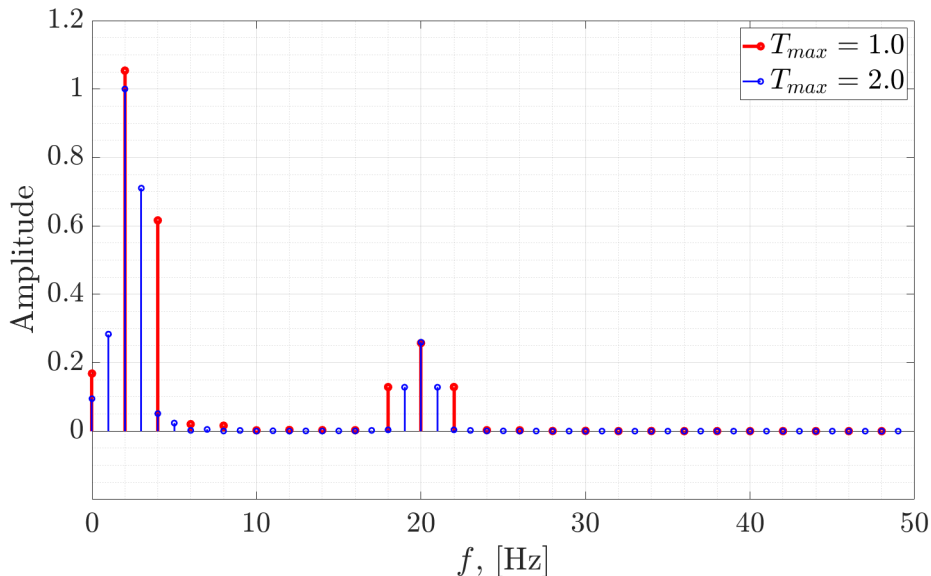
Разрешение по частоте в зависимости от изменения длительности сигнала





# Спектр сигнала

Разрешение по частоте в зависимости от изменения длительности сигнала



# Теорема Котельникова-Найквиста-Шеннона

## Теорема Котельникова-Найквиста-Шеннона

Любую непрерывную функцию, состоящую из частот от 0 до  $f_0$ , можно передавать с любой точностью в виде цифрового сигнала с частотой дискретизации более  $2 f_0$  (или: передавать каждое дискретное значение через период  $1/2 f_0$ ).

### Следствие (по сути, пояснение):

По полученному дискретному набору можно восстановить все частоты аналогового сигнала до частоты  $f_d/2$  с конечной точностью.

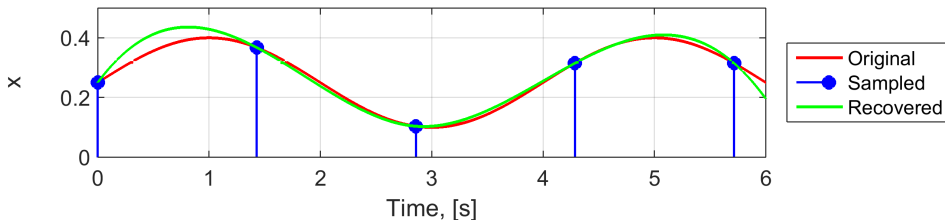
### Частота Найквиста:

Частота, равная половине частоты дискретизации:

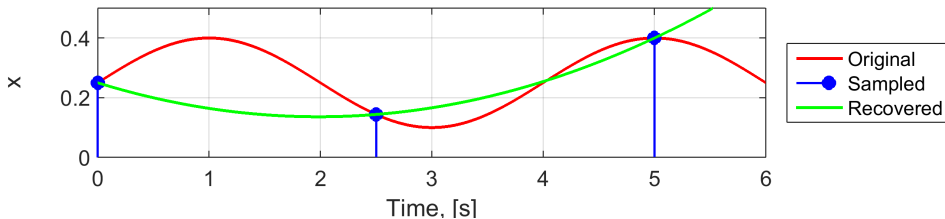
$$f_N = \frac{f_d}{2}$$

# Теорема Котельникова-Найквиста-Шеннона

$f > 2 f_s$ .  $2 \text{ Hz} > 0.5 \text{ Hz} (2 * 0.25)$



$f < 2 f_s$ .  $0.4 \text{ Hz} < 0.5 \text{ Hz} (2 * 0.25)$



# Теорема Котельникова-Найквиста-Шеннона

## Эффект наложения частот (спектров)

### Эффект наложения частот

Эффектом наложения спектров называется проявление в амплитудном спектре частот, превышающих частоту Найквиста. Эти проявления существенным образом могут исказить полученный амплитудный спектр.

Определение частоты  $f_a$ , на которой проявится составляющая  $f_b$ , по своему значению превышающая частоту Найквиста, осуществляется следующим образом:

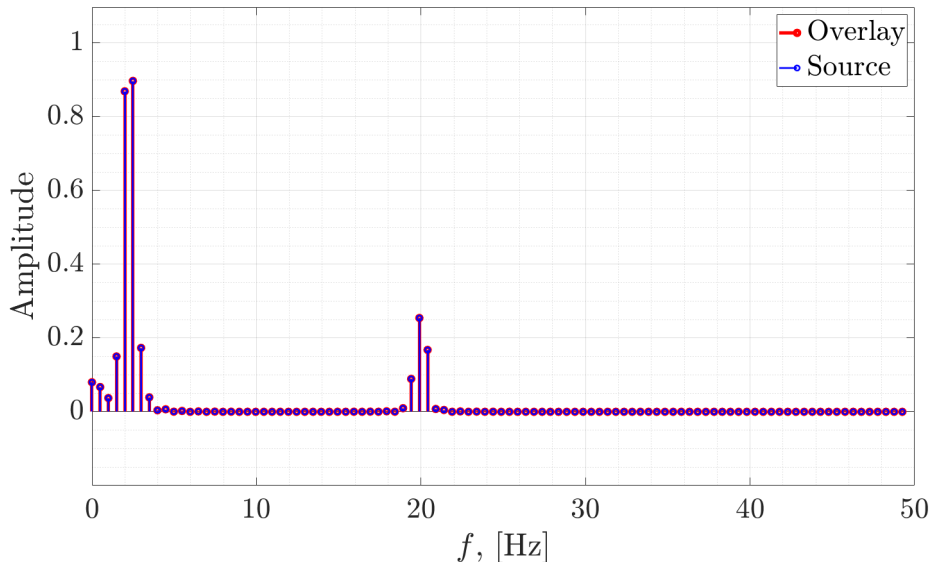
$$f_a = |f_d - f_b|$$

На практике, как правило, это проблема проявляется в переносе высокочастотных составляющих шума в область низких частот. Чтобы избежать этого, перед оцифровкой целесообразно аналоговый сигнал пропускать через ФВЧ.

# Теорема Котельникова-Найквиста-Шеннона

Эффект наложения частот (спектров)

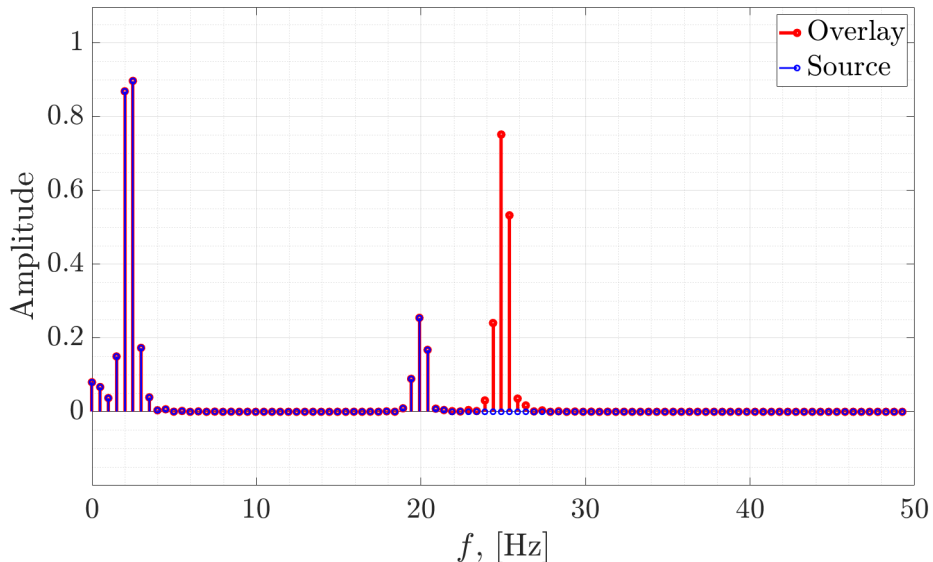
Add 50 Hz with  $F = 100$  Hz



# Теорема Котельникова-Найквиста-Шеннона

Эффект наложения частот (спектров)

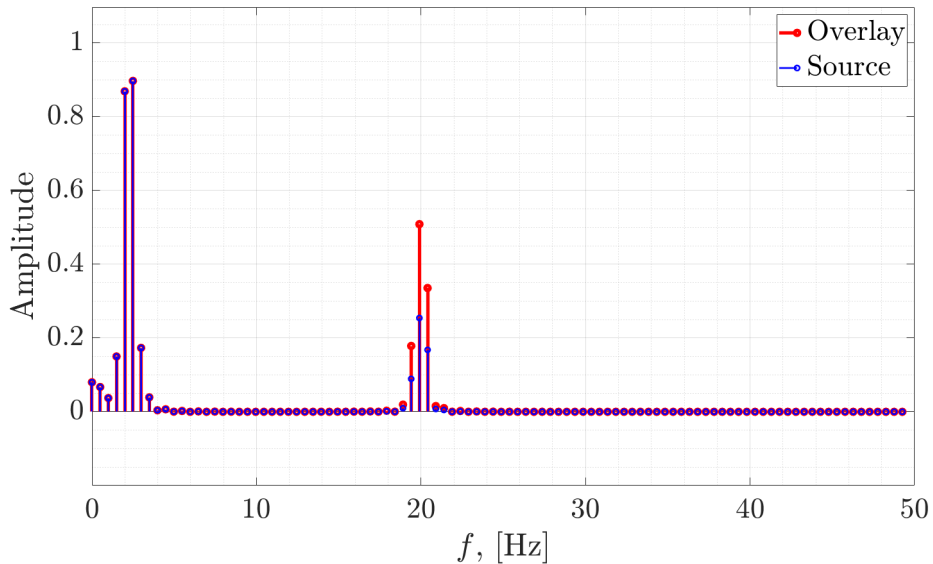
Add 75 Hz with  $F = 100$  Hz



# Теорема Котельникова-Найквиста-Шеннона

Эффект наложения частот (спектров)

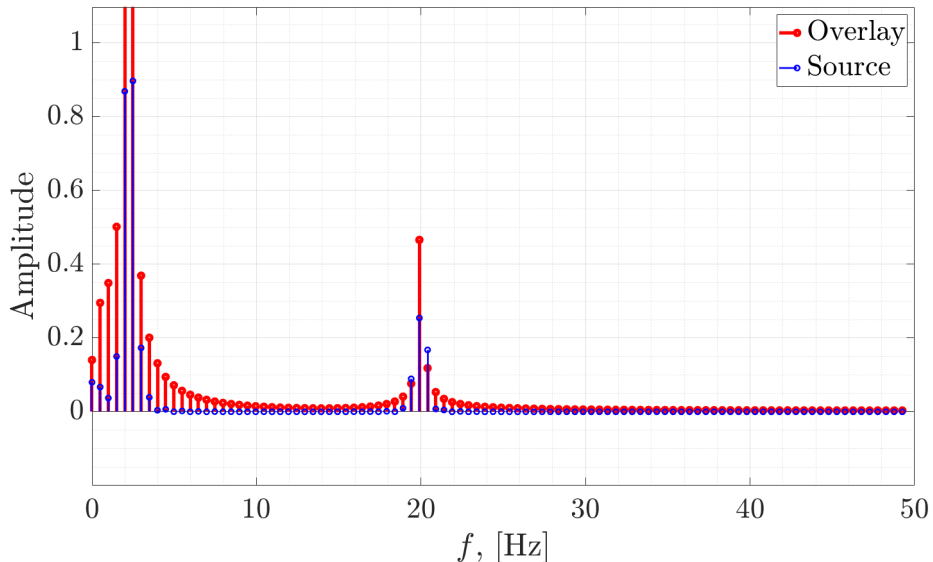
Add 80 Hz with  $F = 100$  Hz



# Теорема Котельникова-Найквиста-Шеннона

Эффект наложения частот (спектров)

Add 100 Hz with  $F = 100$  Hz (keep mean)

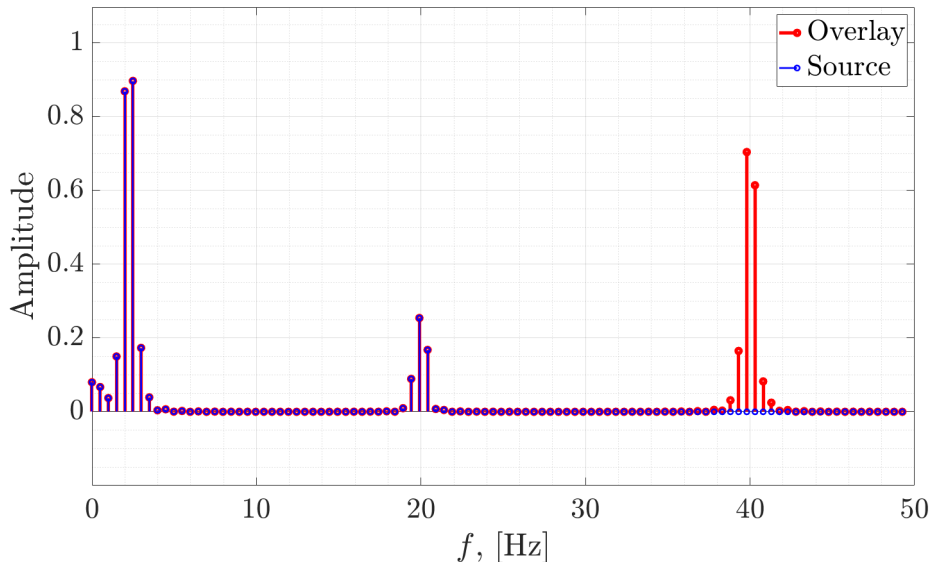




# Теорема Котельникова-Найквиста-Шеннона

Эффект наложения частот (спектров)

Add 240 Hz with  $F = 100$  Hz



# Характеристики сигналов

## Часто применяемые характеристики сигналов:

- во временной области:
  - энергия сигнала;
  - мощность сигнала;
  - средне-квадратичная мощность сигнала;
- в частотной области:
  - спектральная плотность энергии сигнала;
  - спектральная плотность мощности сигнала;
- соотношение сигнал-шум.

Строго говоря, энергия и мощность сигнала - это плотность энергии и мощности во временной области, а добавка “спектральная” относится к тем же характеристикам в частотной области

# Характеристики сигналов во временной области

## Энергия сигнала

### Энергия сигнала

Мера объема сигнала, для непрерывного случая:

$$E = \int_{-\infty}^{+\infty} |s(t)|^2 dt$$

и для дискретного случая:

$$E = \sum_{k=-\infty}^{+\infty} |s[k]|^2$$

Сигналы, для которых энергия бесконечна, - мощностные сигналы, например, периодические или стохастические сигналы, а если энергия конечна, то эти сигналы - энергетические

# Характеристики сигналов во временной области

## Мощность сигнала

### Мощность сигнала

Энергия в единицу времени, для непрерывного случая:

$$P = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{+T/2} |s(t)|^2 dt$$

и для дискретного случая:

$$P = \lim_{K \rightarrow \infty} \frac{1}{2K+1} \sum_{-K}^{+K} |s[k]|^2$$

или как средне-квадратичная мощность (для непрерывного случая):

$$P = \sqrt{\frac{1}{T} \int_0^T |s(t)|^2 dt}$$

# Характеристики сигналов в частотной области

## Спектральная плотность энергии и мощности сигнала

### Спектральная плотность энергии сигнала

$$E = \int_{-\infty}^{+\infty} |S(\omega)|^2 d\omega$$

### Спектральная плотность мощности сигнала

$$P = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\infty}^{+\infty} |S(\omega)|^2 d\omega$$

Как правило, речь идет о вычислении спектральной плотности мощности (Power Spectral Density - PSD) по дискретным значениям сигнала, для чего существуют специальные алгоритмы

# Характеристики сигналов

Связь характеристик во временной и частотной областях

## Равенство Парсеваля

Энергия одного и того же сигнала, рассчитанная во временной и в частотной областях равна друг другу:

$$\int_{-\infty}^{+\infty} |s(t)|^2 dt = \int_{-\infty}^{+\infty} |S(\omega)|^2 d\omega$$

Аналогично можно сказать и про мощности.

Равенство Парсеваля, как правило, указывается одним из свойств преобразования Фурье

# Соотношение сигнал-шум

## Соотношение сигнал-шум

Безразмерная величина, показывающая во сколько мощность полезного (информативного) сигнала превышает мощность шума, и может быть определена одним из следующих способов:

$$\text{SNR} = \frac{P_s}{P_n} = 10 \log_{10} \frac{P_s}{P_n} = \left( \frac{A_s^{\text{rms}}}{A_n^{\text{rms}}} \right)^2 = 20 \log_{10} \frac{A_s^{\text{rms}}}{A_n^{\text{rms}}}$$

где:

$P_s, P_n$  - мощность сигнала и шума;

$A_s^{\text{rms}}, A_n^{\text{rms}}$  - средне-квадратичная амплитуда сигнала и шума;

Пример:

$$\text{SNR} = 60 \text{ дБ} \quad \Rightarrow \quad \frac{A_s^{\text{rms}}}{A_n^{\text{rms}}} = 1000$$

# Основы телекоммуникационного обмена

## Сигнал в контексте телекоммуникаций

Пример сигнала:

$$s(t) = X \sin(2\pi f t + \varphi)$$

### Структурные параметры

Это число степеней свободы сигнала. Определяется спектральными характеристиками сигнала. Как правило, характеризуют широкополосность сигнала. Например: **t**.

### Идентифицирующие параметры

Позволяют идентифицировать (выделить) этот сигнал из всего множество сигналов поступающих на приемник. Например: **f**.

### Информационные параметры

Непосредственно включают в себя передаваемую полезную информацию, закодированную тем или иным методом. Например: **X**.



Вид информативного параметра определяет непрерывность или дискретность сигнала

### **Непрерывный (аналоговый) сигнал**

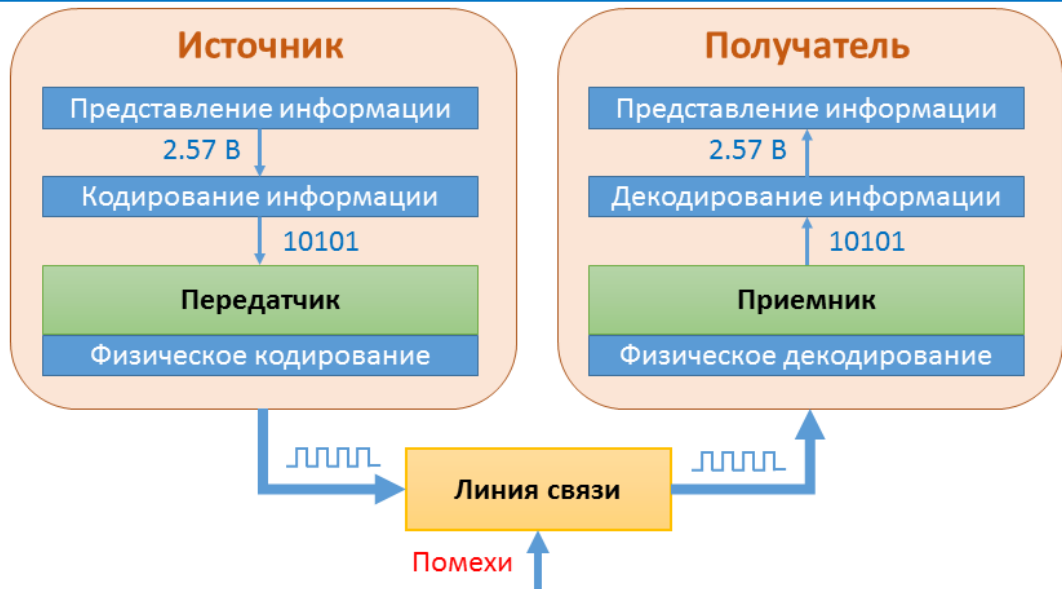
Количество возможных значений информационного параметра бесконечно (континуум). Передаваемая информация заложена в форме сигнала.

### **Дискретный сигнал:**

Количество возможных значений информационного параметра конечно и счетно. Передаваемая информация заложена либо в некоторых дискретных значениях (дискретный сигнал) либо в цифровых значениях (цифровой сигнал).

# Основы телекоммуникационного обмена

## Телекоммуникационные системы



### Требования по функционированию:

- передатчик и приемник должны использовать согласующиеся методы физического кодирования/декодирования;
- источник и получатель должны использовать согласующиеся методы кодирования/декодирования информации;
- источник и получатель должны использовать согласующиеся методы представления информации;
- линия связи должна обеспечивать:
  - требуемое качество/надежность передачи информации;
  - требуемую физическую пропускную способность;
  - требуемую эффективную пропускную способность;
- (по необходимости) должен обеспечиваться контроль и исправления возможных ошибок в полученных данных.

# Основы телекоммуникационного обмена

## Виды связи передатчика и приемника

### **Симплексный**

Данные передаются только в одном направлении от отправителя к получателю.

### **Полудуплексный**

Данные передаются поочередно в двух взаимно обратных направлениях от отправителя к получателю.

### **Дуплексный**

Данные передаются одновременно в двух взаимно обратных направлениях от отправителя к получателю.

# Основы телекоммуникационного обмена

## Согласования передатчика и приемника

### Передатчик и приемник могут согласовывать:

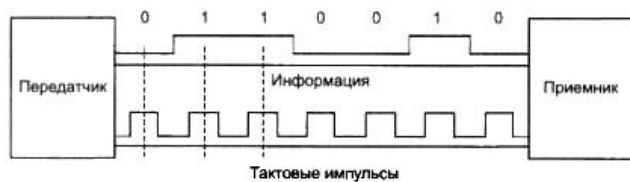
- скорость передачи бит;  
*(побитная / тактовая синхронизация)*
- формат передачи;  
*(наличие или отсутствие технических бит)*
- начало и конец каждого символа / байта;  
*(побайтная / посимвольная синхронизация)*
- начало и конец блока / сообщения / кадра.  
*(поблочная / покадровая синхронизация)*
- механизм использования сигналов;  
*(сигнал готовности данных для передачи и т.п.)*
- вид старт-стопных бит;  
*(отсылка к началу и концу байта / сообщения / кадра)*
- и др.

# Основы телекоммуникационного обмена

## Механизмы синхронизации передатчика и приемника

### Использование тактирующей линии

Между передатчиком и приемником устанавливается дополнительной канал связи (шина/линия синхронизации), по которому передается импульсный сигнал. Данные, получаемые приемником, учитываются только при определенном уровне сигнала на шине синхронизации.



### Самосинхронизация

Реализуется методом кодирования в самом сигнале, не требует дополнительного канала связи между передатчиком и приемником.

# Основы телекоммуникационного обмена

## Синхронизация передатчика и приемника

### **Асинхронный режим**

В асинхронном режиме работы данные передаются порциями, на линии возможны периоды отсутствия любой активности. В этом режиме синхронизация передатчика и приемника осуществляется добавлением в передаваемый код нескольких бит каждый раз, когда требуется передать новую порцию данных.

### **Синхронный режим**

При синхронной передаче старт-стопные биты между байтами данных отсутствуют, т.е. от передатчика к приемнику идет сплошной поток данных. При этом возможна потеря синхронизации, поэтому для использования синхронного режима необходимо использовать методы физического кодирования, обеспечивающие высокую самосинхронизацию.

### **Последовательная передача данных**

Данные одного сообщения от передатчика к приемнику пересылаются последовательно (бит за битом).

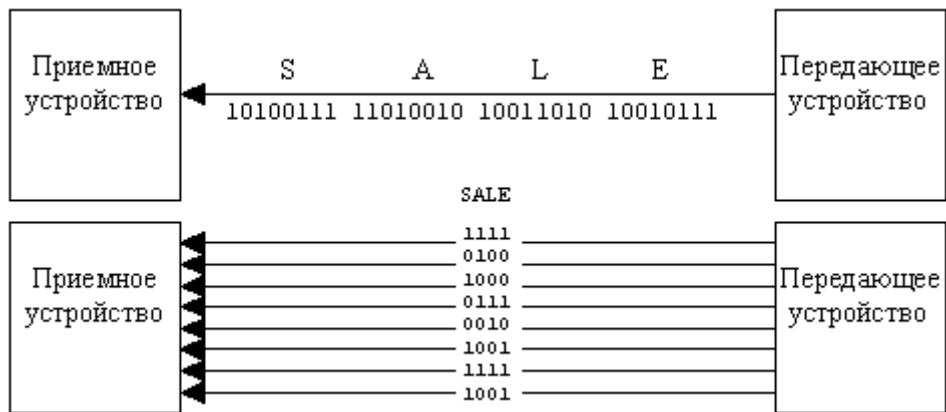
### **Параллельная передача данных**

Данные одного сообщения пересылаются параллельно (бит за битом). Возможные проблема при параллельной передаче данных - рассинхронизация параллельных путей передачи, наличие помех обусловленных влиянием нескольких путей передачи (физически - проводник) друг на друга.



# Основы телекоммуникационного обмена

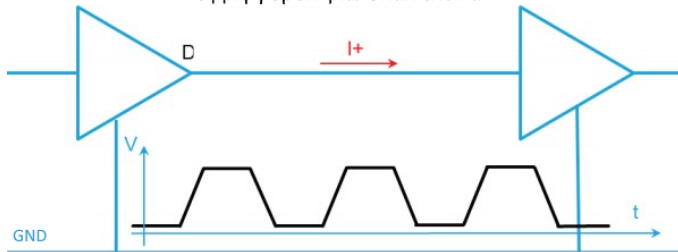
## Последовательная и параллельная передача данных



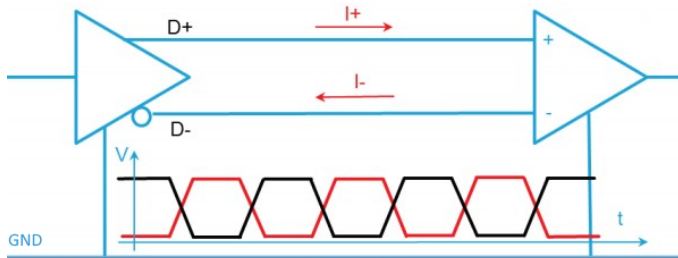
# Основы телекоммуникационного обмена

## Дифференциальная схема

Не дифференциальная схема



Дифференциальная схема



# Основы телекоммуникационного обмена

Канал связи, канал передачи, линия связи

## Канал связи

Технические средства и физическая среда передачи данных в одном направлении: от передатчика к приемнику.

