

Радиомодем. Тактовая и пакетная синхронизация.

Статья 4.



Алексей Воробьев, 21 мая 2013г.

САМОСТОЯТЕЛЬНАЯ РАБОТА

«РАДИОМОДЕМ. ТАКТОВАЯ И ПАКЕТНАЯ СИНХРОНИЗАЦИЯ»

СТАТЬЯ 4.

Выполнили: Воробьев А.Г., Никишкин П.Б.

Необходимо организовать обмен сообщениями между двумя любыми терминалами по радиоканалу (Рисунок 1). Один из терминалов является ведущим устройством (master), отвечающим за выполнение сценариев соединения и за управление радиоресурсом, другой – slave, исполняющим «волну» ведущего терминала.



Рисунок 1. Общая структура.

Постановка задачи:

В рамках данной статьи необходимо реализовать процесс тактовой синхронизации, а также рассмотреть процесс передачи текстового сообщения. В качестве сообщения взять замечание к предыдущей статье ([Радиомодем. Тактовая и пакетная синхронизация. Статья 3.](#)): “Основные цели работы не достигнуты: отсутствует иллюстрация процесса временной синхронизации и не рассмотрен процесс передачи текстового сообщения. Предлагаю в качестве передаваемого сообщения при моделировании использовать текущий комментарий”.

Анализ поставленной задачи. Синхронизация.

Решение поставленной задачи начнем с рассмотрения процесса синхронизации. Для этого вспомним структуру передаваемого пакета (Рисунок 2).

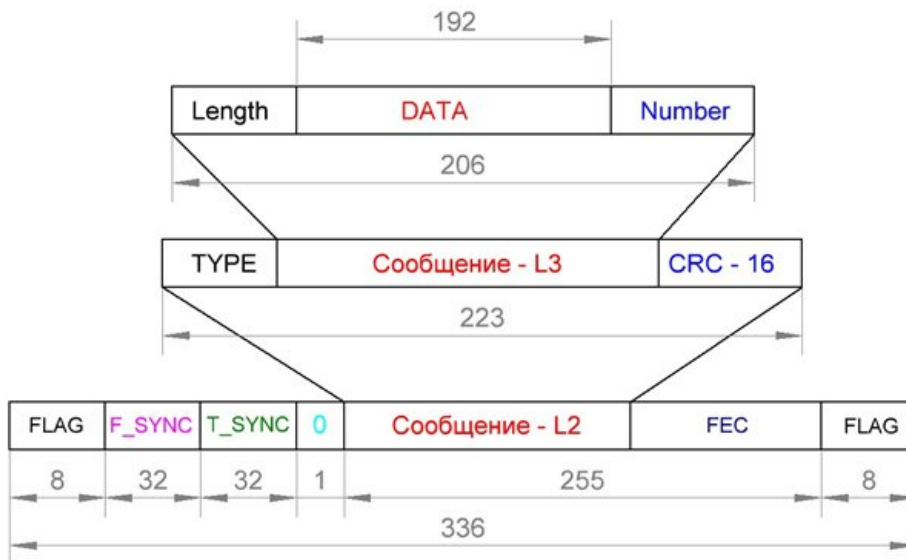


Рисунок 2. Структура передаваемого пакета.

Синхронизация состоит из трех частей:

1. Пакетная (FLAG) – необходима для определения начала и конца пакета;
2. Частотная (F_SYNC) – необходимо для подстройки частоты приемником;
3. Тактовая (T_SYNC) – необходима для подстройки тактового генератора в приемнике, чтобы он верно определял начало бита.

Реализация первых двух типов синхронизации в данной работе не рассматривается. Поля FLAG и F_SYNC включаются в передаваемый пакет, но на приемной стороне с ними никаких манипуляций не производится.

Тактовая синхронизация построена на основе определения взаимнокорреляционной функции (ВКФ). То есть на приемной стороне известен сигнал T_SYNC и, сравнивая его с принимаемым сигналом, определяется степень их схожести. По максимуму ВКФ определяется совпадение пришедшей последовательности с ранее известной на приемной стороне, и приемник воспринимает далее идущие символы как информационные данные.

Чтобы найти ВКФ, необходимо определиться с последовательностью T_SYNC. Так как на передающей стороне используется модуляция QAM – 16, то каждый символ представляется четырьмя битами. Рассмотрим сигнальное созвездие (Рисунок 3).

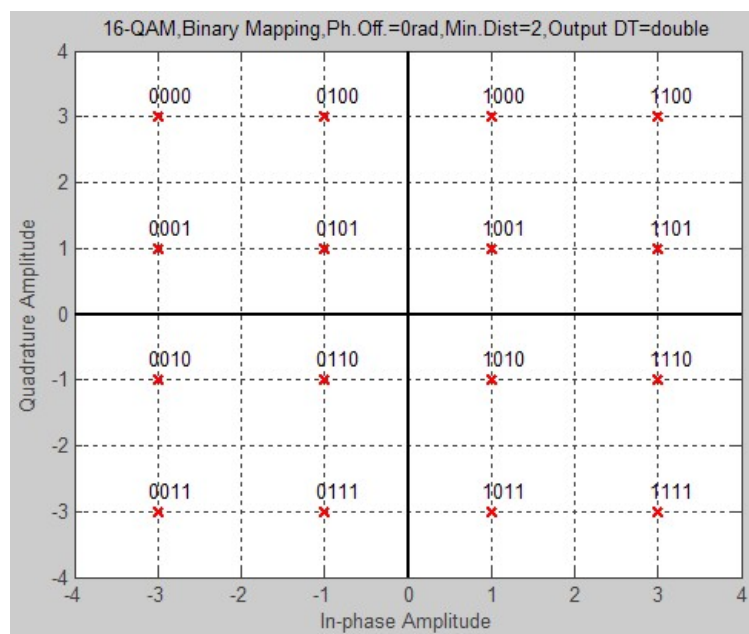


Рисунок 3. Сигнальное созвездие QAM – 16.

Выберем два символа, имеющих наибольшее расстояние, и сделаем из них чередующуюся последовательность. Как видно из рисунка 2, поле T_SYNC состоит из 32 бит, следовательно, из 8 символов QAM – 16. T_SYNC = [11000011110000111100001111000011].

Каждый символ после модулятора представляет собой точку на комплексной плоскости, поэтому необходимо определиться с эталонным сигналом, с которым будет сравниваться приходящая последовательность. Символ [1100] равен числу $3 + i \cdot 3$, а символ [0011] равен $-3 - i \cdot 3$.

Тогда эталонный сигнал $sync = [3 + i \cdot 3 \ -3 - i \cdot 3 \ 3 + i \cdot 3 \ -3 - i \cdot 3 \ 3 + i \cdot 3 \ -3 - i \cdot 3 \ 3 + i \cdot 3 \ -3 - i \cdot 3]$.

Максимум ВКФ будет тогда, если взять интеграл от последовательности $sync \cdot sync$. Это будет ее энергия. В программе Mathcad вычислим ВКФ (Рисунок 4).

$$rxIQ := \begin{pmatrix} 3 + i \cdot 3 \\ -3 - i \cdot 3 \\ 3 + i \cdot 3 \\ -3 - i \