## Канал передачи

Технические средства и физическая среда передачи данных в двух взаимно противоположных направлениях: от передатчика к приемнику и от приемника к тому же передатчику.

## Линия связи

Логическое объединение нескольких каналов передачи в рамках одной физической среды передачи данных.

# Основы телекоммуникационного обмена

## Методы разделения каналов передачи

### **Разделение по времени**

(Time Division Multiplexing - TDM): аппаратными средствами каждому каналу передачи предоставляется свой временной интервал для передачи данных.

### **Разделение по частоте**

(Frequency Division Multiplexing - FDM): каждый канал передачи работает на своей отдельной частоте, передача данных по различным каналам передачи осуществляется параллельно и независимо.

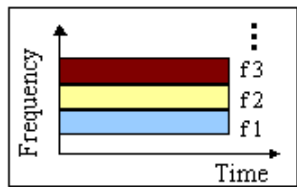
### **Кодовое разделение**

(Code Division Multiple Access - CDMA): каналы передачи имеют общую полосу частот, но разные кодирующие последовательности.

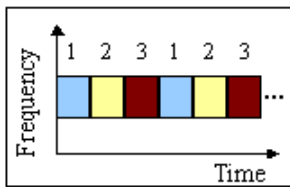
# Основы телекоммуникационного обмена

## Методы разделения каналов передачи

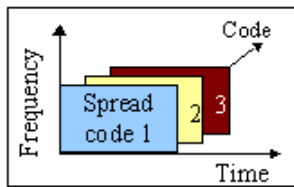
**FDMA**  
(Frequency Division  
Multiple Access)



**TDMA**  
(Time Division  
Multiple Access)



**CDMA**  
(Code Division  
Multiple Access)

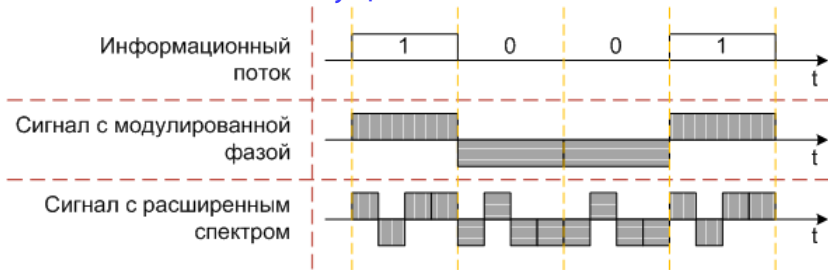


# Основы телекоммуникационного обмена

## Методы разделения каналов передачи

### Кодовое разделение

Несколько каналов одновременно используют весь рабочий диапазон частот. При кодировании применяется техника расширения спектра, т.е. один бит информации кодируется некоторой случайной последовательностью изменения несущего сигнала.



Последовательность для каждой пары приемника и передатчика уникальна, и по ней приемник и выделяет нужный ему информационный сигнал из всего потока.

# Основы телекоммуникационного обмена

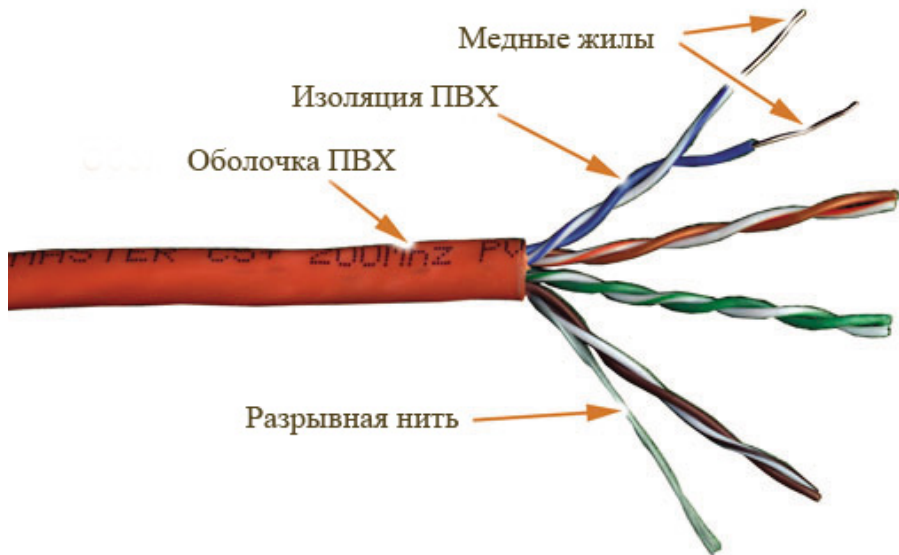
## Линии связи как физическая среда передачи данных

### Линии связи по физическому уровню:

- проводные;  
*(нет изоляции и экранирования)*
- кабельные:  
*(есть изоляция и экранирование)*
  - витая пара;
  - коаксиальный кабель;
  - оптоволоконный кабель;
- радио-волновые;
  - КВ/СВ/ДВ диапазон;
  - УКВ диапазон;
  - СВЧ излучение;  
*(сантиметровые и дециметровые волны)*
- инфракрасное излучение.

# Основы телекоммуникационного обмена

## Кабель витая пара

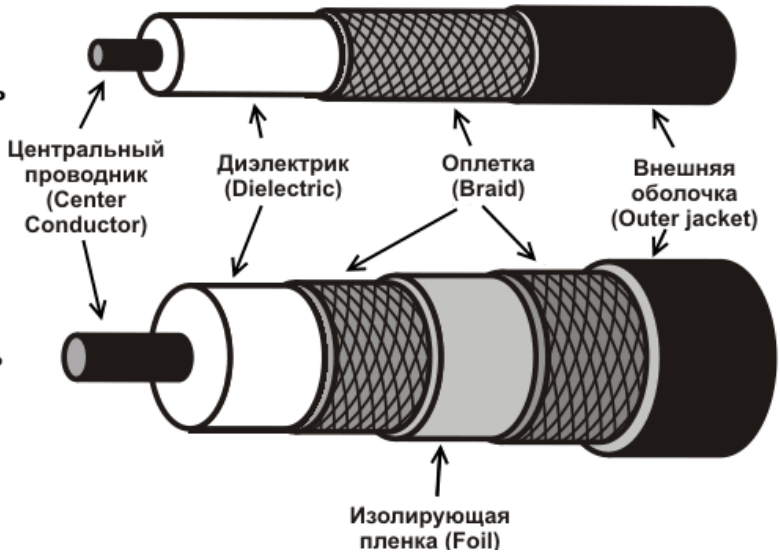




# Основы телекоммуникационного обмена

## Коаксиальный кабель

"Тонкий" (thin)  
коаксиальный кабель



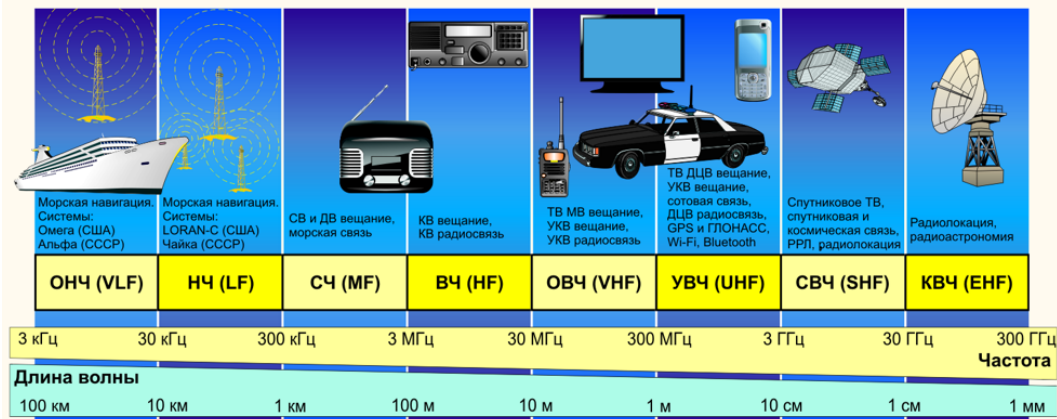
# Основы телекоммуникационного обмена

## Волоконно-оптический кабель



# Основы телекоммуникационного обмена

## Радиоволновый диапазон

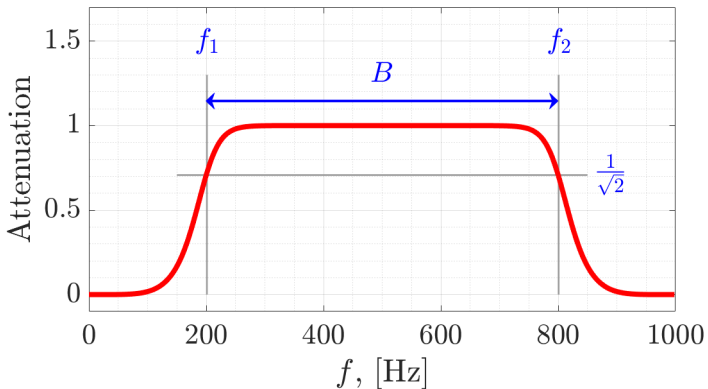


### Основные характеристики:

- амплитудно-частотная характеристика (АЧХ);
- полоса пропускания частот;
- величина затухания сигнала;
- помехоустойчивость на физическом уровне;
- соотношение сигнал-шум (SNR);
- пропускная способность:
  - физическая пропускная способность канала;
  - пропускная способность передачи информации (эффективная);
- достоверность передачи;
- экономическая целесообразность.

### Амплитудно-частотная характеристика (АЧХ)

Зависимость величины подавления сигнала от частоты.



### Полоса пропускания

Диапазон частот на котором мощность сигнала при передаче по линии связи падает менее чем в два раза:

$$\frac{P_{\text{Вых}}}{P_{\text{Вх}}} > 0.5$$

### Величина затухания

Падение мощности передаваемого сигнала, определяемое по амплитуде несущей частоты:

$$A = 10 \log \frac{P_{\text{Вых}}}{P_{\text{Вх}}}$$

### Помехоустойчивость (физическая)

Способность линии связи уменьшать влияние помех, что достигается за счет экранирования, изоляции и скручивания проводов (в кабельных системах).

### Соотношение сигнал-шум (Signal to Noise Ratio)

Отношение мощности полезного сигнала к мощности шума (стохастический/случайный процесс):

$$SNR = \frac{P_{\text{sgnl}}}{P_{\text{noise}}} = \left( \frac{A_{\text{sgnl}}^{\text{rms}}}{A_{\text{noise}}^{\text{rms}}} \right)^2$$

# Основы телекоммуникационного обмена

## Характеристики линии связи

### **Пропускная способность передачи информации (эффективная)**

Количество информации (например, бит), которое можно передать по каналу связи в единицу времени. Единицы измерения: бит/с.

### **Физическая пропускная способность канала**

Количество изменений несущего сигнала (например, количество импульсов) в единицу времени. Единицы измерения: бод/с.

бит/с и бод/с зачастую не совпадают, т.к. один бит может быть закодирован несколькими изменениями несущего сигнала



### Формула Шеннона-Хартли:

$$C = B \log_2(1 + SNR)$$

где:

- $C$  - максимальная пропускная способность (б/с);
- $B$  - ширина полосы пропускания (Гц).

### Формула Найквиста:

$$C = 2 B \log_2(M)$$

где:

- $C$  - максимальная пропускная способность (б/с);
- $B$  - ширина полосы пропускания (Гц);
- $M$  - число различных состояний сигнала.

# Основы телекоммуникационного обмена

## Характеристики линии связи

### Достоверность передачи

При передачи цифровых сигналов отношение одного ошибочного бита на  $N$  переданных бит (Bit Error Rate).

### Экономическая целесообразность:



**Денег нет!  
Но вы держитесь!**

---

# Кодирование информации и данных

---

## Кодирование информации и данных

Под кодированием информации и данных понимается их преобразование с целью повышения надежности, защищенности или скорости передачи.

### Виды кодирования:

- на физическом уровне;  
*(то, как информация и данные будут представлены в виде изменения некоторой физической величины, например характером изменения напряжения на линии связи)*
- на логическом уровне;  
*(применимо только у цифровым данным)*

## Модуляция

Процесс изменения одного или нескольких параметров высокочастотного (несущего) сигнала по закону низкочастотного (информационного) сигнала. По сути, процесс кодирования аналоговой информации.

### Виды модуляции:

- модулируемый сигнал гармонический (узкий спектр):
  - аналоговая модуляция;
  - цифровая модуляция (манипуляция);
- модулируемый сигнал импульсный (широкий спектр):
  - импульсная модуляция;
- расширение спектра (беспроводные системы).

# Аналоговая модуляция

## Аналоговая модуляция

“Кодирование” аналогового сигнала аналоговым сигналом.

### Основные виды аналоговой модуляции:

- амплитудная модуляция (AM):
  - двусторонняя (DSB);
  - с одной боковой полосой (SSB);
  - балансная (подавляется несущая) (SSB-SC);
  - квадратурная (QAM);  
*(применяется в цифровой манипуляции)*
- частотная модуляция (FM);
- фазовая модуляция (PM).

# Аналоговая модуляция

## Амплитудная модуляция

Амплитуда несущего сигнала модулируется амплитудой информационного сигнала.

## Частотная модуляция

Частота несущего сигнала модулируется амплитудой информационного сигнала.

## Фазовая модуляция

Фаза несущего сигнала модулируется амплитудой информационного сигнала.

## Квадратурная модуляция

Сумма двух несущих сигналов одной частоты, сдвинутых по фазе на  $\pi/2$ , модулируются амплитудами двух информационных сигналов.

# Аналоговая модуляция

## Двухсторонняя амплитудная модуляция (DSB AM)

Несущий сигнал:

$$s_c(t) = A \sin(\Omega_0 t)$$

Информационный сигнал:

$$s_i(t) = \sin(\omega t)$$

Амплитуда несущего сигнала:

$$A = A_0 [1 + m s_i(t)] = A_0 [1 + m \sin(\omega t)]$$

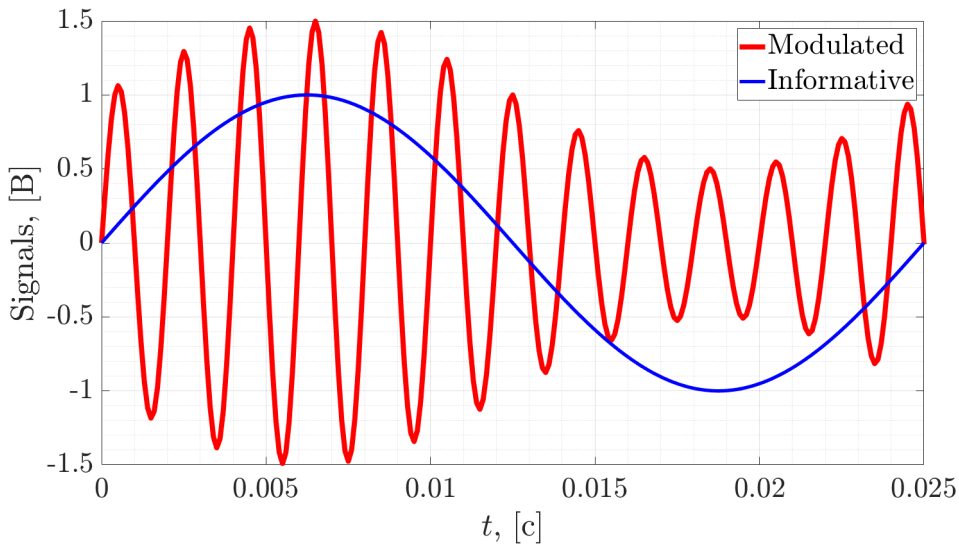
Модулированный сигнал:

$$\begin{aligned} s(t) &= A_0 [1 + m \sin(\omega t)] \sin(\Omega_0 t) \\ &= A_0 \left[ \sin(\Omega_0 t) + \frac{m}{2} \cos((\Omega_0 - \omega) t) - \frac{m}{2} \cos((\Omega_0 + \omega) t) \right] \end{aligned}$$



# Аналоговая модуляция

## Двухсторонняя амплитудная модуляция

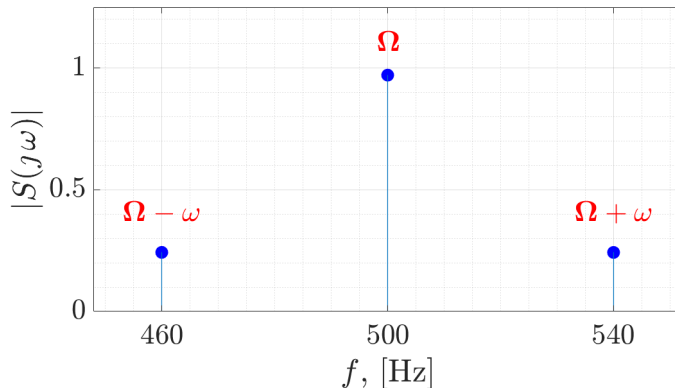


# Аналоговая модуляция

## Двухсторонняя амплитудная модуляция. Спектр

Модулированный сигнал:

$$s(t) = A_0 \left[ \sin(\Omega_0 t) + \frac{m}{2} \cos((\Omega_0 - \omega) t) - \frac{m}{2} \cos((\Omega_0 + \omega) t) \right]$$



# Аналоговая модуляция

Двухсторонняя амплитудная модуляция. Коэффициент модуляции  $m$

Модулированный сигнал:

$$s(t) = A_0 \left[ 1 + m \sin(\omega t) \right] \sin \left( \Omega_0 t \right)$$

где коэффициент модуляции  $m$  определяется как:

$$m = \frac{A_{\max} - A_{\min}}{A_{\max} + A_{\min}} = \frac{\Delta A}{A_0}$$

## Перемодуляция

Возникает при  $m > 1$ , проявляется в искажении сигнала и приводит к потере информации (невозможно однозначно демодулировать сигнал).

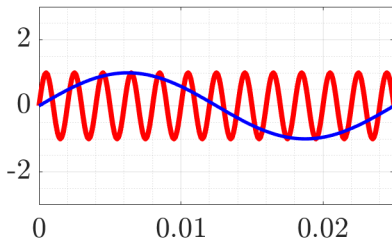
## Отсутствие модуляции (слабая модуляция)

Возникает при  $m \ll 1$  и приводит к очень слабому вкладу информационного сигнала в модулированный, что при наличии помех на линии не позволит выделить информационную составляющую.

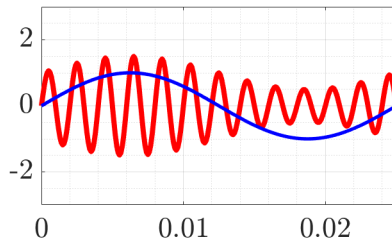
# Аналоговая модуляция

Двухсторонняя амплитудная модуляция. Коэффициент модуляции  $m$

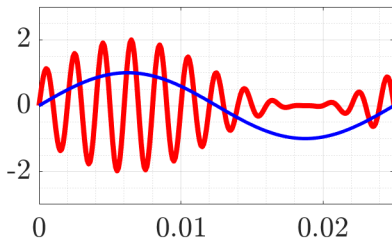
$m = 0.0$



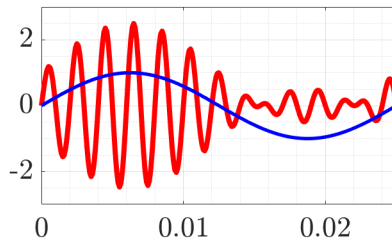
$m = 0.5$



$m = 1.0$



$m = 1.5$



# Аналоговая модуляция

## Другие виды амплитудной модуляции

### **Однополосная модуляция (Single-Side Band)**

Подавляется одна из боковых полос спектра. Реализуется за счет либо фильтрации (фильтр с высокой добротностью), либо фазовой инверсией (одна из боковых полос инвертируется по фазе и складывается сама с собой).

### **Подавление несущей (DSB/SSB Suppress Carrier)**

Реализуется либо фильтрацией (в данном случае не оптимально, т.к. может быть отфильтрована часть боковой полосы с информацией), либо балансной модуляцией, что приводит к существенному усложнению используемых аппаратных элементов.

# Аналоговая модуляция

## Частотная модуляция (FM)

Несущий сигнал:

$$s_c(t) = A_0 \sin(\Omega t)$$

Информационный сигнал:

$$s_i(t) = \sin(\omega t)$$

Частота несущего сигнала:

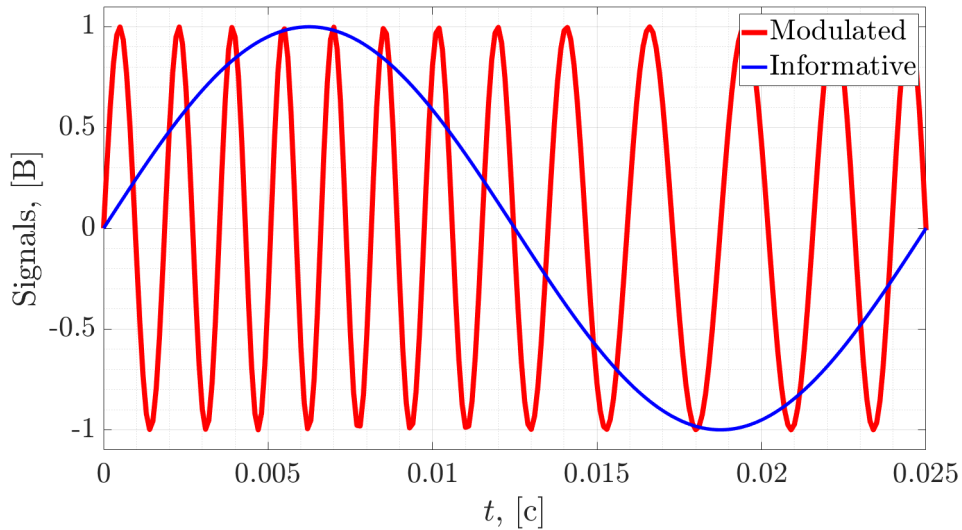
$$\Omega = \Omega_0 [1 + m s_i(t)] = \Omega_0 [1 + m \sin(\omega t)] , \quad m = \frac{\Delta \omega}{\Omega_0}$$

Модулированный сигнал:

$$\begin{aligned} s(t) &= A_0 \sin \left( \int_0^t \Omega dt \right) = A_0 \sin \left( \Omega_0 t + \Delta \omega \int_0^t \sin(\omega t) dt \right) \\ &= A_0 \left[ \sin \left( \Omega_0 t \right) \cos \left( \beta \cos(\omega t) \right) + \cos \left( \Omega_0 t \right) \sin \left( \beta \cos(\omega t) \right) \right] \end{aligned}$$

# Аналоговая модуляция

## Частотная модуляция



# Аналоговая модуляция

## Частотная модуляция. Спектр

Модулированный сигнал:

$$s(t) = A_0 \left[ \sin(\Omega_0 t) \cos(\beta \cos(\omega t)) + \cos(\Omega_0 t) \sin(\beta \cos(\omega t)) \right]$$

где:

$$\beta = \frac{\Delta\omega}{\omega} \quad - \text{ индекс частотной модуляции;}$$

$$\Delta\omega \quad - \text{ девиация частоты.}$$

При  $\beta \ll 1$  (узкий спектр модулированного сигнала):

$$s(t) = A_0 \left[ \sin(\Omega_0 t) + \frac{\beta}{2} \cos((\Omega_0 + \omega)t) - \frac{\beta}{2} \cos((\Omega_0 - \omega)t) \right]$$

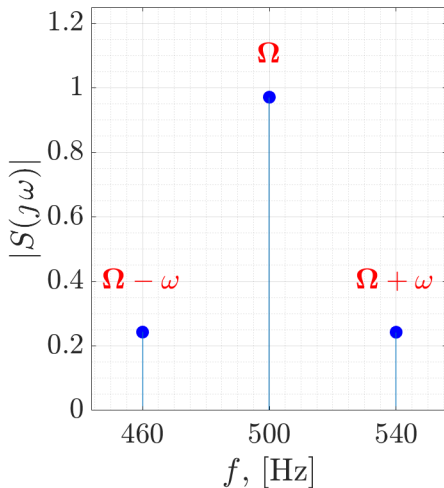
**Спектр ЧМ совпадает со спектром АМ,  
однако ЧМ более помехоустойчива чем АМ**



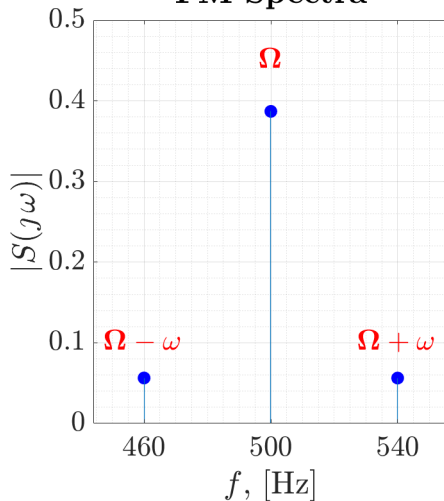
# Аналоговая модуляция

## Частотная модуляция. Спектр

### AM Spectra



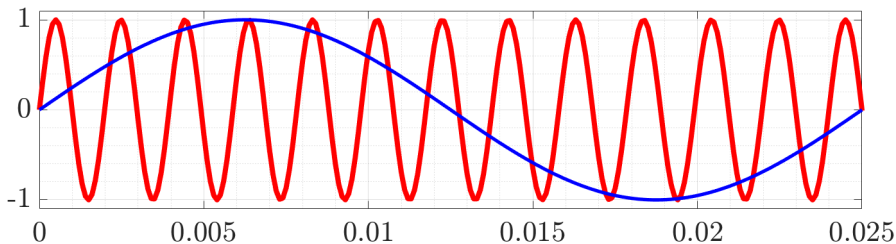
### FM Spectra



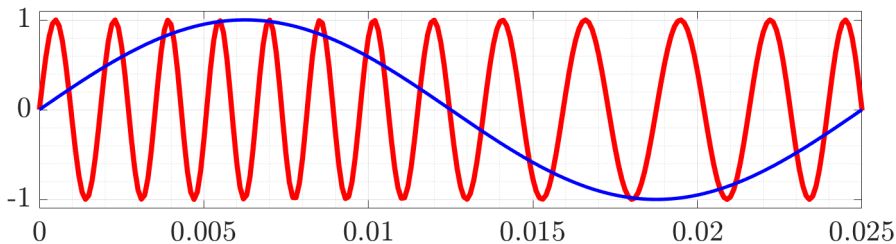
# Аналоговая модуляция

Частотная модуляция. Девиация частоты  $\Delta \omega$

$\Delta f = 5 \text{ Hz}$



$\Delta f = 50 \text{ Hz}$



# Аналоговая модуляция

## Фазовая модуляция (PM)

Несущий сигнал:

$$s_c(t) = A_0 \sin(\Omega_0 t + \varphi)$$

Информационный сигнал:

$$s_i(t) = \sin(\omega t)$$

Фаза несущего сигнала:

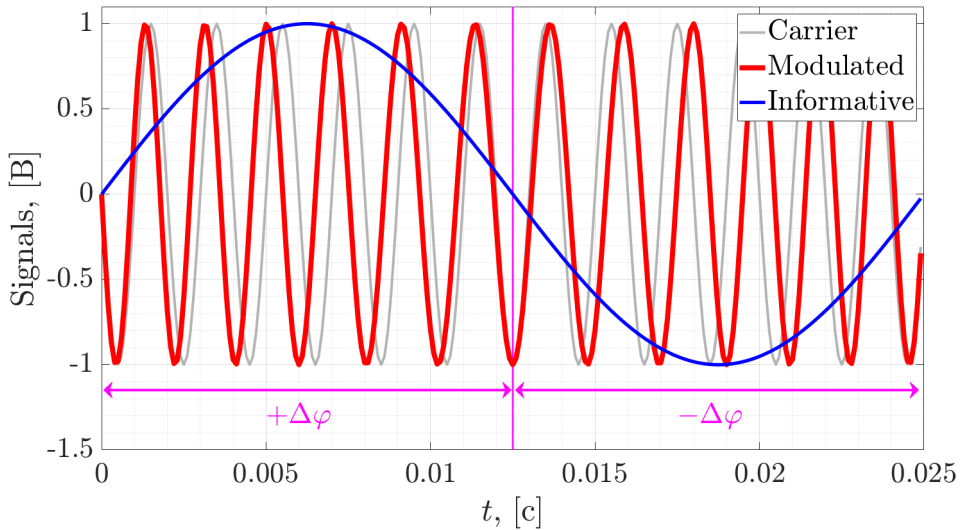
$$\varphi = \varphi_0 [1 + m s_i(t)] , \quad m = \frac{\Delta \varphi}{\varphi_0}$$

Модулированный сигнал:

$$\begin{aligned} s(t) &= A_0 \sin(\Omega_0 t + \varphi) = A_0 \sin(\Omega_0 t + \varphi_0 + \Delta \varphi \sin(\omega t)) \\ &= A_0 \left[ \sin(\Omega_0 t + \varphi_0) \cos(\beta \sin(\omega t)) + \cos(\Omega_0 t + \varphi_0) \sin(\beta \sin(\omega t)) \right] \end{aligned}$$

# Аналоговая модуляция

## Фазовая модуляция



# Аналоговая модуляция

## Фазовая модуляция. Спектр

Модулированный сигнал:

$$s(t) = A_0 \left[ \sin \left( \Omega_0 t + \varphi_0 \right) \cos \left( \beta \sin(\omega t) \right) + \cos \left( \Omega_0 t + \varphi_0 \right) \sin \left( \beta \sin(\omega t) \right) \right]$$

где:

$$\beta = \frac{\Delta \varphi}{\varphi_0} \quad - \text{ индекс фазовой модуляции;}$$

$$\Delta \varphi \quad - \text{ девиация фазы.}$$

При  $\beta \ll 1$ :

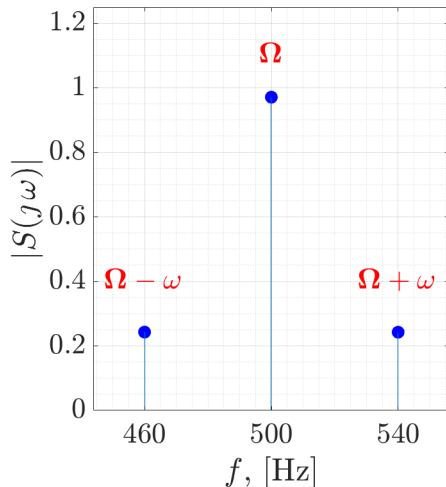
$$s(t) = A_0 \left[ \sin \left( \Omega_0 t + \varphi_0 \right) + \frac{\beta}{2} \sin \left( (\Omega_0 + \omega) t + \varphi_0 \right) - \frac{\beta}{2} \sin \left( (\Omega_0 - \omega) t + \varphi_0 \right) \right]$$

Спектр ФМ совпадает со спектром ЧМ и АМ,  
недостатком ФМ являются скачки фаз

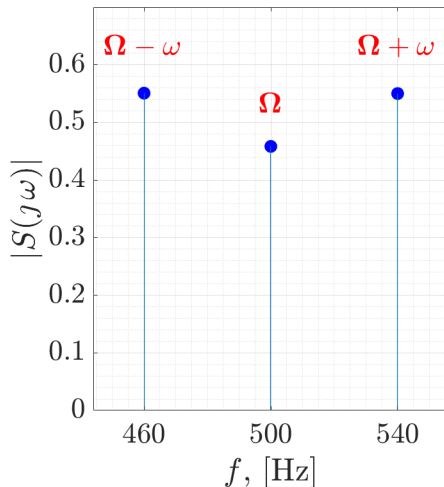
# Аналоговая модуляция

## Фазовая модуляция. Спектр

### AM Spectra



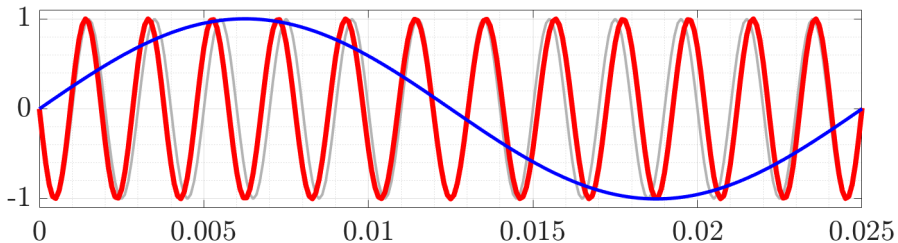
### PM Spectra



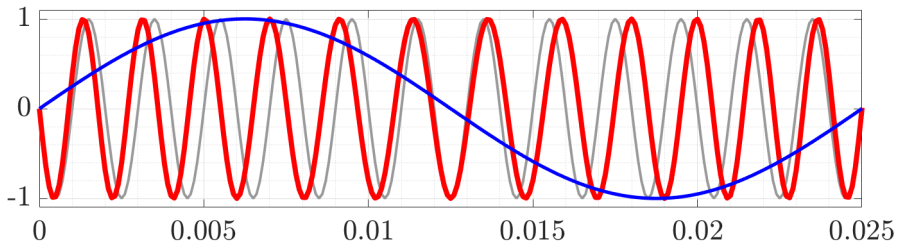
# Аналоговая модуляция

Фазовая модуляция. Девияция фазы  $\Delta\varphi$

$$\Delta\varphi = 45^\circ$$



$$\Delta\varphi = 90^\circ$$



# Аналоговая модуляция

## Квадратурная модуляция (QAM)

### Квадратурная модуляция

Вид модуляции, в котором модулированный сигнал состоит из суммы двух несущих, отстающих друг от друга по фазе на  $\pi/2$ . Модулированный сигнал имеет следующий вид:

$$\begin{aligned} s(t) &= I(t) \cos(\Omega_0 t) + Q(t) \sin(\Omega_0 t) \\ &= I(t) \sin(\Omega_0 t + \pi/2) + Q(t) \sin(\Omega_0 t) \\ &= \Re \left[ \left( I(t) + j Q(t) \right) e^{2\pi j \Omega_0 t} \right] \end{aligned}$$

где  $I(t)$  и  $Q(t)$  - информационные сигналы.



# Аналоговая модуляция

## Демодуляция

### Демодуляция

Процесс восстановления информационного сигнала из полученного высокочастотного (несущего) сигнала.

### Механизмы демодуляции:

- двухполупериодное детектирование;
- синхронное детектирование;
- демодулятор с фазовой автоподстройкой частоты;
- и др.

### Двухполупериодный детектор

Заключается в вычислении модуля полученного сигнала и последующего применения ФВЧ (фильтра высоких частот).

# Аналоговая модуляция

## Демодуляция

### Синхронное детектирование

Заключается в умножении полученного сигнала на сигнал с частотой несущего и последующего применения ФВЧ.

Модулированный сигнал:

$$s(t) = A_0 \left[ 1 + m \sin(\omega t) \right] \sin \left( \Omega_0 t \right)$$

Умножаем на несущий:

$$\begin{aligned} y(t) &= A_0 \left[ 1 + m \sin(\omega t) \right] \sin^2 \left( \Omega_0 t \right) \\ &= \frac{A_0}{2} \left[ 1 + m \sin(\omega t) \right] \left[ 1 - \cos \left( 2 \Omega_0 t \right) \right] \\ &= \frac{A_0}{2} \left[ 1 + m \sin(\omega t) - \cos \left( 2 \Omega_0 t \right) - m \sin(\omega t) \cos \left( 2 \Omega_0 t \right) \right] \end{aligned}$$

После ФВЧ останутся только первые два слагаемых

# Импульсная модуляция

## Импульсная модуляция

Несущий сигнал представляет собой последовательность импульсов.

### Основные виды импульсной модуляции:

- Аналоговая импульсная модуляция:
  - амплитудно-импульсная (АИМ / PAM);
  - широтно-импульсная (ШИМ / PWM);
  - частотно-импульсная (ЧИМ / PFM);
  - фазово-импульсная (ФИМ / PPM);
- Цифровая импульсная модуляция (АЦП):
  - импульсно-кодовая (ИКМ / PM);
  - дельта-импульсная (ДМ / DM /  $\Delta M$ );
  - сигма-дельта (СДМ /  $\Sigma \Delta M$ );

# Импульсная модуляция

## **Амплитудно-импульсная модуляция**

Амплитуда импульсов несущего сигнала модулируется амплитудой информационного сигналом.

## **Широтно-импульсная модуляция**

Длительность импульсов несущего сигнала модулируется амплитудой информационного сигналом.

## **Частотно-импульсная модуляция**

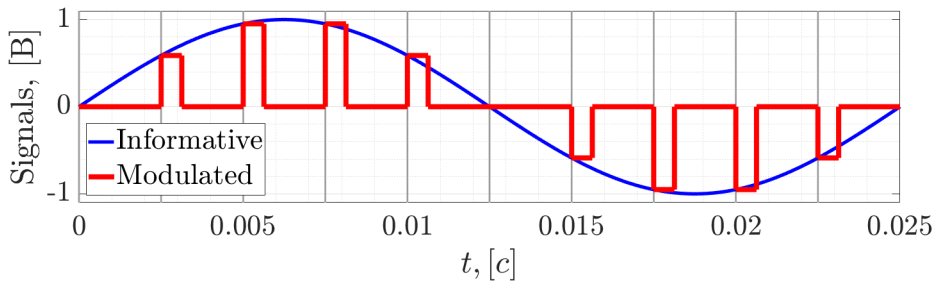
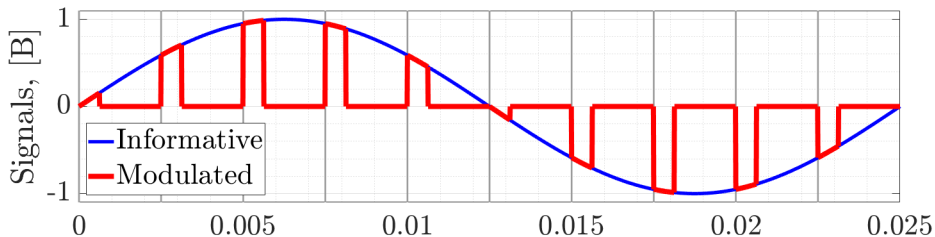
Частота импульсов несущего сигнала модулируется амплитудой информационного сигналом.

## **Фазово-импульсная модуляция**

Фаза (задержка) импульсов несущего сигнала модулируется амплитудой информационного сигналом.

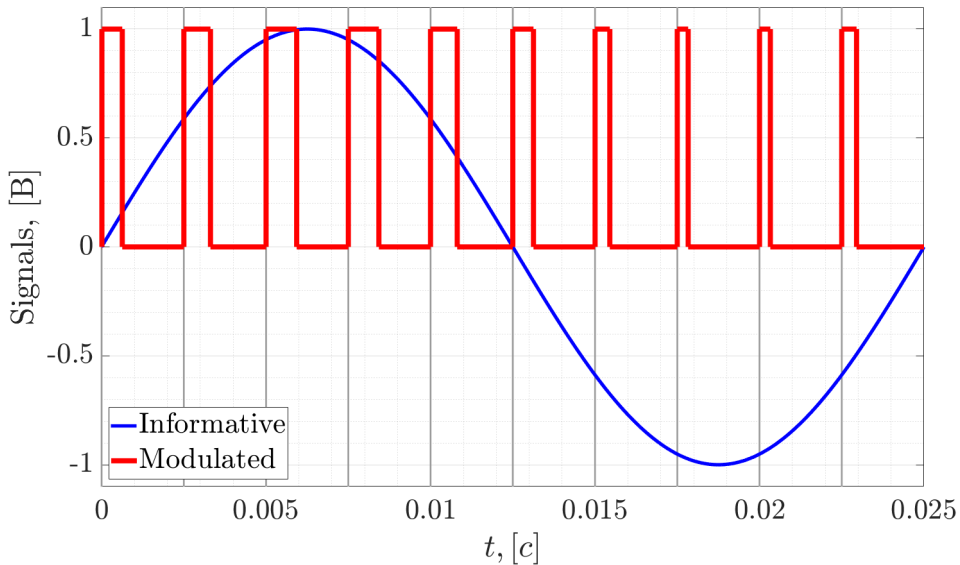
# Импульсная модуляция

## Амплитудно-импульсная модуляция (PAM)



# Импульсная модуляция

## Широтно-импульсная модуляция (PWM)



# Импульсная модуляция

Широтно-импульсная модуляция. Коэффициент заполнения ШИМ

**Коэффициент заполнения ШИМ (скважность):**

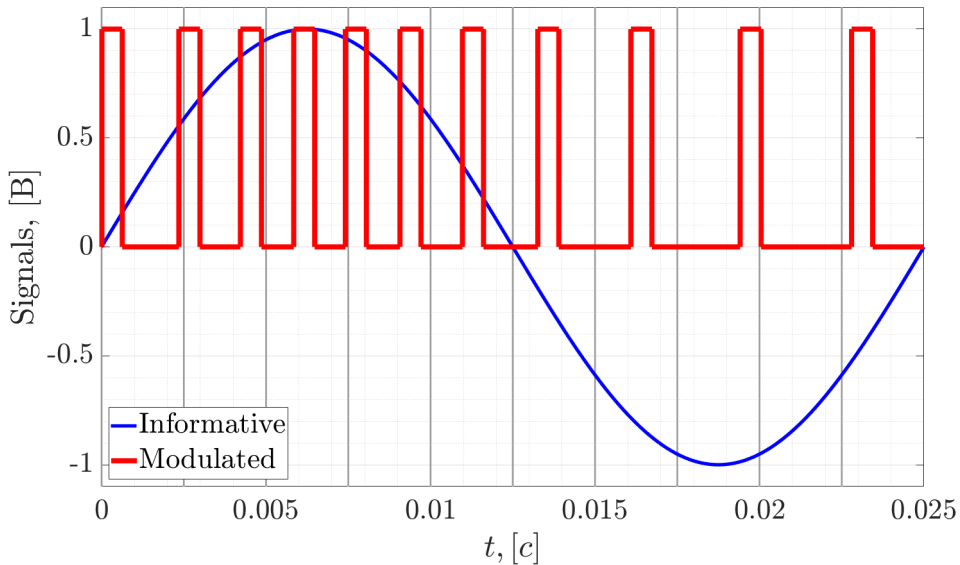
$$S = \frac{T}{\tau} = \frac{1}{D}$$

где:

- $S$  - скважность;
- $D$  - коэффициент заполнения;
- $T$  - период импульсов;
- $\tau$  - длительность импульсов.

# Импульсная модуляция

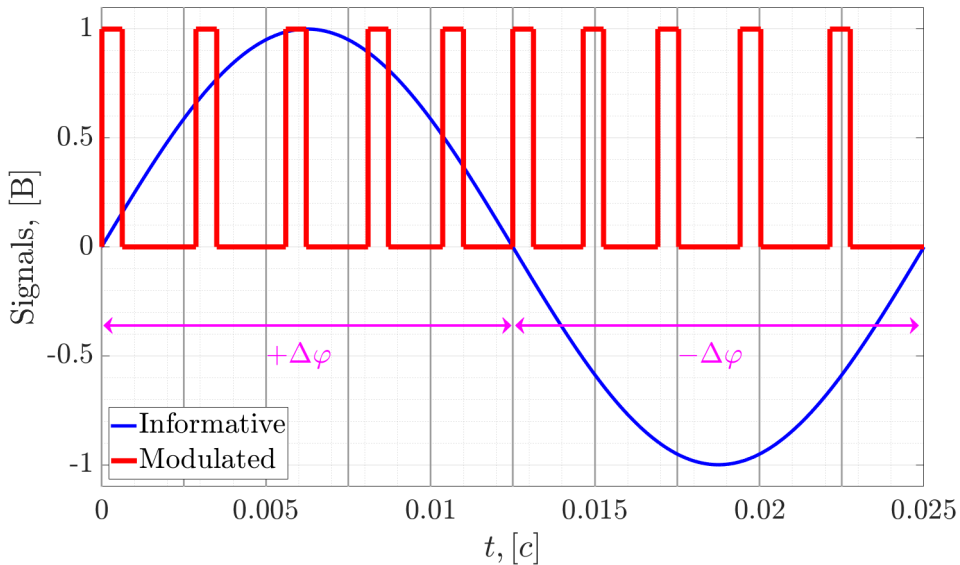
## Частотно-импульсная модуляция (PFM)





# Импульсная модуляция

## Фазово-импульсная модуляция (PPM)



# Цифровая манипуляция

## Shift Keying

### Цифровая манипуляция

“Кодирование” цифрового сигнала аналоговым сигналом.

#### Основные виды цифровой манипуляции:

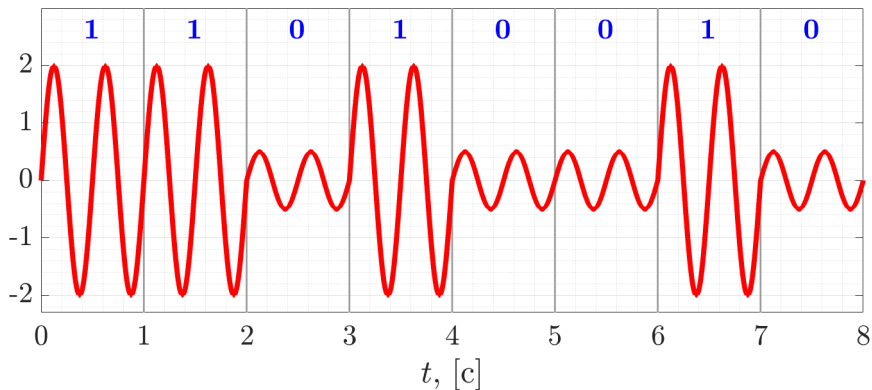
- **Линейные:**
  - амплитудная манипуляция (ASK);
  - квадратурная манипуляция (QAM);
  - фазовая манипуляция (PSK):
    - двоичная (BPSK);
    - квадратурная (QPSK / 4-PSK);
    - восьмеричная (8-PSK);
- **Нелинейные:**
  - частотная манипуляция (FSK):
    - с минимальным сдвигом (MSK);
    - многочастотная (MFSK);

# Цифровая манипуляция

## Амплитудная манипуляция

### Амплитудная манипуляция (Amplitude Shift Keying):

$$s(t) = \begin{cases} A_1 \sin(\Omega t), & s_i(t) = 1 \\ A_2 \sin(\Omega t), & s_i(t) = 0 \end{cases}$$

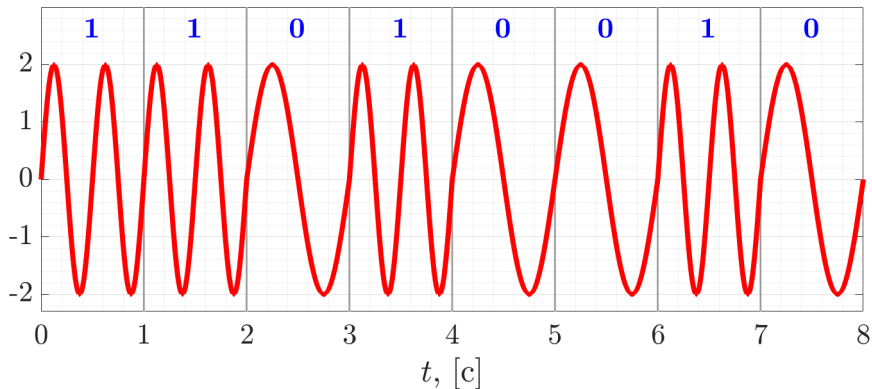


# Цифровая манипуляция

## Частотная манипуляция

### Частотная манипуляция (Frequency Shift Keying):

$$s(t) = \begin{cases} A \sin(\Omega_1 t), & s_i(t) = 1 \\ A \sin(\Omega_2 t), & s_i(t) = 0 \end{cases}$$

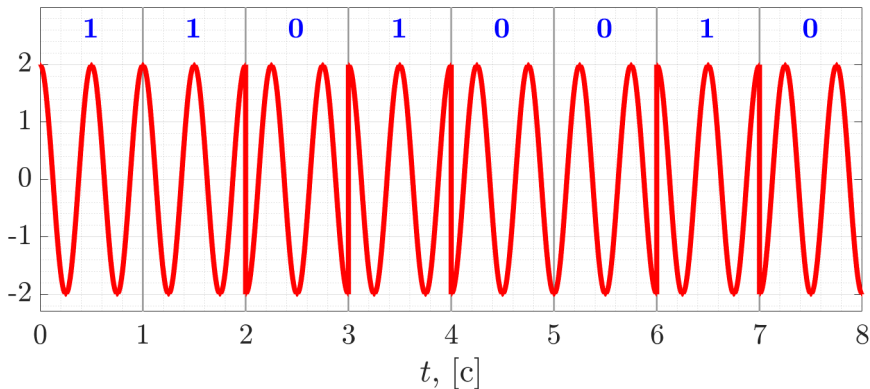


# Цифровая манипуляция

## Двоичная фазовая манипуляция

### Двоичная фазовая манипуляция (Binary Phase Shift Keying):

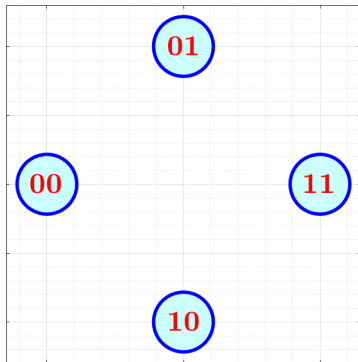
$$s(t) = \begin{cases} A \sin(\Omega t), & s_i(t) = 1 \\ A \sin(\Omega t + \pi/2), & s_i(t) = 0 \end{cases}$$



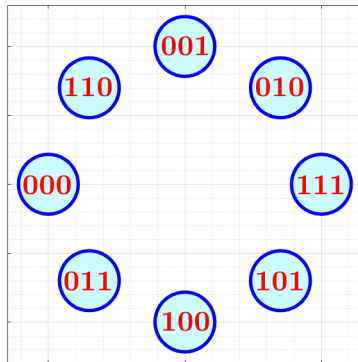
# Цифровая манипуляция

## Виды фазовой манипуляции

QPSK



8PSK



# Цифровая манипуляция

## Особенности

### Амплитудная манипуляция:

- форма “импульса” (единицы или нуля) неидеальна, но это позволяет ограничить спектр модулированного сигнала.

### Частотная манипуляция:

- ресурс полосы частот расходуется неэффективно;
- можно использовать при низком SNR.

### Фазовая манипуляция:

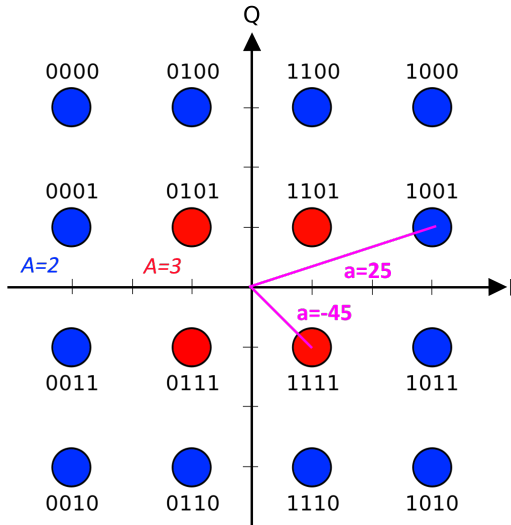
- фаза сигнала меняется скачкообразно;
- чем больше значений, тем технически сложнее декодировать.

# Цифровая манипуляция

## Квадратурная манипуляция (QAM)

### Квадратурная модуляция

Цифровая манипуляция при которой меняется и амплитуда и фаза.





# Физическое кодирование цифровых данных

## Физическое кодирование

Кодирование цифрового сигнала (побитно) аналоговым сигналом для передачи по линии связи.

### Виды физического кодирования:

- потенциальное кодирование;
- импульсное кодирование.

### Требования, предъявляемые к физическому кодированию:

- уменьшение ширины спектра передаваемого сигнала;  
*(выше скорость передачи данных, больше каналов передачи)*
- исключение постоянной составляющей в передаваемом сигнале;  
*(нет утечки энергии)*
- синхронизация передатчика и приемника.  
*(приемник должен знать когда считывать данные)*

# Потенциальное кодирование

## Потенциальное кодирование

“0” и “1” кодируются последовательно соответствующими выбранному методу уровнями напряжения аналогового сигнала.

### Основные потенциальные коды:

- без возвращения к нулю (NRZL);
  - однополярный (unipolar NRZL);
  - биполярный (bipolar NRZL);
- без возвращения к нулю с инверсией (NRZI);
  - с инверсией при единице (NRZM);
  - с инверсией при нуле (NRZS);
- биполярный код с альтернативной инверсией (AMI);
- потенциальный код 2B1Q.

# Потенциальное кодирование

## Кодирование без возвращения к нулю (NRZL)

однополярный:

“0”: 0

“1”:  $+V$

биполярный #1:

“0”:  $-V$

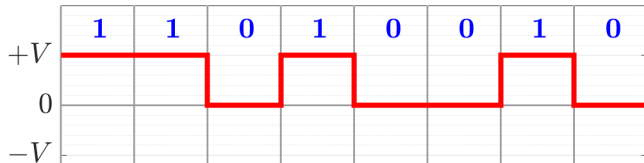
“1”:  $+V$

биполярный #2:

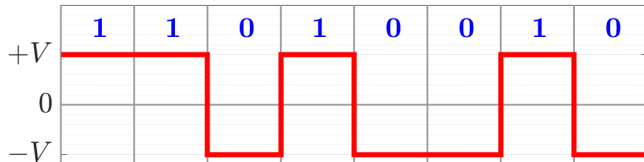
“0”:  $+V$

“1”:  $-V$

### Unipolar



### Bipolar



Биполярный NRZL используется в RS-232  
Уровни напряжения  $\pm V$  задаются в виде интервалов

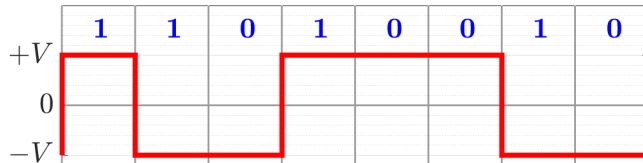
# Потенциальное кодирование

## Кодирование без возвращения к нулю с инверсией (NRZI)

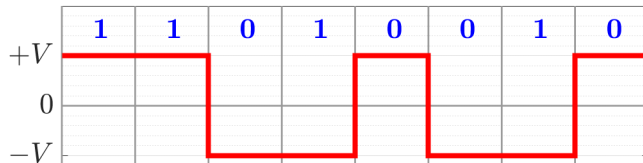
с инверсией при "1":  
смена  $\pm V$  при "1"

с инверсией при "0":  
смена  $\pm V$  при "0"

NRZM (Mark)



NRZS (Space)



NRZI используется в USB

# Потенциальное кодирование

## Особенности методов кодирования без возвращения к нулю

### Достоинства:

- простота реализации;
- хорошая распознаваемость ошибок;
- узкий спектр.

### Недостатки:

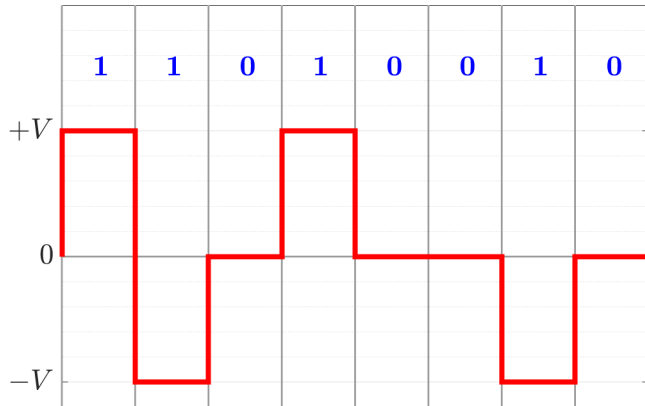
- наличие постоянной составляющей;  
*(особенно в однополярном NRZL)*
- проблемы с синхронизацией при последовательности “0” или “1”.  
*(самосинхронизация отсутствует как таковая)*

# Потенциальное кодирование

## Биполярный код с альтернативной инверсией (AMI)

“0”: 0

“1”: поочередно  $\pm V$



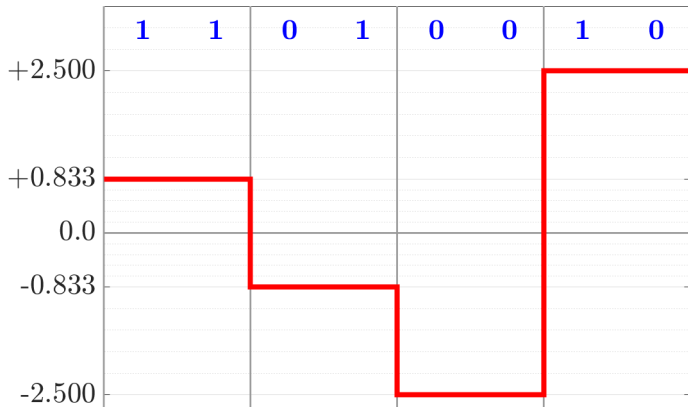
Плохо работает при длинных последовательностях “0”  
Имеет несколько модификаций

# Потенциальное кодирование

## Код 2B1Q

Кодовая таблица:

Код	V, [В]
00	-2.500
01	-0.833
11	+0.833
10	+2.500



За одно изменение сигнала передаем в два раза больше информации  
Требуется более мощный передатчик

## Импульсное кодирование

“0” и “1” цифрового сигнала (побитно) кодируются перепадом уровня напряжения (фронтом или спадом импульса) аналогового сигнала в начале или середине такта (зависит от выбранного метода кодирования).

### Основные импульсные коды:

- с возвращением к нулю (RZ);
- Манчестерский код:
  - по Томасу;
  - по стандарту IEEE 802.4 или 802.3;
- Дифференциальный Манчестерский код (IEEE 802.5).



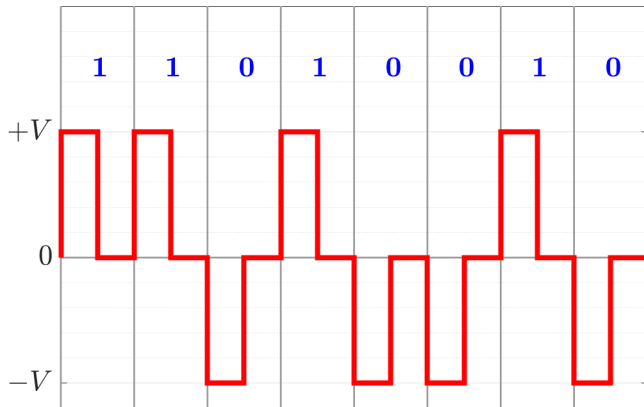
# Импульсное кодирование

## Кодирование с возвращением к нулю (RZ)

“0”: перепад с  $-V$  на 0

“1”: перепад с  $+V$  на 0

**перепады  
в середине такта**



**Есть постоянная составляющая  
Код обладает самосинхронизацией**

# Импульсное кодирование

## Манчестерский код

по Томасу:

“0”: перепад с  $-V$  на  $+V$

“1”: перепад с  $+V$  на  $-V$

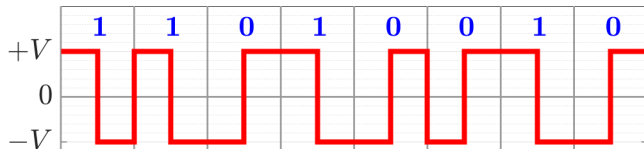
по IEEE 802.4:

“0”: перепад с  $+V$  на  $-V$

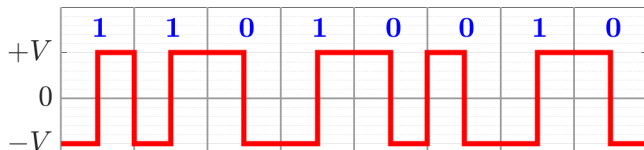
“1”: перепад с  $-V$  на  $+V$

**перепады  
в середине такта**

by Thomas



by IEEE 802.4 / 802.3



Нет постоянной составляющей  
Код обладает самосинхронизацией

# Импульсное кодирование

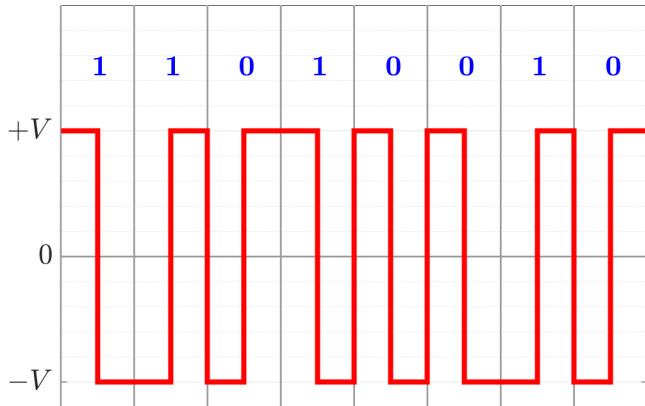
## Дифференциальный Манчестерский код

“0”: есть перепад

“1”: перепада нет

“0” и “1” кодируются  
перепадами  
в начале такта

перепады  
в середине такта  
для синхронизации



Нет постоянной составляющей  
Код обладает самосинхронизацией

# Кодирование данных на логическом уровне

## Кодирование данных на логическом уровне

Преобразование (побитное и/или побайтное) цифровых значений в определенный формат для передачи.

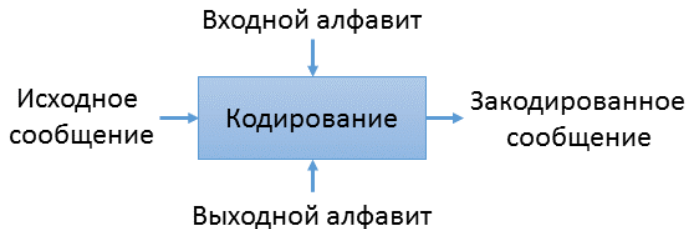
### Виды кодирования цифровых данных:

- сжатие;
- шифрование;
- помехоустойчивое кодирование.

# Кодирование данных на логическом уровне

Кодовая таблица:

$\Sigma_A$	$\Sigma_B$
p	aa
q	ab
r	ba
s	bb



## Алфавит кодирования

Входной алфавит - это набор символов или их комбинаций, который требуется закодировать. Выходной алфавит - тот, которым кодируем.

## Кодовая комбинация

Пара соответствующих элементов входного и выходного алфавитов.

## Помехоустойчивое кодирование

Кодирование, которое позволяет на стороне получателя осуществить проверку на наличие ошибок (возникших при передаче) и их исправить. Как правило, за счет добавления к исходному сообщению некоторой избыточности (проверочные данные).

### Виды помехоустойчивых кодов:

- обнаруживающие ошибки;
- исправляющие ошибки;
- корректирующие (обнаруживают и исправляют ошибки).

# Помехоустойчивое кодирование

## Протоколы коррекции ошибок

### Протоколы коррекции ошибок

Как правило сочетают в себе циклическое кодирование с решающей обратной связью (контрольной суммой). К информационному сообщению добавляется избыточность в виде данных, полученных применением известного алгоритма к исходному сообщению (контрольная сумма). Получатель этот же алгоритм применяет к полученным информационным данным и сравнивает с контрольной суммой, делая вывод о наличии ошибки передачи.

На низком уровне, как правило, используется CRC или простой XOR

# Помехоустойчивое кодирование

## Протоколы коррекции ошибок

### Преимущества использования протоколов коррекции ошибок:

- меньше объем передаваемых данных;  
*(для небольшого объема данных не работает)*
- быстрее чем коррекция данных;  
*(особенно в системах с небольшой вероятностью ошибки)*

### Варианты исправления ошибок:

- запрос на повторную передачу (сетевые протоколы);
- игнорирование ошибки (потокковое видео, игры);  
*(контрольная сумма может вообще не рассчитываться)*
- коррекция ошибок.  
*(в протоколах коррекции ошибок не используется)*



# Помехоустойчивое кодирование

## Классификация и условия декодирования

### Виды кодов:

- **блочные:** *(размер  $n$  закодированного сообщения постоянен)*
  - **равномерные / неравномерные;**  
*(одинаковая или нет длина кодовых комбинаций)*
  - **разделимые:**  
*(место в сообщении проверочных данных определено)*
    - **систематические;** *(строятся сложением по модулю 2)*
    - **несистематические;** *(строятся суммированием)*
  - **неразделимые;**
- **непрерывные.**

### Условия декодирования:

- **уникальность сопоставления символов в алфавитах входа и выхода;**
- **никакая кодовая комбинация не должна быть частью другой.**

# Помехоустойчивое кодирование

## Блочные коды

### Блочный код

Последовательное кодирование фрагментов длиной  $k$  бит в кодовое слово длиной  $n$  бит. Код фиксированной длины:

$$C : \Sigma^k \longrightarrow \Sigma^n$$

где:

- $m \in \Sigma^k$  - исходное сообщение;
- $c \in \Sigma^n$  - кодовое слово, такое что  $c = C(m)$ ;
- $k$  - длина исходного сообщения;
- $n$  - длина блока передаваемого по каналу связи;
- $\Sigma$  - алфавит размерностью  $q$ .

### Скорость кода

Скорость кода  $R = k/n \leq 1$ .  $R > 1$  означает потерю данных, а  $R = 1$  отсутствие помехоустойчивости кода.

# Помехоустойчивое кодирование

Блочные коды. Кодовое расстояние между двумя сообщениями

## Кодовое расстояние

Минимальное расстояние  $d$ , равное количеству позиций в двух блоках отличающихся по значению:

$$d = \min_{\substack{m_1, m_2 \in \Sigma^k \\ m_1 \neq m_2}} \Delta [C(m_1), C(m_2)]$$

## Расстояние Хэмминга:

$$d_{ij} = \sum_{k=1}^p |x_{ik} - x_{jk}|$$

Примеры:

$$d(1011101, 1001001) = 2$$

$$d(2173896, 2333796) = 3$$

$$d(\text{toned}, \text{roses}) = 3$$

# Помехоустойчивое кодирование

## Блочные коды. Коррекция ошибок в блочных кодах

### Корректирующая способность

Корректирующая способность кода показывает сколько ошибок  $t$  в побитной передаче можно гарантированно исправить:

$$t = \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor$$

Требования к корректирующим кодам:

- способность исправлять как можно больше ошибок;  
*(требуется большая избыточность)*
- минимальная избыточность  $n - k$ ;  
*(меньше данных для передачи)*
- простота алгоритма кодирования.

# Помехоустойчивое кодирование

## Линейные коды

Кодирование:

$$\mathbf{c} = C(\mathbf{m}) = \mathbf{G} \mathbf{m}$$

где:

- $\mathbf{m}$  - исходное сообщение;
- $\mathbf{c}$  - закодированное сообщение;
- $\mathbf{G}$  - порождающая матрица.

Передача закодированного сообщения:

$$\mathbf{r} = \mathbf{c} \oplus \mathbf{e}$$

где:

- $\mathbf{r}$  - полученное сообщение;
- $\mathbf{e}$  - математическая модель ошибки.

# Помехоустойчивое кодирование

## Линейные коды

Вычисление синдрома ошибки:

$$\mathbf{s} = \mathbf{H} \mathbf{r}^T$$

где:

$\mathbf{s}$  - синдром ошибки;

$\mathbf{H}$  - проверочная матрица.

Проверочная матрица:

$$\mathbf{H} = \left[ \begin{array}{c|c} -\mathbf{P}^T & \mathbf{I}_{n-k} \end{array} \right]$$

Связь порождающей матрицы с проверочной:

$$\mathbf{G} \mathbf{H}^T = \mathbf{P} - \mathbf{P} = \mathbf{0}$$

Если синдром ошибки  $\mathbf{s} = \mathbf{0}$ , то ошибки нет  
в противном случае определяем ошибочно переданный бит

# Помехоустойчивое кодирование

## Циклические коды

### Циклический код

это код при котором сообщение (кодовое слово)  $\mathbf{m}$ :

$$\mathbf{m} = \{m_0, m_1, \dots, m_{k-1}\}$$

при циклической перестановке также является сообщением

$$\mathbf{m}^* = \{m_{k-1}, m_0, \dots, m_{k-2}\}$$

Представление в виде полинома:

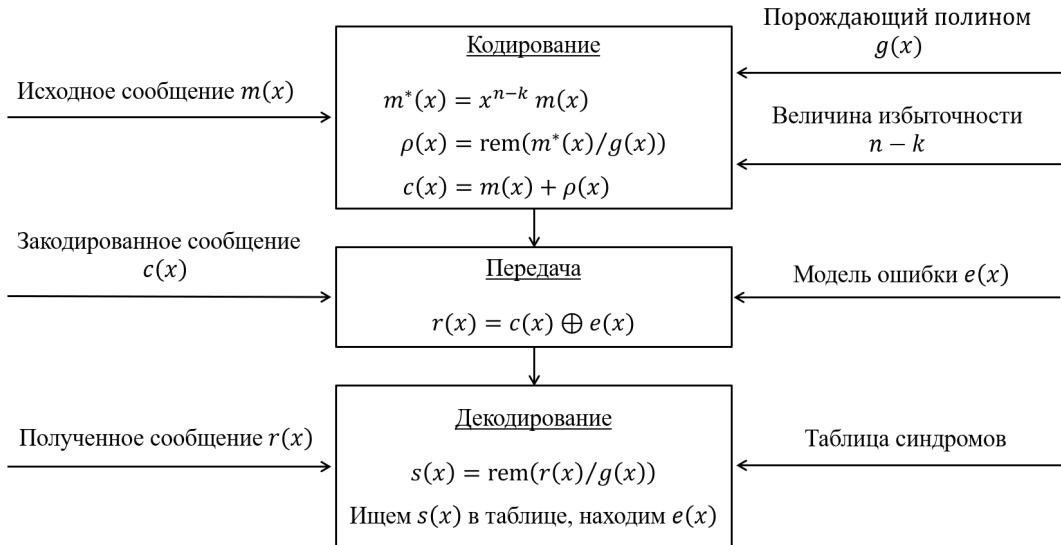
$$m(x) = m_0 + m_1 x + \dots + m_{k-1} x^{k-1}$$

и после перестановки

$$m^*(x) = m_{k-1} + m_0 x + \dots + m_{k-2} x^{k-1}$$

# Помехоустойчивое кодирование

## Циклические коды. Основная идея





# Помехоустойчивое кодирование

## Циклические коды. Порождающий полином

### Свойства порождающего полинома:

- $g(x)$  является делителем полинома  $x^n + 1$ ;
- $g(x)$ , как правило, берут примитивный полином порядка  $n - k$ .  
*(из числа неприводимых полиномов этой степени неприводимый полином делится только на себя и на единицу)*
- $g(x)$ , как правило, выбирается из полиномов CRC.

### Систематические циклические коды (есть избыточность):

$$c(x) = m(x) + \rho(x)$$

### Несистематические циклические коды (нет избыточности):

$$c(x) = m(x) g(x)$$

Далее по тексту “+” - побитный сдвиг, “ $\oplus$ ” побитное “исключающее или”

# Помехоустойчивое кодирование

## Циклические коды. Выбор порождающего полинома

Например для кода с длиной блока  $n = 7$  полином  $x^7 + 1$  можно разложить следующим образом:

$$x^7 + 1 = (x + 1)(x^3 + x^2 + 1)(x^3 + x + 1)$$

Все три сомножителя являются порождающими полиномами, но для синтеза циклического кода  $(7, 4)$  с избыточностью  $n - k = 3$  подходят только два из них:

$$g_1(x) = x^3 + x^2 + 1$$

$$g_2(x) = x^3 + x + 1$$

**Непосредственное вычисление полиномов можно проводить различными аналитическими и численными подходами**

# Помехоустойчивое кодирование

## Циклические коды. Алгоритм кодирования

### Алгоритм кодирования:

1. определяем сообщение  $m(x)$ , которое требуется закодировать;
2. определяем требуемую избыточность  $n - k$ ;
3. выбираем порождающий полином  $g(x)$  с порядком  $n - k$ ;  
*(можно сначала выбрать полином, а потом определить избыточность и удостовериться, что ее достаточно)*
4. осуществляем циклический сдвиг на величину избыточности:

$$m^*(x) = x^{n-k} m(x) = m_{k-1} x^{n-1} + \dots + m_1 x^{n-k+1} + m_0 x^{n-k}$$

5. разделив  $m^*(x)$  на  $g(x)$  получим:

$$m^*(x) = x^{n-k} m(x) = q(x) g(x) + \rho(x)$$

где  $q(x)$  - частное, а  $\rho(x)$  - остаток с порядком меньше  $n - k$ :

$$\rho(x) = \rho_{n-k-1} x^{n-k-1} + \dots + \rho_2 x^2 + \rho_1 x + \rho_0$$

# Помехоустойчивое кодирование

## Циклические коды. Алгоритм кодирования

6. перенося  $\rho(x)$  влево и раскрывая скобки, получим:

$$\begin{aligned}c(x) &= x^{n-k} m(x) + \rho(x) \\ &= m_{k-1} x^{n-1} + \dots + m_1 x^{n-k+1} + m_0 x^{n-k} \\ &\quad + \rho_{n-k-1} x^{n-k-1} + \dots + \rho_1 x + \rho_0\end{aligned}$$

7. закодированное сообщение примет вид:

$$\mathbf{c} = \{m_{k-1}, \dots, m_1, m_0, \rho_{n-k-1}, \dots, \rho_1, \rho_0\}$$

Закодированное сообщение, которое будет передано по каналу связи, формируется добавлением к исходному сообщению остатка от его деления на порождающий полином

## Помехоустойчивое кодирование

Циклические коды. Пример кодирования циклическим кодом (7,4)

Исходное сообщение:

$$m(x) = x^3 + x^2 + 1 = 1101_2, \quad k = 4$$

Получим  $m^*(x)$ :

$$m^*(x) = x^{n-k} m(x) = x^3 (x^3 + x^2 + 1) = x^6 + x^5 + x^3 = 1101000_2$$

Выберем порождающий полином:

$$g(x) = x^3 + x + 1 = 1011_2$$

Разделим  $m^*(x)$  на  $g(x)$  получим:

$$m^*(x) = 1111_2 g(x) + 001_2$$

Что соответствует следующему закодированному сообщению:

$$c(x) = x^6 + x^5 + x^3 + 1 = 1101001_2$$

# Помехоустойчивое кодирование

Циклические коды. Пример кодирования циклическим кодом (7,4)

Бинарный вариант:

1	1	0	1	0	0	0	1	0	1	1
1	0	1	1				1	1	1	1
1	1	0	0							
1	0	1	1							
	1	1	1	0						
	1	0	1	1						
		1	0	1	0					
		1	0	1	1					
		0	0	0	1					

$$\rho(x) = 001_2$$

Полиномиальный вариант:

$x^6$	$+x^5$	$+x^3$					$x^3$	$+x^1$	$+x^0$
$x^6$		$+x^4$	$+x^3$				$x^3$	$+x^2$	$+x^1$
	$x^5$	$+x^4$							
	$x^5$		$+x^3$	$+x^2$					
		$x^4$	$+x^3$	$+x^2$					
		$x^4$		$+x^2$	$+x^1$				
			$x^3$		$+x^1$				
			$x^3$		$+x^1$	$+x^0$			
									$x^0$

$$\rho(x) = x^0$$

# Помехоустойчивое кодирование

## Циклические коды. Алгоритм декодирования

Полученное сообщение:

$$\mathbf{r} = \mathbf{c} \oplus \mathbf{e}$$

или в форме полинома

$$r(x) = c(x) \oplus e(x)$$

**Алгоритм декодирования:**

1. делим  $r(x)$  на  $g(x)$ :

$$r(x) = q(x)g(x) + s(x)$$

при этом ошибку можно выразить как:

$$e(x) = [m(x) + q(x)]g(x) + s(x)$$

2. По таблице определяем позицию ошибочного бита.  
(таблица для каждого кода своя)

# Помехоустойчивое кодирование

Циклические коды. Пример декодирования кода (7, 4)

Переданное сообщение:

$$(x) = x^6 + x^5 + x^3 + 1 = 1101001_2$$

Модель ошибки:

$$e(x) = x^4 = 10000_2$$

Полученное сообщение:

$$r(x) = x^6 + x^5 + x^4 + x^3 + 1 = 1111001_2$$

В результате деления  $r(x)$  на  $g(x)$  получим:

$$r(x) = 1101_2 g(x) + 110_2$$

Что соответствует синдрому ошибки:

$$s(x) = x^2 + x^1 = 110_2$$



# Помехоустойчивое кодирование

Циклические коды. Пример декодирования циклическим кодом (7, 4)

Бинарный вариант:

1	1	1	1	0	0	1	1	0	1	1	
1	0	1	1				1	1	0	1	
-----											
1	0	0	0								
1	0	1	1								
-----											
	0	1	1	0							
				1	1	0	1				
				1	0	1	1				
				-----							
				0	1	1	0				

$$s(x) = 110_2$$

Полиномиальный вариант:

$x^6$	$+x^5$	$+x^4$	$+x^3$					$+x^0$	$x^3$	$+x^1$	$+x^0$
$x^6$		$+x^4$	$+x^3$						-----		
	$x^5$										
	$x^5$					$+x^3$	$+x^2$				
						$x^3$	$+x^2$				
						$x^3$	$+x^2$				
						$x^3$		$+x^1$	$+x^0$		
							$x^2$	$+x^1$			

$$s(x) = x^2 + x^1$$

# Помехоустойчивое кодирование

Циклические коды. Определение ошибочного бита по таблице

Таблица синдромов ошибки:

$e(x)$	$s(x)$	$s(x)_2$
$x^0$	$x^0$	001
$x^1$	$x^1$	010
$x^2$	$x^2$	100
$x^3$	$x^1 + x^0$	011
$x^4$	$x^2 + x^1$	110
$x^5$	$x^2 + x^1 + x^0$	111
$x^6$	$x^2 + x^0$	101

Подобный подход позволяет обнаружить ошибку до  $n - k$  бит, но годен для исправления ошибки только в одном бите  
Таблица синдромов определяется получением остатков от делением каждого полинома ошибки на порождающий полином

# Помехоустойчивое кодирование

Циклические коды. Вычисление таблицы синдромов

$$e(x) = 0000001_2:$$

$$\begin{array}{cccccccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ \hline & & & & & & & & & & & 0 \end{array}$$

$$s_1(x) = 001_2$$

$$e(x) = 0000010_2:$$

$$\begin{array}{cccccccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ \hline & & & & & & & & & & & 0 \end{array}$$

$$s_2(x) = 010_2$$

$$e(x) = 0000100_2:$$

$$\begin{array}{cccccccc|cccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ \hline & & & & & & & & & & & 0 \end{array}$$

$$s_3(x) = 100_2$$

В рассмотренных случаях деление не возможно, т.к. нет четырех разрядов со старшим битом с единицей, т.е. побитное представление ошибки сразу является остатком

# Помехоустойчивое кодирование

## Циклические коды. Вычисление таблицы синдромов

$$e(x) = 0001000_2:$$

$$\begin{array}{r|rrrr|rrrr}
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
 & & & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
 \hline
 & & & 0 & 0 & 1 & 1 & & & & 
 \end{array}$$

$$s_4(x) = 011_2$$

$$e(x) = 0010000_2:$$

$$\begin{array}{r|rrrrrr|rrrr}
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
 & & 1 & 0 & 1 & 1 & & 0 & 0 & 1 & \\
 \hline
 & & 0 & 1 & 1 & 0 & & & & & 
 \end{array}$$

$$s_5(x) = 110_2$$

$$e(x) = 0100000_2:$$

$$\begin{array}{r|rrrrrrr|rrrr}
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
 & 1 & 0 & 1 & 1 & & & 0 & 1 & 0 & 1 \\
 \hline
 & & 1 & 1 & 0 & 0 & & & & & \\
 & & 1 & 0 & 1 & 1 & & & & & \\
 \hline
 & & 0 & 1 & 1 & 1 & & & & & 
 \end{array}$$

$$s_6(x) = 111_2$$

В рассмотренных случаях деление возможно осуществить, соответственно находится остаток

# Помехоустойчивое кодирование

## Циклическая избыточная проверка. Cyclic Redundancy Check (CRC)

### Виды циклической избыточной проверки:

- побитный;  
*(по сути, как обычный циклический код)*
- побайтный;  
*(использует таблицы байтовых кодов)*

### Основные виды кодов:

- CRC-8;
- CRC-16-IBM;  
*(Modbus, USB)*
- CRC-32-IEEE 802.3;  
*(V.42, Ethernet, Serial ATA, MPEG-2, PNG, POSIX cksum)*

**CRC-128 и выше вытеснен криптографическими хэш функциями**

# Помехоустойчивое кодирование

Контрольная сумма через побитное “исключающее или”

Исходное сообщение:

$$m(x) = x^3 + x^2 + 1 = 1101_2$$

Инициализация контрольной суммы:

$$\rho(x) = 1$$

Вычисление контрольной суммы:

$$\rho(x) = \rho(x) \oplus m_1(x) = 1 \oplus 1 = 0$$

$$\rho(x) = \rho(x) \oplus m_2(x) = 0 \oplus 1 = 1$$

$$\rho(x) = \rho(x) \oplus m_3(x) = 1 \oplus 0 = 1$$

$$\rho(x) = \rho(x) \oplus m_4(x) = 1 \oplus 1 = 0$$

Закодированное сообщение:

$$c(x) = x^4 + x^3 + x^1 = 11010_2$$

# Помехоустойчивое кодирование

Контрольная сумма через побитное “исключающее или”

Переданное сообщение:

$$c(x) = x^4 + x^3 + x^1 = 11010_2$$

Модель ошибки:

$$e_1(x) = x^2 = 100_2$$

Полученное сообщение:

$$r_1(x) = c(x) \oplus e_1(x) = x^4 + x^3 + x^2 + x^1 = 11110_2$$

Модель ошибки:

$$e_2(x) = x^2 + x_1 = 110_2$$

Полученное сообщение:

$$r_2(x) = c(x) \oplus e_2(x) = x^4 + x^3 + x^2 = 11100_2$$

# Помехоустойчивое кодирование

Контрольная сумма через побитное “исключающее или”

Полученное сообщение:

$$r(x) = r_1(x) = c(x) \oplus e_1(x) = x^4 + x^3 + x^2 + x^1 = 11110_2$$

Вычисление контрольной суммы:

$$\rho(x) = \rho(x) \oplus r_1(x) = 1 \oplus 1 = 0$$

$$\rho(x) = \rho(x) \oplus r_2(x) = 0 \oplus 1 = 1$$

$$\rho(x) = \rho(x) \oplus r_3(x) = 1 \oplus 1 = 0$$

$$\rho(x) = \rho(x) \oplus r_4(x) = 0 \oplus 1 = 1$$

В результате получаем, что:

$$\rho(x) = 1 \neq r_5(x) = 0$$

Следовательно при передаче произошла ошибка



# Помехоустойчивое кодирование

Контрольная сумма через побитное “исключающее или”

Полученное сообщение:

$$r(x) = r_2(x) = c(x) \oplus e_2(x) = x^4 + x^3 + x^2 = 11100_2$$

Вычисление контрольной суммы:

$$\rho(x) = \rho(x) \oplus r_1(x) = 1 \oplus 1 = 0$$

$$\rho(x) = \rho(x) \oplus r_2(x) = 0 \oplus 1 = 1$$

$$\rho(x) = \rho(x) \oplus r_3(x) = 1 \oplus 1 = 0$$

$$\rho(x) = \rho(x) \oplus r_4(x) = 0 \oplus 0 = 0$$

В результате получаем, что:

$$\rho(x) = 0 \implies r_5(x) = 0$$

Следовательно при передаче ошибки нет, однако это не так.

Это явный пример недостатка контрольной суммы “исключающим или”, что при четном количестве ошибок алгоритм этого не покажет

# Помехоустойчивое кодирование

## Код Рида-Соломона

### Код Рида-Соломона

Недвоичный, один из самых популярных видов кодирования. Добавляет к сообщению  $2t$  символов, позволяя исправлять пакеты ошибок длиной не более  $t$ . Может быть как систематическим, так и несистематическим.

### Алгоритм кодирования (аналогичен циклическому коду):

1.  $T(x) = x^{2t} m(x)$ ;
2. делим  $T(x)$  на  $g(x)$ , получим:  $T(x) = Q(x)g(x) + R(x)$ , где  $R(x)$  - корректирующий код Рида-Соломона;
3. закодированное сообщение примет вид:  $c(x) = T(x) + R(x)$ .

### Алгоритм декодирования:

Процесс декодирования аналогичен циклическим кодам, но определение полинома ошибки требует более сложного подхода.

---

# Телекоммуникационные системы

---

### **Распределенная система обработки данных**

Система, позволяющая организовывать взаимодействие независимых, но связанных между собой ЭВМ.

#### **Категории ресурсов РСОД:**

- аппаратные;
- базы данных;
- управление системами.

#### **Функции РСОД:**

- организация доступа к ресурсам;
- выполнение заданий и интерактивное общение “клиент-сервер”;
- сбор данных о функционировании системы;
- обеспечение надежности и живучести в целом.

# Технологии межсетевого взаимодействия

## Распределенные системы обработки данных

### **РСОД по степени однородности:**

*(по спектру/разнообразию используемого оборудования)*

- полностью неоднородные;
- частично однородные;
- однородные;

### **РСОД по архитектурным особенностям:**

- на основе систем телеобмена;  
*(независимость узлов отсутствует)*
- на основе сетевых технологий.

### **РСОД по степени распределенности:**

- региональные; *(требуют механизмов маршрутизации)*
- локальные. *(полносвязность узлов, ограничение на технологии)*

### Архитектура РСОД

Взаимосвязь логической, физической и программной структур РСОД.

### Физическая структура

Весь комплекс аппаратных средств, ЭВМ (понимается в широком смысле, как некоторое устройство): главные, коммуникационные, интерфейсные, терминальные, административные. Одно устройство может реализовывать несколько логических служб (функций).

### Программная структура

Программные средства как высокого, так и низкого уровня, реализующие работу РСОД. Другими словами, программная структура реализует работу логической структуры РСОД.

# Технологии межсетевых взаимодействия

Распределенные системы обработки данных. Логическая структура РСОД



Элемент логической структуры должен соответствовать (реализовываться)  
элементу физической структуры

Один ЭВМ может реализовывать несколько служб

# Технологии межсетевого взаимодействия

Распределенные системы обработки данных. Программная структура

## Требования предъявляемые к ПО:

- минимизация затрат на изменение программного обеспечения при изменении используемого в сети оборудования;
- изменения в аппаратной части сети не должны отражаться на работе пользователей и выполняемых задачах.

Для удовлетворения этих требований используют различные стандарты



# Технологии межсетевого взаимодействия

## Полносвязные/неполносвязные топологии

### **Сетевая топология**

Описывает принципиальный подход по связи узлов сети между собой. Как правило, подразумевается связь на физическом уровне, т.е. наличие линий связи.

### **Полносвязная топология**

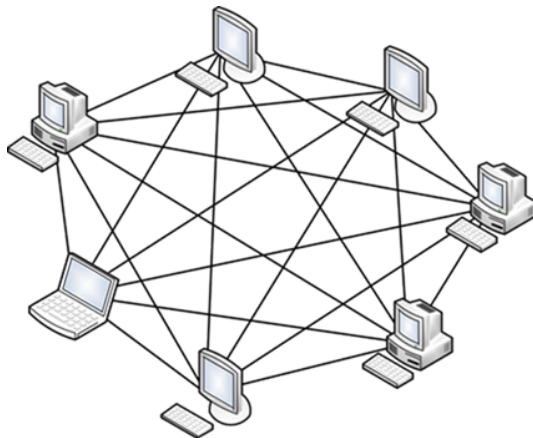
Топология, при использовании которой каждый элемент сети имеет прямую линию связи со всеми остальными элементами сети.

### **Неполносвязные топологии**

Топология, при которой связь элементов исключает обязательное наличие линии связи между всеми элементами сети.

# Технологии межсетевого взаимодействия

## Полносвязная топология



### Особенности:

- экономически не выгодно;
- технически не реализуемо.

# Технологии межсетевое взаимодействие

## Неполносвязные топологии. Топология “Шина”

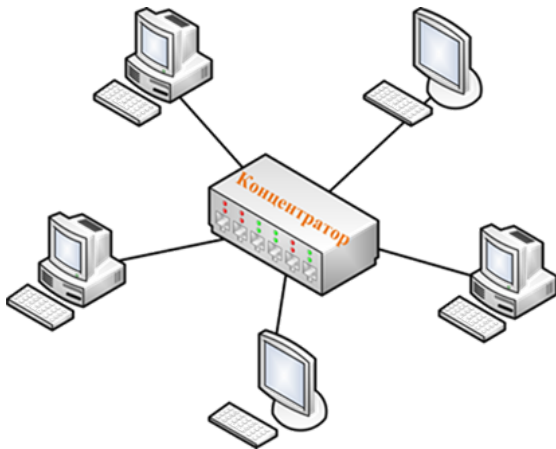


### Особенности:

- разрыв кабеля полностью выводит сеть из строя;
- малая масштабируемость ввиду ограничения на длину кабеля.

# Технологии межсетевого взаимодействия

## Неполносвязные топологии. Топология “Звезда”

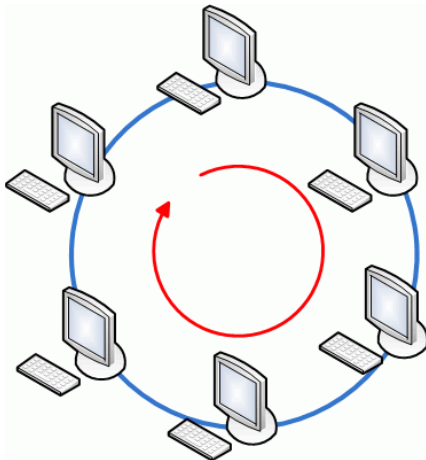


### Особенности:

- возможен большой расход кабеля;
- отказ коммутатора полностью выводит сеть из строя.

# Технологии межсетевых взаимодействия

## Неполносвязные топологии. Топология “Кольцо”

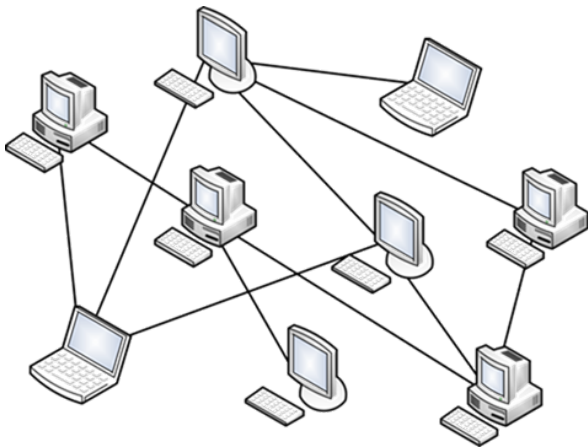


Особенности:

- возможен большой расход кабеля;
- разрыв кабеля полностью выводит сеть из строя.

# Технологии межсетевого взаимодействия

## Неполносвязные топологии. Ячеистая топология

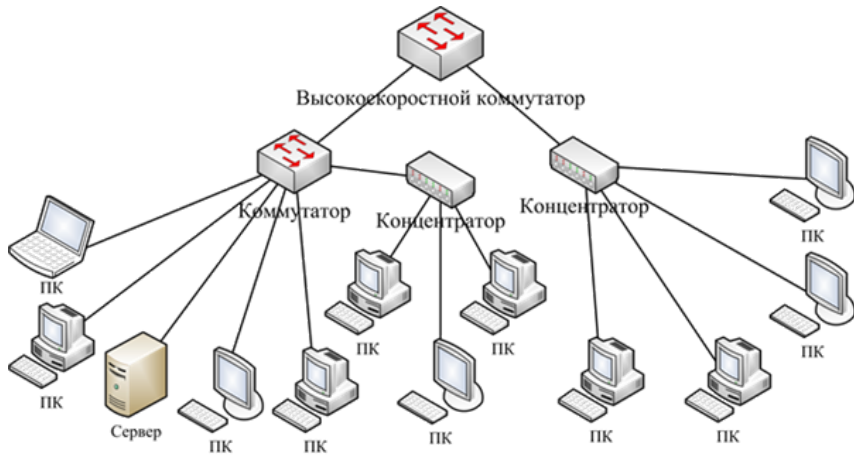


### Особенности:

- возможен большой расход кабеля;
- технически сложно реализуемо.

# Технологии межсетевых взаимодействия

## Неполносвязные топологии. Топология “Дерево”



### Особенности:

- подходит для крупных сетей;
- большая масштабируемость.

### Концепция открытых систем

Эталонная модель, которая используется при реализации конкретной вычислительной сети. Модель OSI (Open System Interconnection) - частный случай, наиболее общий.

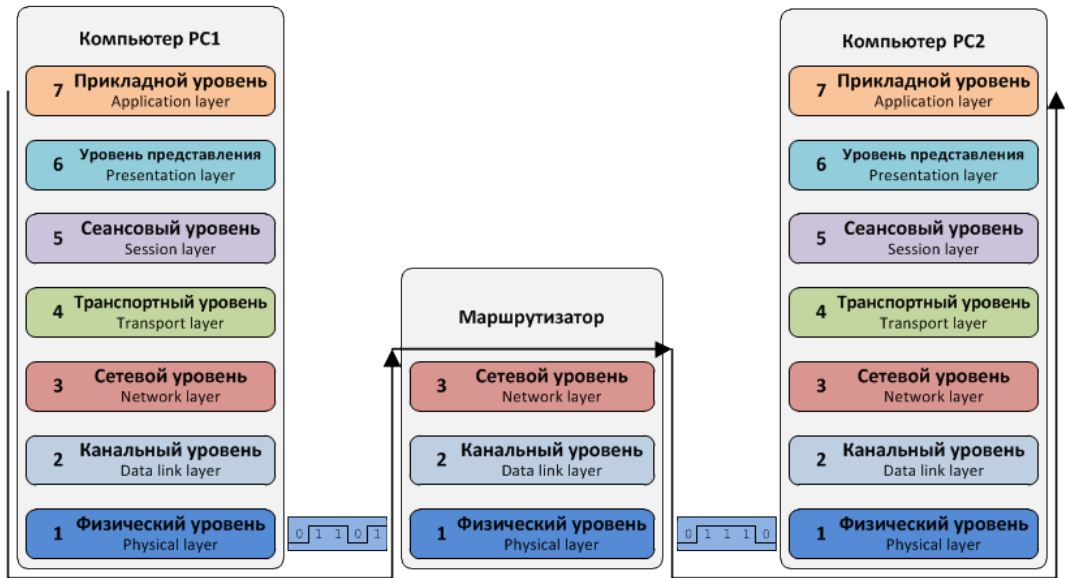
### 7 уровней модели OSI:

- физический (physical);
- канальный (data-link);
- сетевой (network);
- транспортный (transport);
- сеансовый (session);
- представительский (presentation);
- прикладной (application).



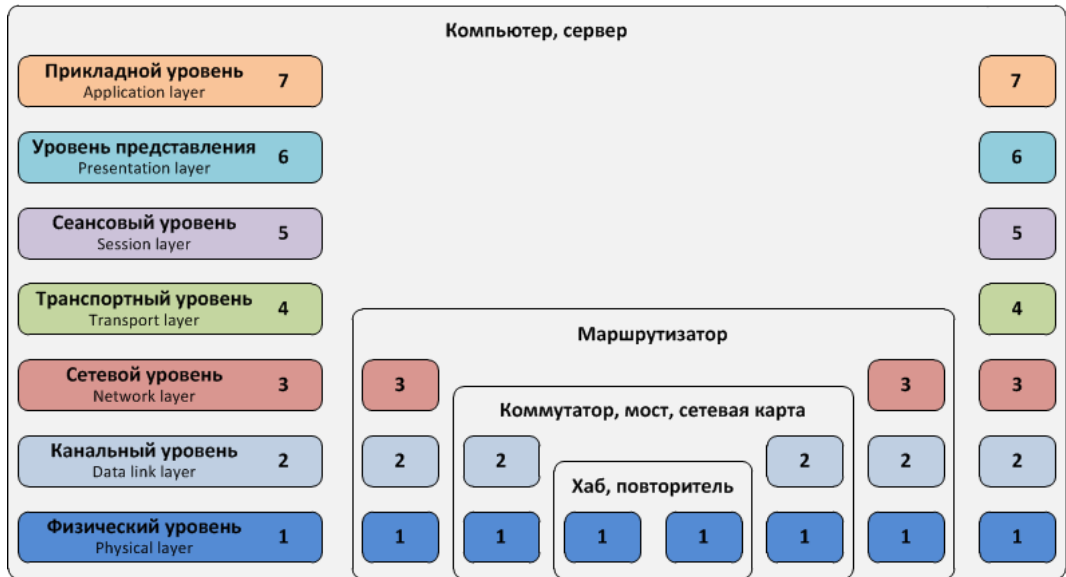
# Технологии межсетевого взаимодействия

## Концепция открытых систем. Модель OSI



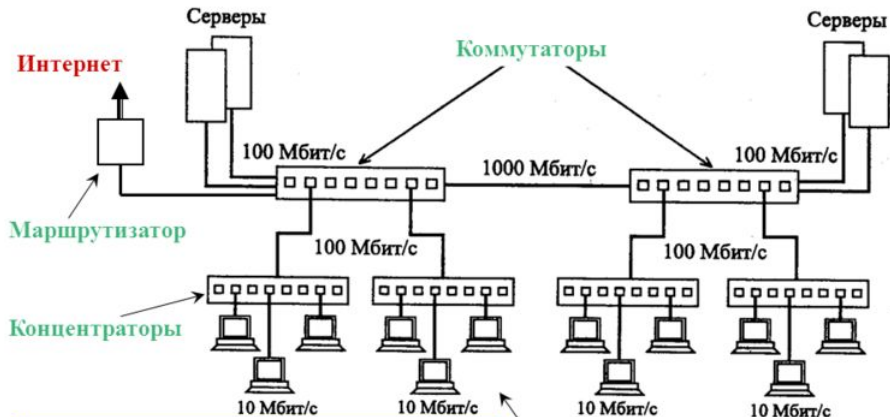
# Технологии межсетевого взаимодействия

## Модель OSI. Аппаратная реализация



# Технологии межсетевого взаимодействия

## Модель OSI. Аппаратная реализация



### Коммутационное оборудование:

1. Концентратор – **Hub**
2. Коммутатор – **Switch**  
(коммутирует по MAC: 00-A0-B1-C1-D3)
3. Маршрутизатор – **Router**  
(маршрутизирует по IP-адреса: 192.168.5.27)

# Технологии межсетевого взаимодействия

## Модель OSI. Объект, Протокол, Интерфейс

### Объект

Часть системы, реализующая какую-то функцию (логическую службу), находящаяся на определенном уровне. Объектом может быть как программное обеспечение, так и контроллер сетевого доступа.

### Протокол

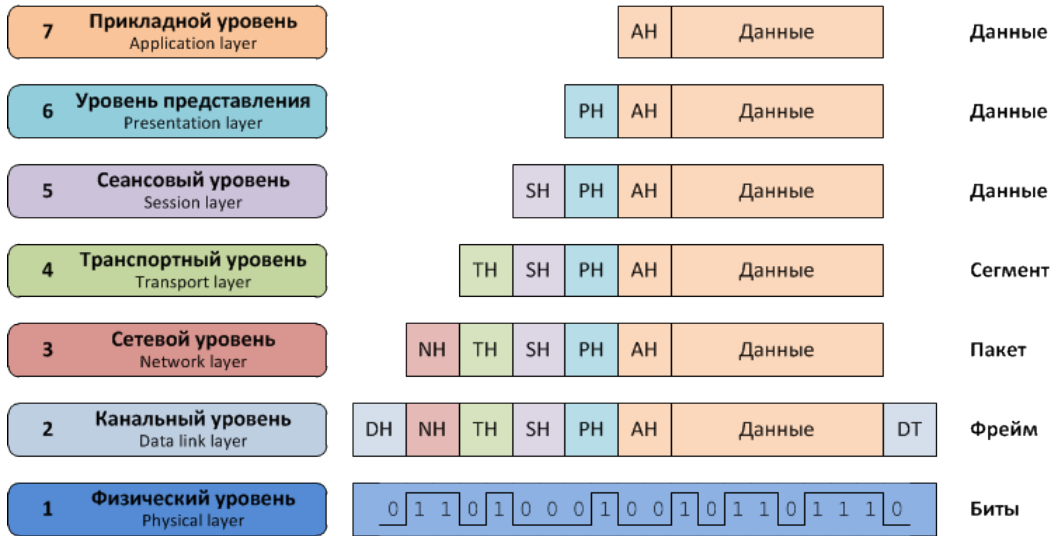
Набор правил взаимодействия объектов одного и того же уровня, реализуемый программно (для сетезависимых уровней - TCP/IP, FTP, и др.) или аппаратно (для сетезависимых уровней - физическое кодирование и др.).

### Интерфейс

Набор правил/преобразований, реализующий взаимодействие между объектами соседних уровней (порты ввода вывода, драйвера).

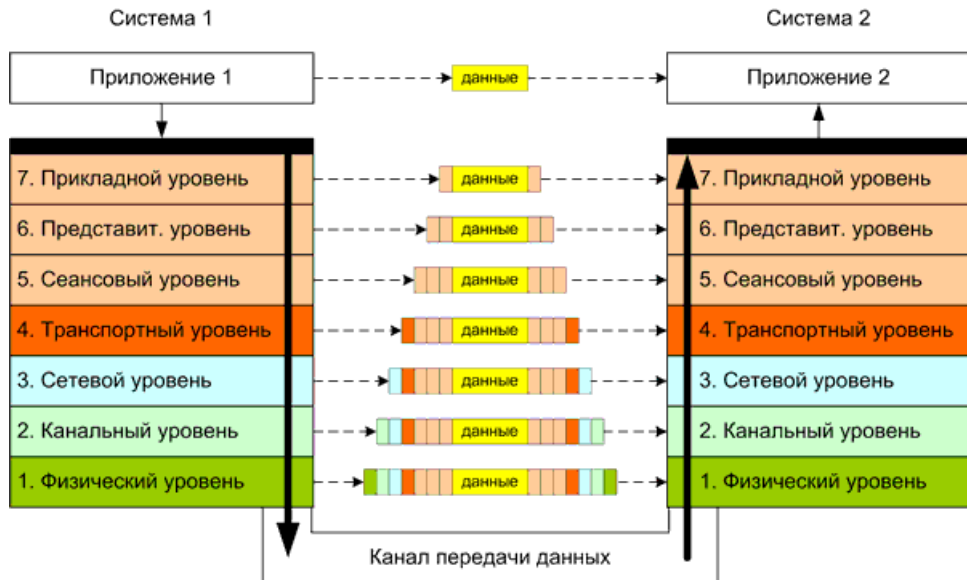
# Технологии межсетевого взаимодействия

## Модель OSI. Данные и их фрагментация



# Технологии межсетевого взаимодействия

## Модель OSI. Данные и их фрагментация



# Технологии межсетевого взаимодействия

## Модель OSI. Данные и их фрагментация

### **Данные (дейтаграмма / датаграмма)**

Набор байт, несущий в себе полезную информацию, необходимую для передачи.

### **Сегмент (фрагмент)**

Часть данных, объем которых может быть технически передан получателю.

### **Пакет**

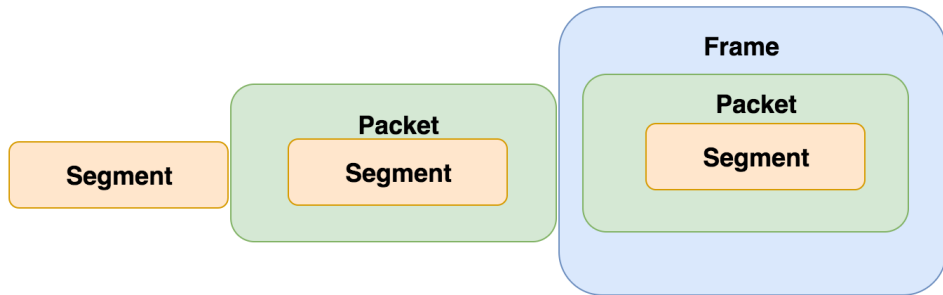
Сегмент данных, сформированный в пакет (добавляется техническая информация) и готовый к передаче получателю по линии связи.

### **Кадр (фрейм)**

Готовая к передаче (физическому кодированию) на физическом уровне, последовательность байт (бит).

# Технологии межсетевого взаимодействия

## Модель OSI. Данные и их фрагментация





### Назначение канального уровня:

Формирует кадры из набора бит/байт, дополняя пакет данных, полученных с сетевого уровня, дополнительными битами (если требуется) и передает на физический уровень. Аппаратно реализуется сетевой картой и драйверами.

### Реализуемые функции:

- проверка доступности линии связи;
- коррекция и контроль ошибок (аппаратный).

# Технологии межсетевого взаимодействия

## Модель OSI. Сетевой уровень (Network Layer)

### Назначение сетевого уровня:

Служит для объединения подсетей (в том числе построенных по разным топологиям) в единую сеть, решает задачу маршрутизации.

### Проблемы выбора наилучшего пути:

- интенсивность трафика через конкретный маршрутизатор;
- пропускная способность;
- надежность передачи через конкретную подсеть.

### Протоколы сетевого уровня:

- сетевые протоколы (routed protocols) для передачи данных;
- протоколы маршрутизации (routing protocols);  
*(в том числе сбор информации о топологиях подсетей) .*
- протоколы разрешения адресов.  
*(определение MAC адреса по IP адресу)*

# Технологии межсетевого взаимодействия

## Модель OSI. Сетезависимые уровни

### **Транспортный уровень:**

Формирование пакетов, разбиение пакетов на кадры, реализация механизмов, если требуется, позволяющих передать данные от отправителя к получателю без потерь (дублирование передаваемых данных, повторная передача).

### **Сеансовый уровень:**

На текущий момент редко реализуется отдельно от прикладного.

### **Представительский уровень:**

Осуществляет преобразование данных (пользовательский пакет), тут же шифрование и дешифрование.

### **Прикладной уровень:**

Работа с данными на уровне приложения.

### **Протоколы без установления соединения**

Connectionless Protocols / Дейтаграммные протоколы. Протоколы, в которых передача данных осуществляется без предварительной связи отправителя и получателя. Данные отправляются “в слепую”.

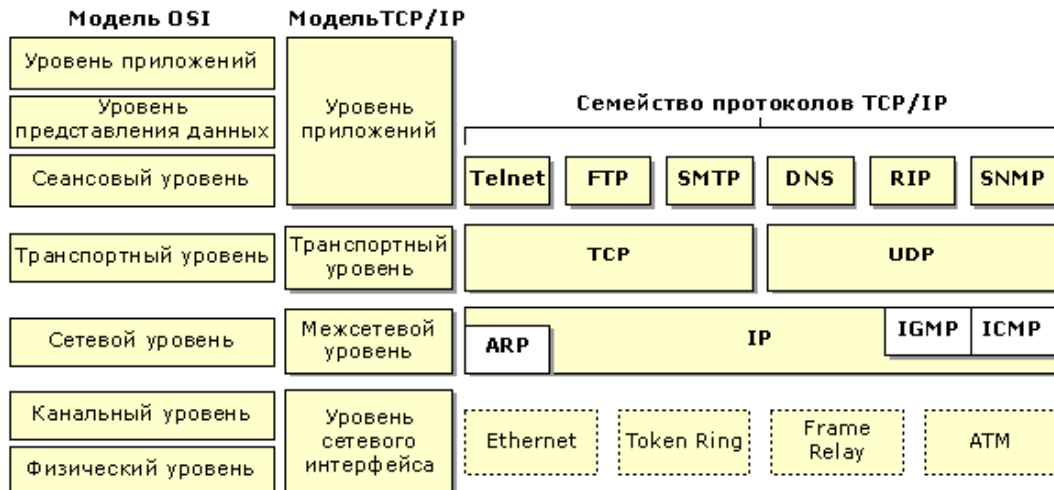
### **Протоколы с предварительным установлением соединения**

Перед отправкой данных отправитель и получатель устанавливают связь и обмениваются технической информацией. Такой тип протоколов часто называют “точка-точка” и используется для:

- взаимной аутентификации (PPP, PPPoE, PPPoA);
- согласование параметров передачи данных (количество бит данных в одном кадре, скорость передачи и др.);
- реализация контроля ошибок путем индексирования (аналог Tracking Number) отправляемых пакетов (кадров) с данными.

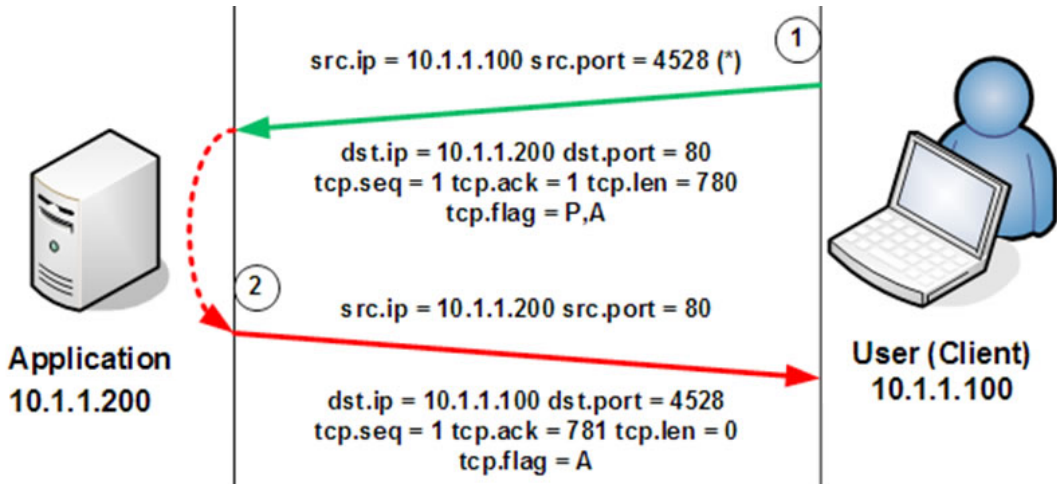
# Протоколы межсетевого взаимодействия

## Стек протоколов TCP/IP



# Протоколы межсетевого взаимодействия

## Взаимодействие узлов через TCP/IP



# Протоколы межсетевого взаимодействия

## Формат пакета протокола TCP

Биты	0-3	4-6	7-15	16-31
0-31	Порт отправителя			Порт получателя
32-63	Порядковый номер			
64-95	Номер подтверждения			
96-128	Длина заголовка	Резерв	Флаги	Размер окна
128-159	Контрольная сумма			Указатель важности
160-191	Опции			
160+	Данные			

# Протоколы межсетевого взаимодействия

## Формат пакета протокола TCP

Порты отправителя и получателя:

- список основных портов см. Wikipedia.

Порядковый номер:

- по сути, идентификатор пакета.
- конкретная роль зависит от флага **SYN**.

Номер подтверждения:

- конкретная роль зависит от флага **ACK**.

Длина заголовка:

- соответствует количеству четырех слов (октетов);
- минимальное - 5, максимальное - 15.

Резерв:

- 3 бита в резерве;



# Протоколы межсетевого взаимодействия

## Формат пакета протокола TCP

Флаги:

- 9 бит;
- **NS** - для ECN (Explicit Congestion Notification);
- **CWR** - для ECN (Explicit Congestion Notification);
- **ECE** - для ECN (Explicit Congestion Notification);
- **URG** - поле “Указатель важности” задействовано;
- **ACK** - поле “Номер подтверждения” задействовано;
- **PSH** - “проталкивает” буфер получателя;
- **RST** - оборвать соединения, сбросить буфер (очистка буфера);
- **SYN** - синхронизация номеров последовательности;
- **FIN** - указывает на завершение соединения.

# Протоколы межсетевого взаимодействия

## Формат пакета протокола TCP

### Размер окна:

- размер данных в байтах, которые можно отправить без подтверждения.

### Указатель важности:

- показывает сколько бит данных являются важными;
- необходима установка флага **URG**.

### Опции:

- может использоваться для расширения протокола;
- можно писать пустую команду;
- можно писать временную метку;

# Протоколы межсетевого взаимодействия

## Формат пакета протокола IPv4

Октет	0	1	2	3
0-3	Версия / Размер заголовка	DCSP / ECN	Полный размер пакета	
4-7	Идентификатор		Флаги / Смещение фрагмента	
8-11	Время жизни	Протокол	Контрольная сумма заголовка	
12-15	IP адрес отправителя			
16-19	IP адрес получателя			
20-23	Опции			
20+ / 24+	Данные			

# Протоколы межсетевого взаимодействия

## Формат пакета протокола IPv4

Версия протокола:

- старшие 4 бита первого октета (для IPv4 - 4).

Размер заголовка:

- младшие 4 бита первого октета, соответствуют количеству октетов;
- т.к. “Опции” не обязательны, то размер заголовка изменяется.

Differentiated Services Code Point (DSCP):

- старшие 6 бит второго октета;
- по сути, приоритет пакета по назначению трафика.

Explicit Congestion Notification (ECN):

- младшие 2 бита второго октета;
- предупреждение о перегрузке сети без потери пакетов.

# Протоколы межсетевого взаимодействия

## Формат пакета протокола IPv4

Полный размер пакета:

- минимальный размер - 20 байт (заголовок без данных);
- максимальный размер - 65 535 байт;
- большой пакет фрагментируется.

Идентификатор:

- порядковый номер фрагмента.

Флаги:

- старшие 3 бита 7 октета;
- осуществляет контроль фрагментации:
  - 0 бит - ставится  $1_2$  только для последнего фрагмента;
  - 1 бит - не фрагментировать;  
*(если кадр не вмещает пакет, он уничтожается)*
  - 2 бит - ставится  $1_2$  если у пакета есть фрагменты.

# Протоколы межсетевого взаимодействия

## Формат пакета протокола IPv4

Смещение фрагмента:

- смещение текущего пакета в байтах от нулевого.

Время жизни:

- количество маршрутизаторов, через которые пакет может пройти.

Протокол:

- какой протокол IP используется (ICMP / IGMP и др.).

IPv6 аналогичен IPv4 с небольшими изменениями и увеличением байт для записи адресов отправителя и получателя

# Протоколы межсетевого взаимодействия

## Протокол UDP. Формат пакета UDP

Биты	0-15	16-31
0-31	Порт отправителя	Порт получателя
32-63	Длина датаграммы	Контрольная сумма
64+	Данные	

### Особенности:

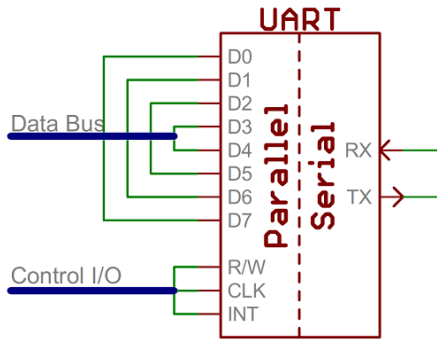
- нет гарантии доставки;
- формат пакета - минимальный для транспортного уровня.

# Стандарты цифровой передачи данных

## Универсальный асинхронный приемопередатчик (UART)

### Универсальный асинхронный приемопередатчик

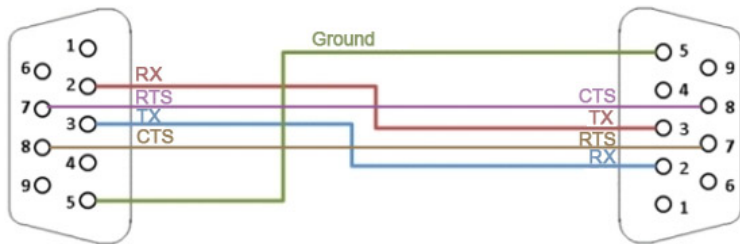
Логическая схема, реализующая преобразование данных в последовательный формат (и наоборот) и их последующую передачу. Через сам UART поток данных напрямую не передается. На его базе строятся стандарты передачи, например RS-232.





# Стандарты цифровой передачи данных

## Универсальный асинхронный приемопередатчик



Контакты:

- **RX / RXD** - линия приема данных;
- **TX / TXD** - линия отправки данных;
- **RTS** - запрос на передачу;
- **CTS** - разрешение на передачу;
- **GND** - земля.

**RTS / CTS можно не использовать**

# Стандарты цифровой передачи данных

Универсальный асинхронный приемопередатчик. Временная диаграмма



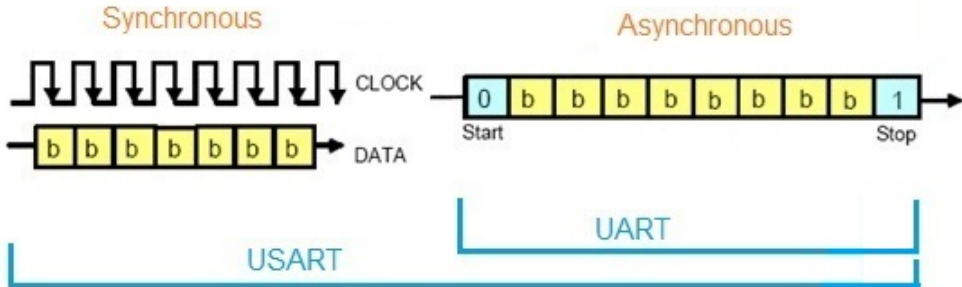
Если передатчик и приемник не настроены на одну скорость работы, то корректное декодирование данных невозможно

# Стандарты цифровой передачи данных

## Универсальный синхронный/асинхронный приемопередатчик (USART)

### Универсальный синхронный/асинхронный приемопередатчик

Логическая схема, похожая UART, но более универсальная, и позволяет передавать данные в синхронном режиме, т.е. требует наличие тактирующей линии, за счет которой и осуществляется синхронизация приемника и передатчика (а не за счет стартовых/стоповых бит, как сделано в UART).



# Стандарты цифровой передачи данных

## Recommended Standard

### **RS-232**

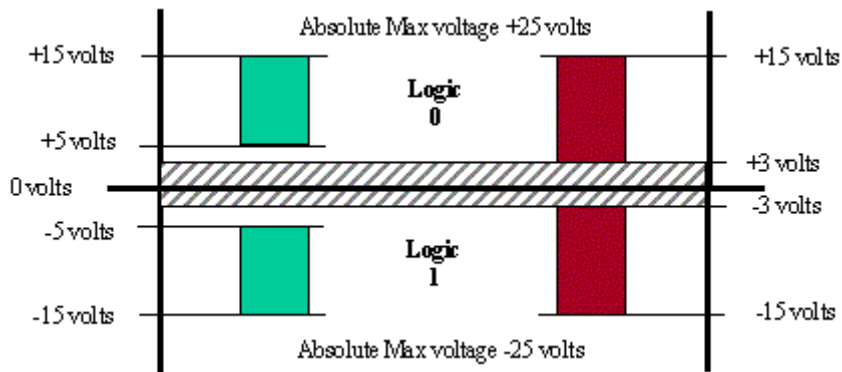
Проводной дуплексный, не дифференциальный, стандарт последовательной передачи данных на базе микросхем UART. Предложен в 1962 году, имеет несколько модификаций. Физическое кодирование осуществляется методом NRZ.

#### **Основные модификации RS-232:**

- RS-422 - дифференциальный;
- RS-423 - не дифференциальный;
- RS-449 - по сути, разъем для RS-422 и RS-423;
- RS-485 - дифференциальный, многоточечный.

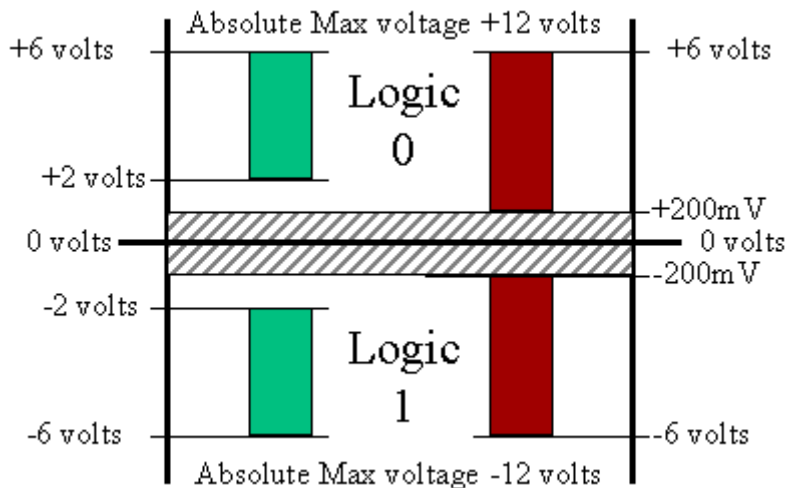
# Стандарты цифровой передачи данных

## Recommended Standard. RS-232



# Стандарты цифровой передачи данных

Recommended Standard. RS-422/RS-485



Для двух стандартов более специфичны  $\pm 7$  Вольт  
Для RS-485 верхний диапазон расширен до +12 Вольт

# Стандарты цифровой передачи данных

## Recommended Standard

	RS-232	RS-423	RS-422	RS-485
Дифференциальный	нет	нет	да	да
Мах передатчиков	1	1	1(5)	32
Мах приемников	1	10	10	32
Режим связи	полудуп. дуплекс	полудуп.	полудуп.	полудуп.
Мах длина кабеля, м	15	1200	1200	1200
Мах скорость при длине кабеля 12 м	20 Кб/с	100 Кб/с	10 Мб/с	35 Мб/с
Мах скорость при длине кабеля 1200 м		1 Кб/с	100 Кб/с	100 Кб/с

# Стандарты цифровой передачи данных

## Recommended Standard

	RS-232	RS-423	RS-422	RS-485
Входная чувствительность приемника, В	$\pm 3$	$\pm 0.2$	$\pm 0.2$	$\pm 0.2$
Диапазон входного сигнала приемника, В	$\pm 15$	$\pm 12$	$\pm 10$	$-7 \dots 12$
Max выходное напряжение, В	$\pm 25$	$\pm 6$	$\pm 6$	$-7 \dots 12$
Min выходное напряжение с нагрузкой, В	$\pm 5$	$\pm 3.6$	$\pm 2.0$	$\pm 1.5$

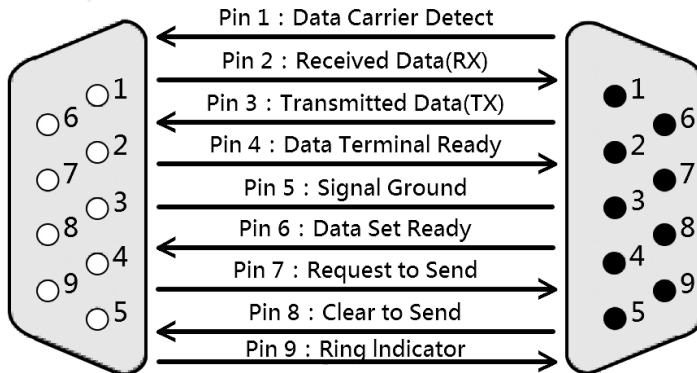


# Стандарты цифровой передачи данных

Recommended Standard. 9 pin разъем

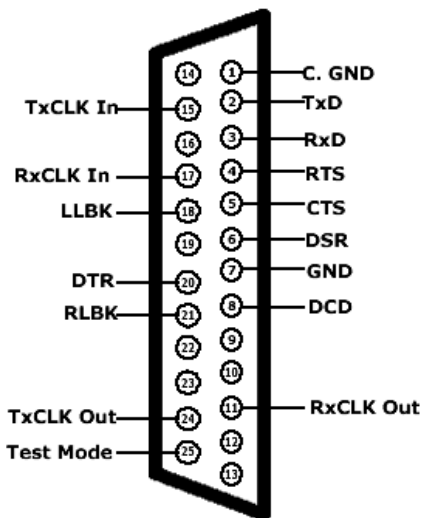
9-pin RS232-c

9-pin RS232-c



# Стандарты цифровой передачи данных

Recommended Standard. 25 pin разъем



**Pin 1 Chassis Ground**

**Pin 2 Transmit Data**

**Pin 3 Receive Data**

**Pin 4 Request to Send**

**Pin 5 Clear to Send**

**Pin 6 Data Set Ready**

**Pin 7 Signal Ground**

**Pin 8 Carrier Detect**

**Pin 11 Receive Clock Out**

**Pin 15 Transmit Clock In**

**Pin 17 Receive Clock In**

**Pin 18 Local Loopback**

**Pin 20 Data Terminal Ready**

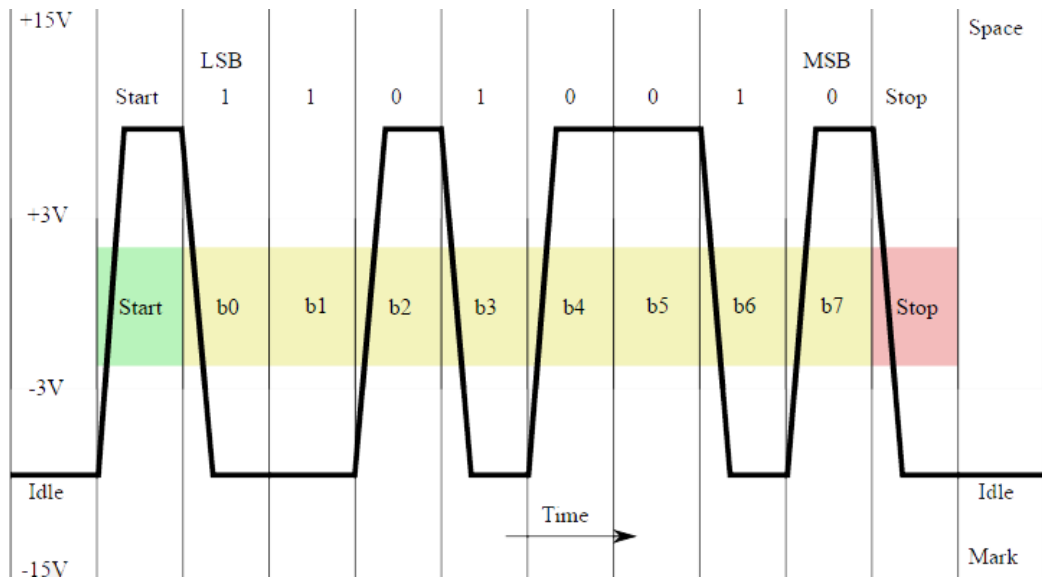
**Pin 21 Remote Loopback**

**Pin 24 Transmit Clock Out**

**Pin 25 Test Mode**

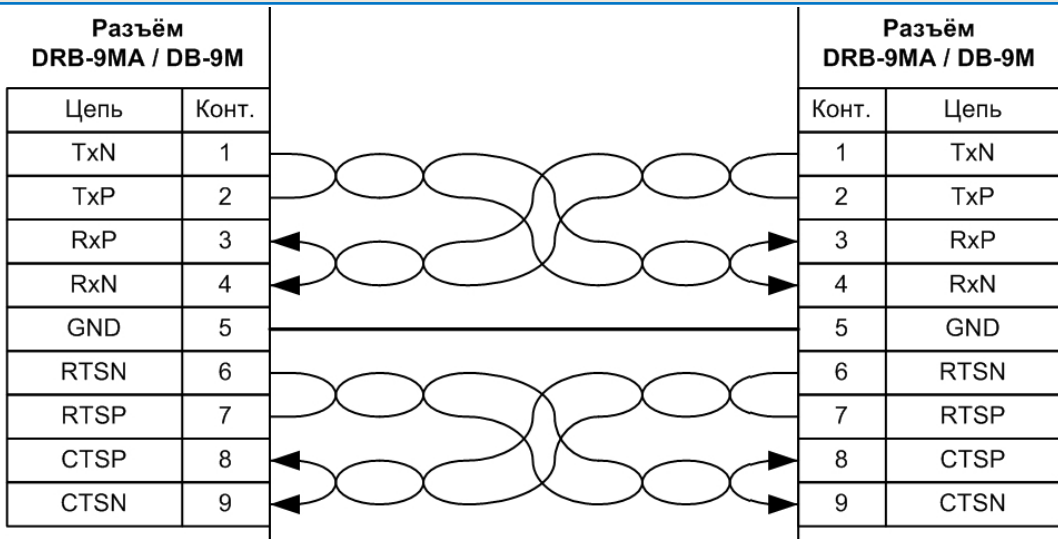
# Стандарты цифровой передачи данных

## RS-232. Временная диаграмма



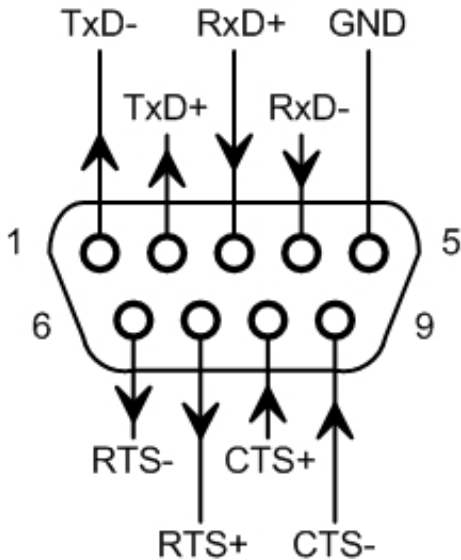
# Стандарты цифровой передачи данных

## RS-422. Дифференциальная схема подключения



# Стандарты цифровой передачи данных

## RS-422. Контакты COM порта

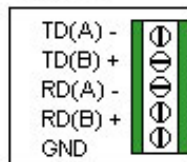


# Стандарты цифровой передачи данных

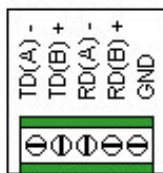
## RS-485. Многоточечное соединение

Connecting 4-wire  
RS485 devices to a  
4-wire RS485 converter

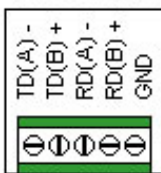
4-wire RS485  
Master



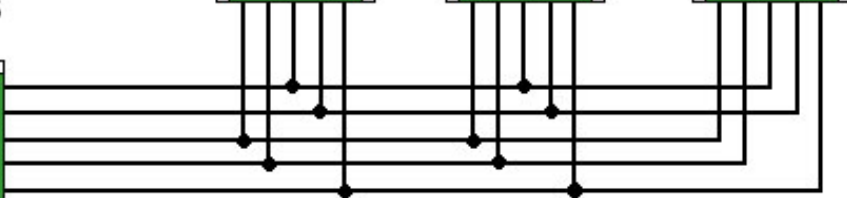
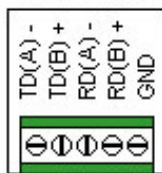
4-Wire  
RS485 Slave



4-Wire  
RS485 Slave



4-Wire  
RS485 Slave



# Стандарты цифровой передачи данных

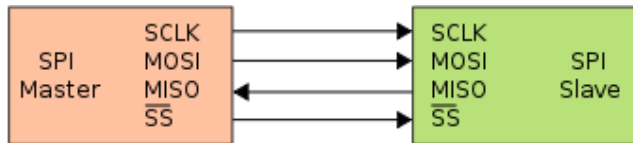
## Serial Peripheral Interface

### SPI

Последовательный периферийный интерфейс (де-факт - стандарт), работающий в синхронном дуплексном режим. Используется для сопряжения микроконтроллеров и периферийных устройств.

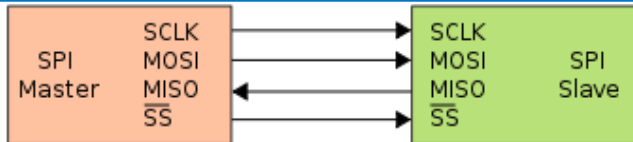
**Ведущий (Master)** - это устройство, инициализирующее связь, отвечает за синхронизацию.

**Ведомый (Slave)** - это подчиненное устройство, работает по команде ведущего.



# Стандарты цифровой передачи данных

## Serial Peripheral Interface



Контакты (на практике наименование может отличаться):

- **SCLK** - тактирующая линия;
- **MOSI** - передача данных от ведущего к ведомому;
- **MISO** - передача данных от ведомого к ведущему;
- **$\overline{SS}$**  - выбор ведомого.

*(низкий уровень напряжения иницирует связь с ведомым)*

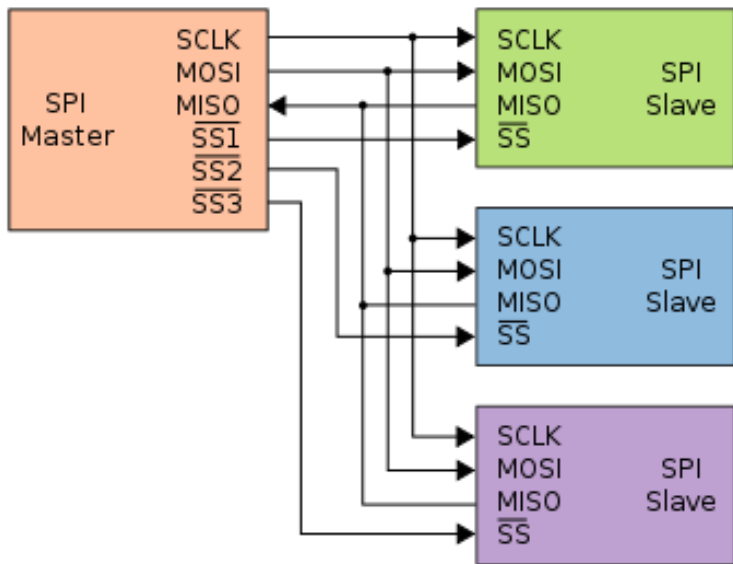
Две топологии подключения ведомых:

- радиальная схема подключения;
- кольцевая схема подключения.



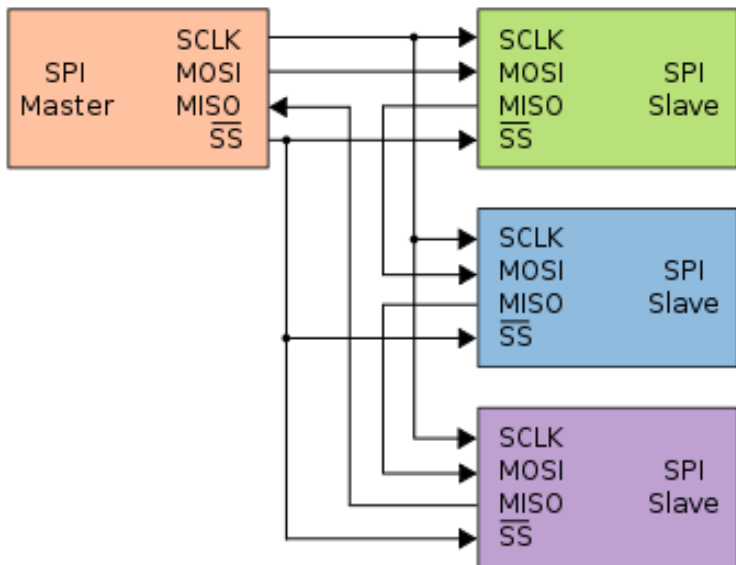
# Стандарты цифровой передачи данных

Serial Peripheral Interface. Радиальная схема подключения



# Стандарты цифровой передачи данных

## Serial Peripheral Interface. Кольцевая схема подключения



# Стандарты цифровой передачи данных

## Serial Peripheral Interface. Четыре режима работы SPI

### Режимы работы SPI:

- режим 0 (CPOL = 0, CPHA = 0);
- режим 1 (CPOL = 0, CPHA = 1);
- режим 2 (CPOL = 1, CPHA = 0);
- режим 3 (CPOL = 1, CPHA = 1).

### Сигнал синхронизации:

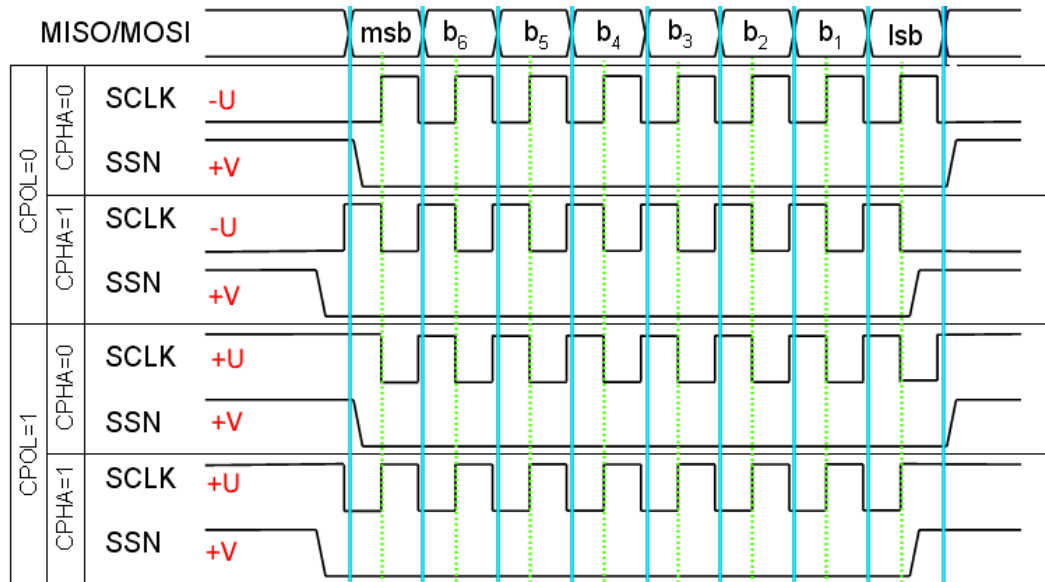
- CPOL = 0 — начинается с низкого уровня;
- CPOL = 1 — начинается с высокого уровня;

### Выборка данных производится:

- CPHA = 0 — по переднему фронту сигнала синхронизации;
- CPHA = 1 — по заднему фронту сигнала синхронизации.

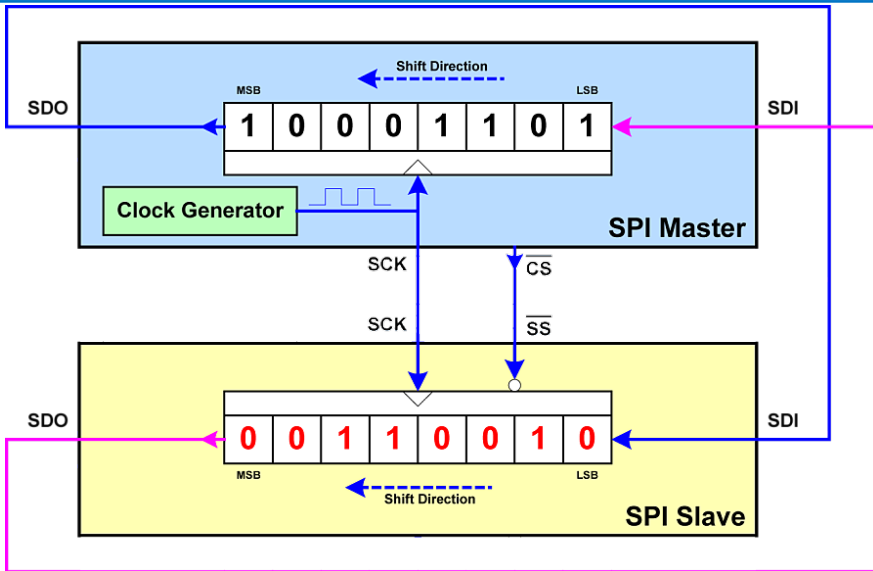
# Стандарты цифровой передачи данных

## Serial Peripheral Interface. Четыре режима работы SPI



# Стандарты цифровой передачи данных

## Serial Peripheral Interface. Механизм передачи данных

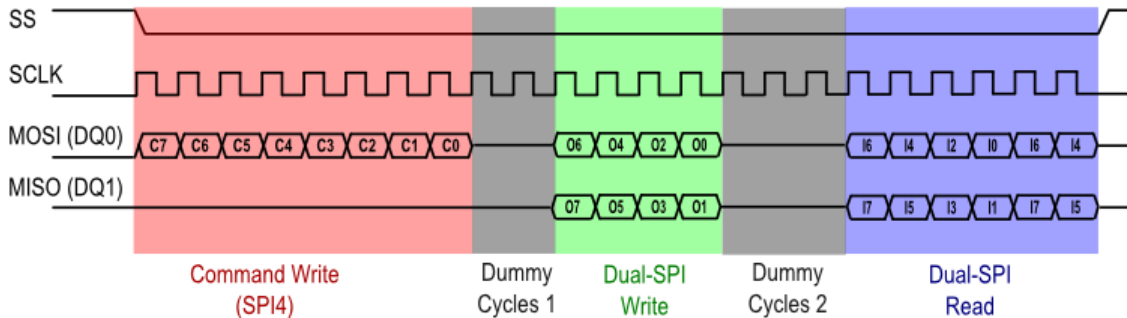


# Стандарты цифровой передачи данных

## Serial Peripheral Interface. Dual SPI

### Dual SPI

Интерфейс с полудуплексным режимом, в котором на один такт синхронизации передаются два бита за счет использования параллельного канала по линии MISO. Команды управления отправляются в обычном режиме, а передача данных осуществляется уже в dual режиме.

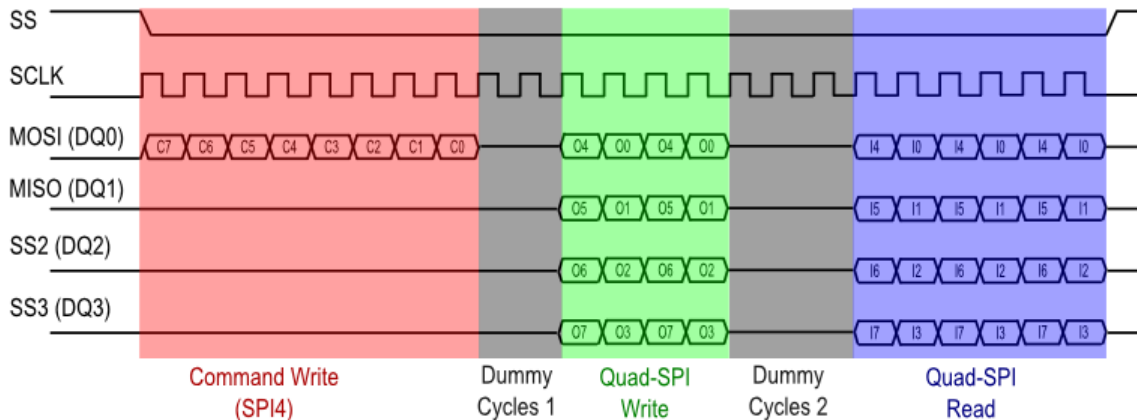


# Стандарты цифровой передачи данных

## Serial Peripheral Interface. Quad SPI

### Quad SPI

Интерфейс с полудуплексным режимом, в котором на один такт синхронизации передаются четыре бита информации, т.е. к Dual SPI добавляются еще два канала связи.



# Стандарты цифровой передачи данных

## Serial Peripheral Interface. Особенности

- низкая масштабируемость;
- более высокая пропускная способность (по сравнению с I<sup>2</sup>C);  
*(до 25 Mb/s)*
- нет ограничения на максимальную частоту передачи;
- возможность формирования не 8-битного пакета;
- относительная простота аппаратной реализации;
- отсутствие возможности подтверждения приема;
- отсутствие протокола обнаружения и коррекции ошибок;
- небольшая дальность передачи;
- нет четко сформулированного стандарта.

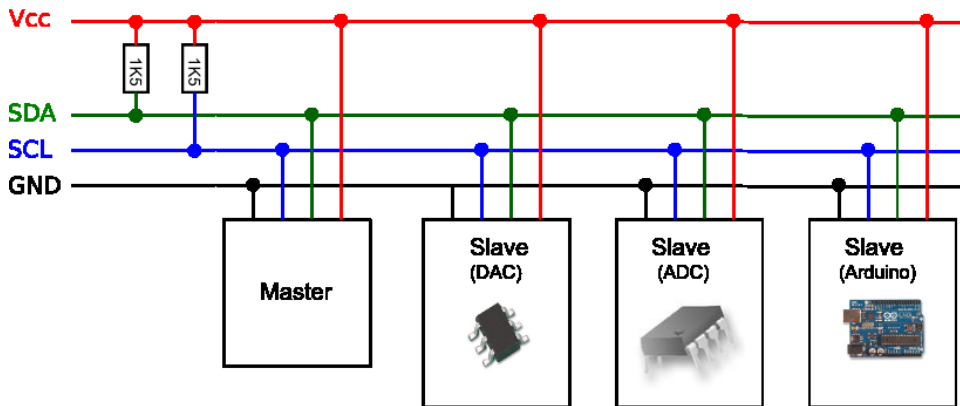


# Стандарты цифровой передачи данных

## Inter-Integrated Circuit

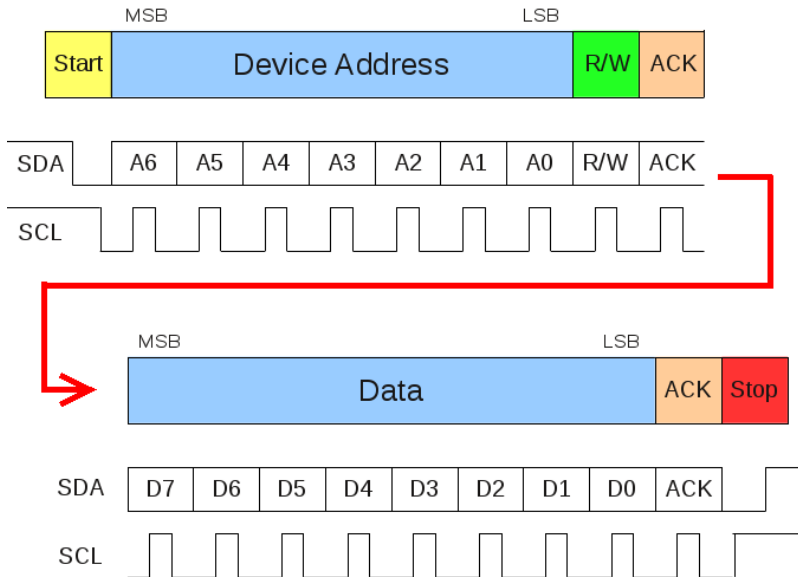
### I<sup>2</sup>C

Inter-Integrated Circuit — последовательная асимметричная шина для связи между интегральными схемами внутри электронных приборов.



# Стандарты цифровой передачи данных

## Inter-Integrated Circuit. Временная диаграмма



# Стандарты цифровой передачи данных

## Inter-Integrated Circuit. Особенности

- высокая масштабируемость;  
*(подключение устройств не требует новых каналов связи)*
- более низкая пропускная способность (по сравнению с SPI);  
*(до 5 Mb/s, а по факту не более 1 Mb/s, стандарт - 400 Kb/s)*
- формирование только 8-битного пакета;
- существует простой контроль готовности передачи;  
*(через флаг АСК)*
- отсутствие протокола обнаружения и коррекции ошибок;
- небольшая дальность передачи;

# Стандарты цифровой передачи данных

## MIPI I3C - SenseWire

### **MIPI I3C - SenseWire**

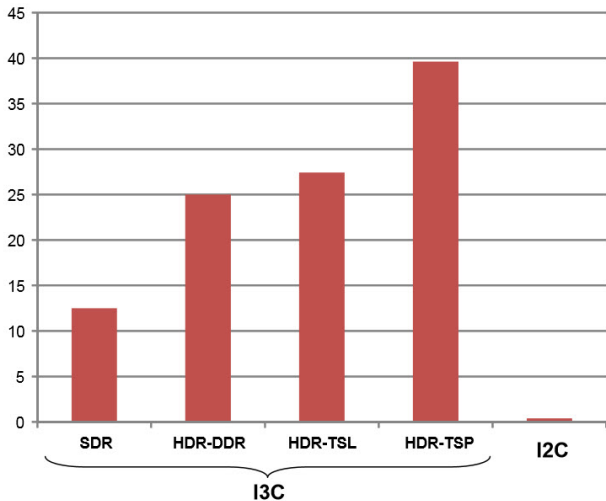
Последовательный, полудуплексный усовершенствованный стандарт I<sup>2</sup>C, появившийся в 2016 году, обеспечивающий скорости передачи на уровне SPI и выше, с наличием механизма контроля передачи и протокола обнаружения ошибок.

Стандарт, также как и I<sup>2</sup>C, ориентирован на системы сбора данных некоторым контроллером с нескольких MEMS (микроэлектромеханических) датчиков с большой скоростью опроса.

# Стандарты цифровой передачи данных

## MIPI I3C - SenseWire

**Raw Bitrate**  
**Mbps for I3C Data Modes (@12.5MHz)**  
**vs I2C (@400KHz)**



# Стандарты цифровой передачи данных

## Universal Serial Bus

### **Universal Serial Bus**

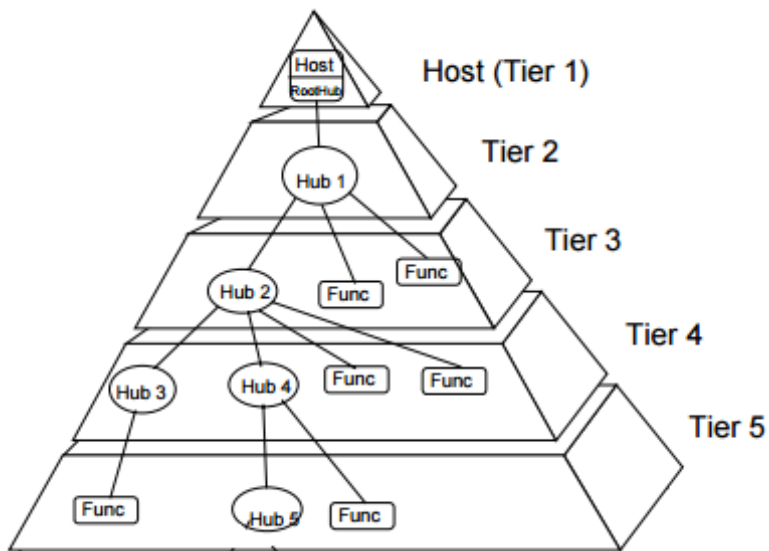
Универсальная последовательная шина, используемая для подключения периферийной аппаратуры к ПЭВМ

### **Принцип обмена данными**

Обмен данными в USB осуществляется транзакциями - неразрывными последовательностями из нескольких пакетов. Ввиду того, что USB позволяет сформировать иерархическую сеть из устройств, в пакете обязательно присутствует адрес устройства. Обмен начинается с отправки токена (token), по сути, команды, показывающей, что нужно сделать. Токен IN или OUT служат для приема или передачи данных.

# Стандарты цифровой передачи данных

## Universal Serial Bus. Физическая топология USB сети



# Стандарты цифровой передачи данных

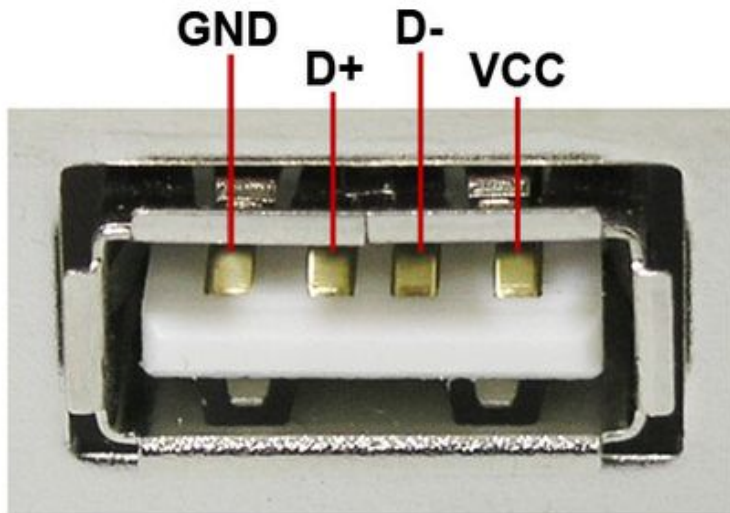
## Universal Serial Bus

	USB 1.1	USB 2.0	USB 3.0	USB 3.1
Пиковая скорость, Мб/с	12	480	5000	10000
Пиковая скорость, МБ/с	1.5	60	625	1250
Мах выходное напряжение, В	2.5	2.5	5	20
Мах выходной ток, А	0.5	0.5	0.9	5
Питание	⇒			⇔
Год появления	1998	2000	2008	2015



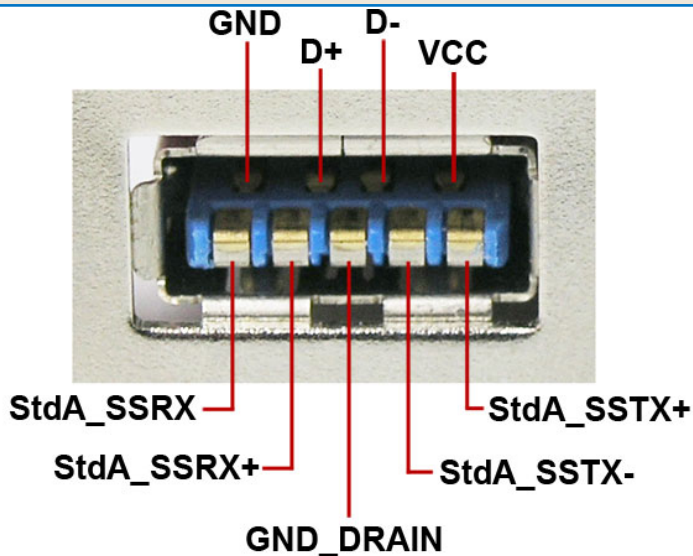
# Стандарты цифровой передачи данных

## USB 1.0/1.1/2.0. Контакты



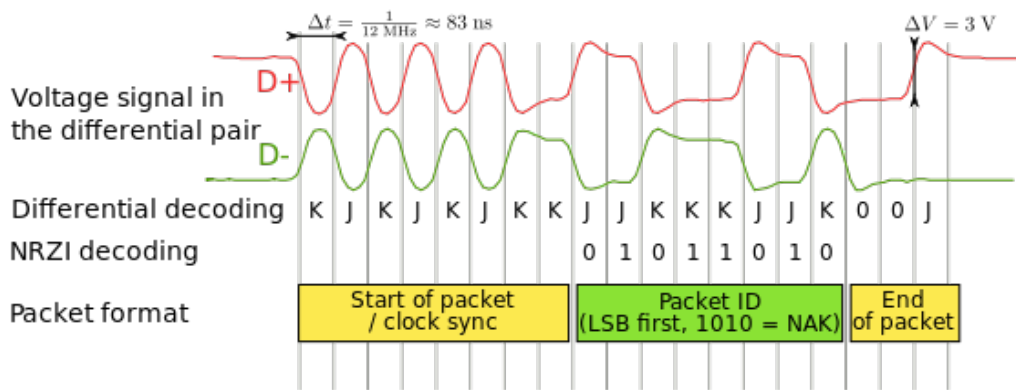
# Стандарты цифровой передачи данных

## USB 3.0/3.1. Контакты



# Стандарты цифровой передачи данных

## Universal Serial Bus. Временная диаграмма



# Стандарты цифровой передачи данных

## General Purpose Interface Bus

### **IEEE 488**

Шина общего назначения. Используется для подключения измерительных устройств. Содержит 8 линий для передачи данных.

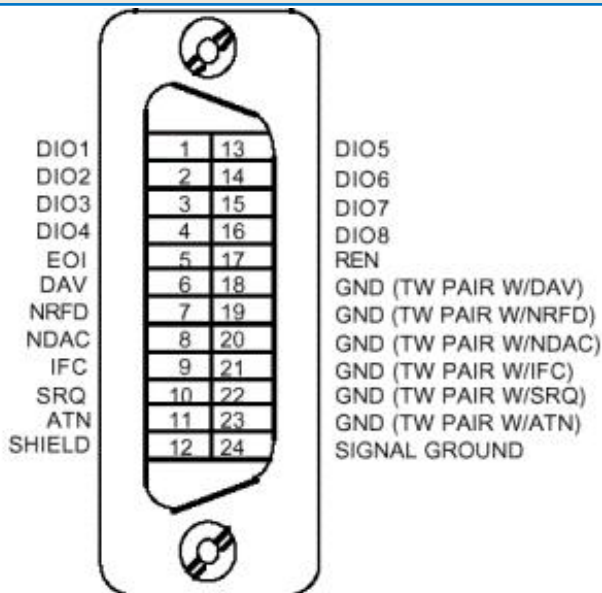
### **IEEE 488.1**

Модификация интерфейса, созданная компанией National Instruments. Позволяет передавать данные со скоростью 8 Мб/с.

По факту, стандарт вытеснен USB и Ethernet

# Стандарты цифровой передачи данных

## General Purpose Interface Bus. Разъем

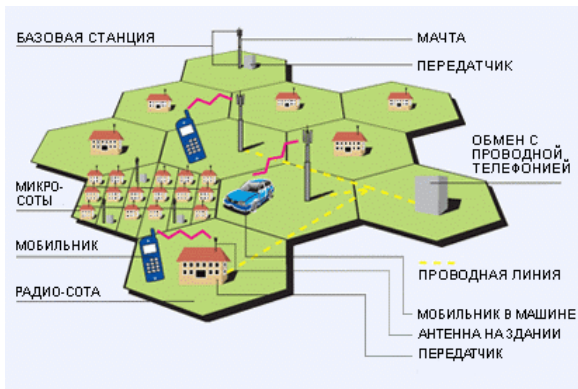


# Стандарты беспроводной передачи данных

## Сотовая связь

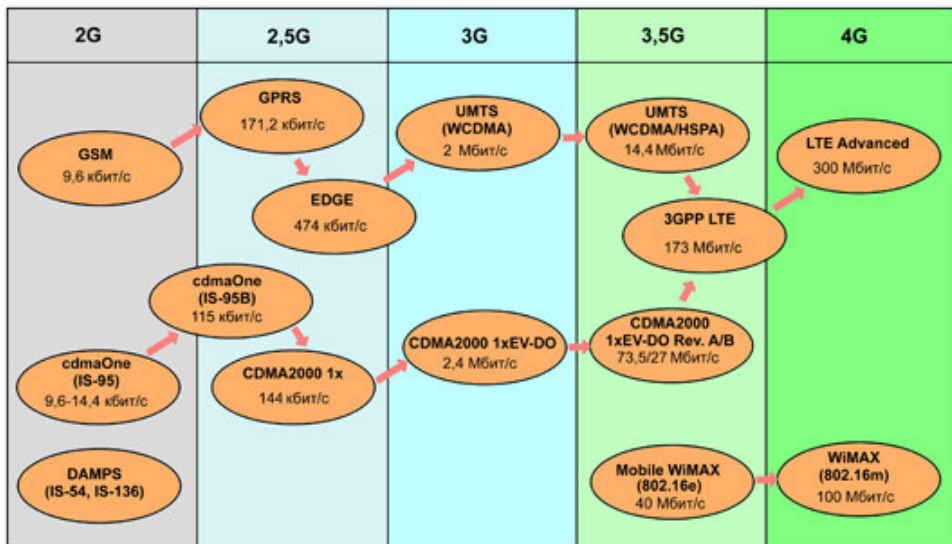
### Сотовая связь

Вид связи по радиоканалу, в котором зона покрытия разделяется на соты, в каждой из которых находится вышка с приемопередатчиком. Сами вышки объединяются в сеть и соединяются с центральной базой по высокоскоростным кабельным (как правило) каналам.



# Стандарты беспроводной передачи данных

## Сотовая связь



## Стандарты беспроводной передачи данных



**Bluetooth®**





# Стандарты беспроводной передачи данных

## Wi-Fi

### Wi-Fi (Wireless Fidelity)

Беспроводная технология организации локальных сетей, базирующаяся на группе стандартов IEEE серии 802.11. Может строиться как с использованием точки доступа (централизованной и децентрализованной), так и по механизму точка-точка.

При использовании точки доступа, она каждые 100 мс передает свой идентификатор SSID с помощью специальных сигналов. Пользователи по данному SSID согласуют с точкой доступа формат обмена данными при соединении.

На физическом уровне используется радиоканал с различными механизмами модуляции и цифрового кодирования (как правило, различные варианты BPSK, QPSK, и QAM) .

# Стандарты беспроводной передачи данных

## Сравнение стандартов Wi-Fi

	802.11	802.11b	802.11a	802.11g	802.11n	802.11ac	802.11ax
Год ратификации	1997	1999	1999	2003	2009	2014	2017-2019
Рабочая частота	2.4 GHz/IR	2.4 GHz	5 GHz	2.4 GHz	2.4/5 GHz	5 GHz	2.4/5 GHz
Частотные каналы	20 MHz	20 MHz	20 MHz	20 MHz	20/40 MHz	20/40/80/160 MHz	20/40/80/160 MHz
Пиковая физическая скорость (PHY)	2 Mbps	11 Mbps	54 Mbps	54 Mbps	600 Mbps	6.8 Gbps	10 Gbps
Модуляция	DSSS, FHSS	DSSS, CCK	OFDM	OFDM	OFDM	OFDM	OFDM, OFDMA
Макс тип и скорость кодирования	DQPSK	CCK	64-QAM, 3/4	64-QAM, 3/4	64-QAM, 5/6	256-QAM, 5/6	1024-QAM, 5/6

Расстояние от точки доступа, физические преграды прохождению радиосигнала влияют на надежность передачи, физическую и эффективную пропускную способность

В среднем радиус действия точки доступа не превышает 50-100 метров

# Стандарты беспроводной передачи данных

## WiMAX

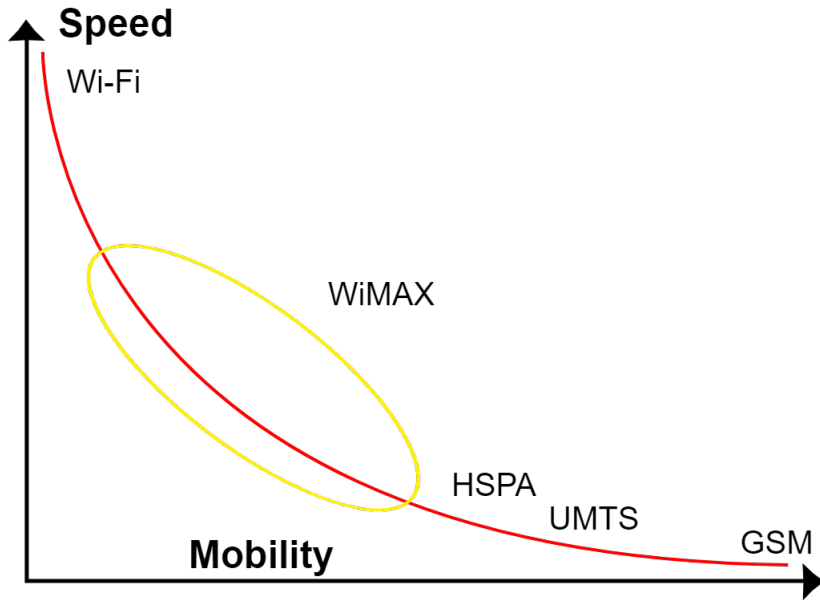
### **WiMAX (Worldwide Interoperability for Microwave Access)**

Беспроводная технология организации универсальных сетей с широкой областью покрытия (десятки километров), базирующаяся на группе стандартов IEEE серии 802.16.

По своей сути, WiMAX работает на схожей идее с сотовыми сетями (GSM, GPRS, EDGE, LTE и д.р.), но в большей степени ориентированы на стационарных пользователей с приоритетом на передачу цифровых данных.

# Стандарты беспроводной передачи данных

## WiMAX



# Стандарты беспроводной передачи данных

## Сравнение пропускных способностей и дальности передачи



# Стандарты беспроводной передачи данных

## Bluetooth

### **Bluetooth**

Беспроводная технология организации связи, на физическом уровне используются радиосигналы.

В отличие от Wi-Fi имеет более низкое энергопотребление, как следствие ограниченный диапазон действия (до 16 метров в лучшем случае), и более низкую пропускную способность.

# Стандарты беспроводной передачи данных

## IrDA

### **IrDA (Infrared Data Association)**

Группа стандартов цифровой передачи, на физическом уровне реализуемая в инфракрасном диапазоне волн.

Обладает низким энергопотреблением, низкой пропускной способностью и требует прямой видимости приемника и передатчика. Используется в пультах управления телевизором.

В системах передачи данных вытеснена Wi-Fi или Bluetooth.

---

# **Разработка приложений телекоммуникационного обмена**

---



### Аппаратная часть

Аппаратная часть приложений телекоммуникационного обмена в задачах управления, как правило, связана с контроллерами обработки данных, получаемых от датчиков (сенсоров).

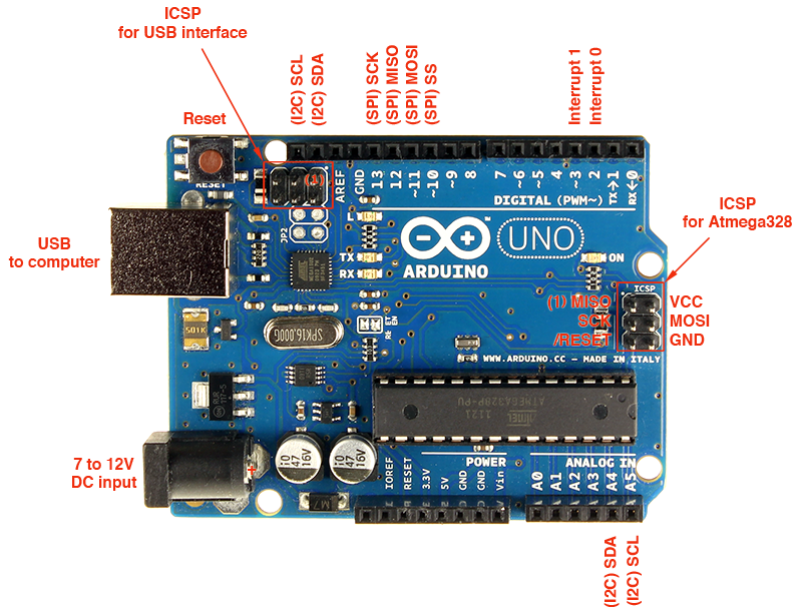
Существует несколько аппаратных платформ и типовых решений:

- Arduino-совместимые платы;
- платы на базе контроллеров STM32;
- одноплатные компьютеры (Raspberry Pi, NI MyRIO);
- и др.

Каждое из решений имеет большое количество реализаций, адаптированных под конкретные задачи, имеющие свои достоинства и недостатки. Универсального решения не существует.

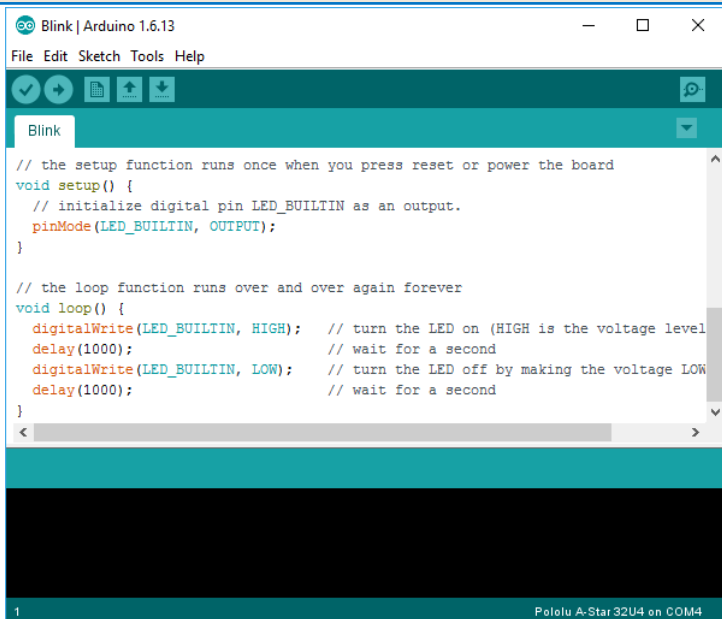
# Разработка приложений телекоммуникационного обмена

## Аппаратная часть. Arduino



# Разработка приложений телекоммуникационного обмена

## Аппаратная часть. Arduino IDE



The screenshot shows the Arduino IDE interface. The window title is "Blink | Arduino 1.6.13". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for saving, running, uploading, and downloading. The sketch name "Blink" is visible in the top left. The code editor contains the following C++ code:

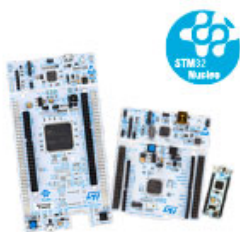
```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

At the bottom of the IDE, the status bar shows "1" on the left and "Pololu A-Star 32U4 on COM4" on the right.

# Разработка приложений телекоммуникационного обмена

## Аппаратная часть. STM32



STM32 Nucleo  
development boards

Discovery kits

Evaluation boards

STM32 Nucleo  
expansion boards

Third-party  
boards

Flexible  
prototyping

Key feature  
prototyping

Full feature  
evaluation

Add-on  
functionalities

From full  
evaluation to  
open hardware

# Разработка приложений телекоммуникационного обмена

## Аппаратная часть. STM32



Configure  
& Generate Code

Compile and Debug  
IDEs

Monitor, Program  
& Utilities

# Разработка приложений телекоммуникационного обмена

## Аппаратная часть. STM32

MicroXplorer Untitled.1ec: STM32F4052GTx

File Tools Windows Help

Pinout Configuration Power Consumption Calculator

Settings

Family: STM32F4  
MCU: STM32F4052GTx  
Package: LQFP144

Expand All + Collapse All -  
Disable Modes

Tree Interface Accordion Interface

Peripherals

- ADC1
- ADC2
- ADC3
- CAN1
- CAN2
- DAC
- FSMC\_CF
- FSMC\_NAND16
- FSMC\_NOR\_M16
- FSMC\_NOR\_32M
- I2C1
- I2C2
- I2C3
- I2S2

MCUs Selection

Family	Sub-Family	McU	Package	Required Peripherals
STM32F4	STM32F405	STM32F4050VE-GTx	WLCSP90	None
STM32F4	STM32F405	STM32F405RGTx (RevA)	LQFP64	None
STM32F4	STM32F405	STM32F405RGTx	LQFP64	None
STM32F4	STM32F405	STM32F405VGTx (RevA)	LQFP100	None
STM32F4	STM32F405	STM32F405VGTx	LQFP100	None
STM32F4	STM32F405	STM32F405VGTx (RevA)	LQFP144	None

### Программная часть

Программная часть приложений телекоммуникационного обмена в задачах управления, как правило, связана с отображением информации и обработкой данных на ПЭВМ.

Формально ПО для контроллеров тоже входит сюда, но на текущий момент, это более низкий уровень работы, как правило, выполняемый другим специалистом.

Для реализации программной части можно:

- National Instruments LabVIEW;
- slowmotion sex with Mathworks MATLAB;
- hardcore action с GUI на C/C++;
- ~~BDSM with Assembler~~ (GUI - анриал);
- и др.

# Разработка приложений телекоммуникационного обмена

## Программная часть. LabVIEW

PASS 195 - C:\Wheel\_Pubs\rcv

### Vehicle Drive Simulator

#### POWER SUPPLIES

**MAIN**

Primary Power Supply: **13.532** (Current)

Backup Battery Supply: **0.024** (Current)

Backup Battery Supply: **0.000** (Current)

#### ECU

ECALL ICALL RCALL

LED PWM %: 0  
PWM Freq: 0  
ACTIVE DTCs: 0

AUDIO: TX, RX, TX, RX

GPS DATA: UTC DATE: 1/30, SATELLITES VISIBLE: 9, IN USE: 0

HEADING: 6554  
Latitude: 27.000000  
Longitude: 180.000000

HEIGHT: 179.0  
SPEED: 2359.24

#### GPS

Selected GPS Scenario File: **DCC\_CircleDR.scn**

Scenario Status: **Running** 5

Start Time: 14-Nov-2007 13:07:10  
Current Time: 14-Nov-2007 13:07:10  
End Time: 21-Nov-2007 13:00:00

Generated: **08:08:00.00**  
RF Data: **0**  
Duration: 0:00:00.00

GPS Simulation Status

Latitude: 42.110058  
Longitude: -83.600731  
HEIGHT: 33.90  
SPEED: 72.00

Seek Chart

Signal OFF: 1, 2, 10, 21, 6, 31, 22, 13, OFF, OFF, OFF, OFF

Speed: 0, 4.3, 10.7, 17.1, 23.5, 29.9, 36.3, 42.7, 49.1, 55.5, 61.9, 68.3, 74.7, 81.1, 87.5, 93.9, 100.3

#### SYSTEM CONTROL

Mode 1 Signal Status: **SOFT**  
Mode 2 Signal Status: **SOFT**

RF Output: **Signal OFF**

Connection Control: **Connect Hold**

See Backend SW Controls Window  
See GPS Message

Received SMS Message

#### STATUS

Current Operating Mode: **Running**

Logging Status: **Logging OFF**

ECU DTC Status: **No DTCs Detected**

Fault Insertion: **No Faults Inserted**

#### SUBSYSTEMS

- Power Supplies
- ECU Interface
- GPS Interface
- GSM Interface
- Vehicle CAN Interface
- Fault Insertion
- Scripting
- Logging

#### VEHICLE

Vehicle Ignition: **LOCK** **ACT** **OFF** **P.M.** **START**

Modem CAN Bus Traffic: **Vehicle Traffic Enabled**

Crash Sensor Impact Type: **No Shock**

**CRASH**

Network State: **AWAK**

**Change Vehicle Configuration**

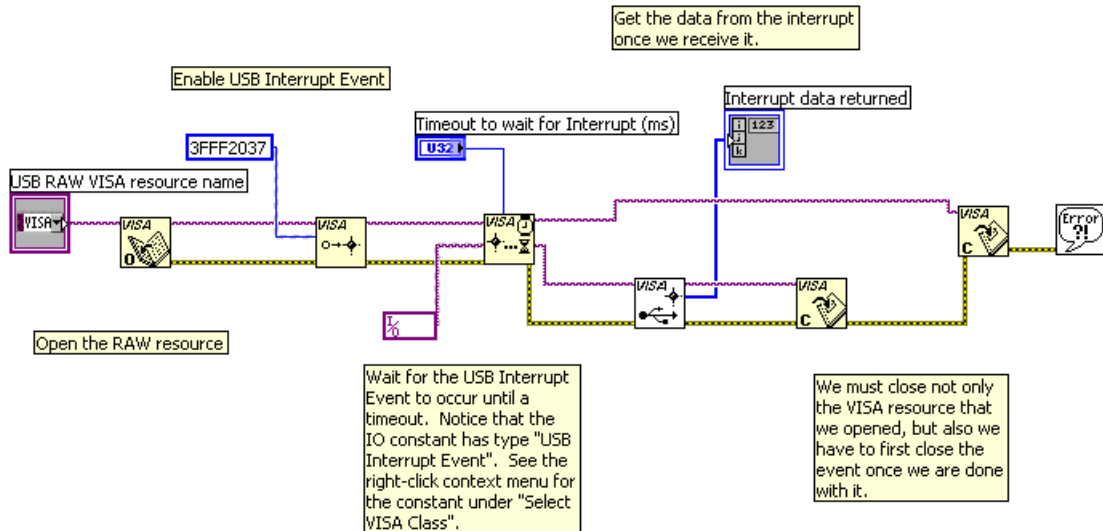
#### SYSTEM MESSAGES

01:39:07.214: Logger: Sys PS Owner: PWR\_Output



# Разработка приложений телекоммуникационного обмена

## Программная часть. LabVIEW



# Разработка приложений телекоммуникационного обмена

## Программная часть

Существует несколько основных типовых задач при разработке приложений телекоммуникационного обмена, как правило, связанных с величиной объема передачи данных, а именно:

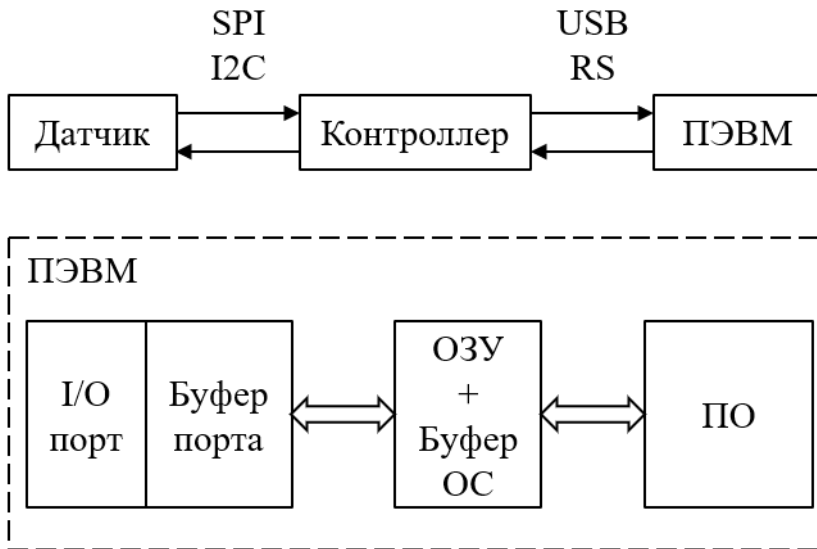
- режим обработки телеметрии;  
*(поток данных идет с датчика в ПЭВМ, почти сплошной)*
- режим программного управления.  
*(поток данных идет с ПЭВМ к контроллеру, почти сплошной)*

при этом взаимодействие ПЭВМ и контроллера строится по механизму:

- запрос-ответ;  
*(чаще всего)*
- по независимым каналам передачи.

**Всегда в канале передачи присутствует служебная информация**

## Разработка приложений телекоммуникационного обмена



# Хранение данных в памяти вычислителей

## Последовательности бит бывают:

- right-msb  
*(старший бит справа)*
- left-msb  
*(старший бит слева)*

MSB				LSB			
8	7	6	5	4	3	2	1
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

# Хранение данных в памяти вычислителей

## Целочисленные типы

### Знаковые и беззнаковые типы

При хранении знаковых и беззнаковых данных старший бит (положение которого зависит от выбора right-msb или left-msb) принимает значение 1, если число с минусом и 0, если с плюсом.

Например для числа  $-7_{10}$ , исходно заданного в знаковом формате, меняя представление на беззнаковое, получим  $135_{10}$ .

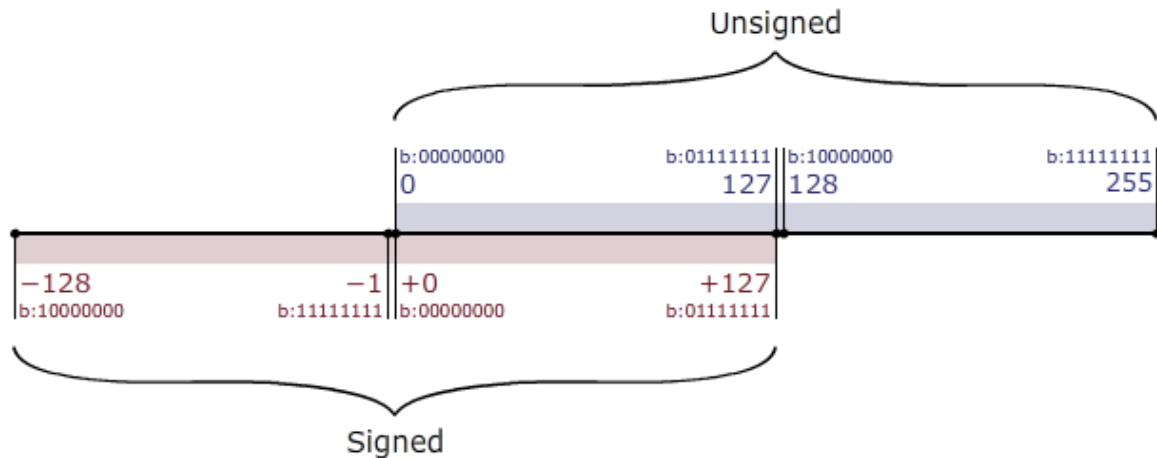
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
1	0	0	0	0	1	1	1

Для двухбайтного случая получится  $32775_{10}$ .

$2^{15}$	$2^{14}$	...	$2^3$	$2^2$	$2^1$	$2^0$
1	0	...	0	1	1	1

# Хранение данных в памяти вычислителей

## Целочисленные типы



# Хранение данных в памяти вычислителей

## Числа с плавающей точкой

Существует несколько стандартов хранения в памяти вычислителя чисел с плавающей точкой. Наиболее распространены (стандарт IEEE 754):

- число с одинарной точностью;  
(*single precision, 4 байта*)
- число с двойной точностью.  
(*double precision, 8 байт*)

В которых последовательность байт состоит из знака (S), порядка (Q) и мантисы (C).

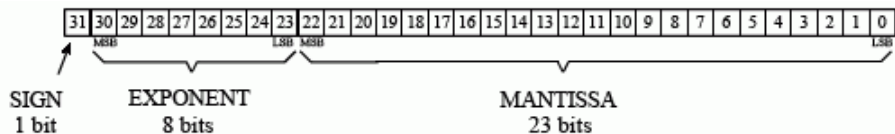
**Формула определения числа:**

$$V = (-1)^S \cdot C \cdot b^Q$$

где  $b$  - основание (2 или 10), а  $S$  равно 1 для отрицательных чисел, а порядок  $Q$  смещен, т.е. старший бит порядка отвечает за знак.

# Хранение данных в памяти вычислителей

## Числа с плавающей точкой



### Example 1

0 00000111 110000000000000000000000

↓     ↓                     ↓  
+     7                     0.75

$$+ 1.75 \times 2^{(7-127)} = + 1.316554 \times 10^{-36}$$

### Example 2

1 10000001 011000000000000000000000

↓     ↓                     ↓  
-     129                     0.375

$$- 1.375 \times 2^{(129-127)} = - 5.500000$$



# Хранение данных в памяти вычислителей

## Последовательности байт бывают:

- big-endian  
*(от старшего байта к младшему)*
- little-endian  
*(от младшего байта к старшему)*

31	23	15	7	0
Byte 3	Byte 2	Byte 1	Byte 0	
0	0	0	1	

Little Endian

31	23	15	7	0
Byte 0	Byte 1	Byte 2	Byte 3	
1	0	0	0	

Big Endian

# Хранение данных в памяти вычислителей

## Обработка байт данных

Зачастую при чтении данных с устройств необходимо “подправлять” представление полученных байт с данными следующими операциями (через функции MATLAB):

- перестановка байт;  
(`swapbytes()`)
- побитный сдвиг;  
(`bitshift()`)
- преобразование (приведение к) типа;  
(`typecast()` и `cast()`)
- преобразование типа данных;  
(`uint8()`, `int8()`, `double()` и др.)
- побитное сравнение.  
(`bitand()`, `bitxor()` и др.)



*The*

*End*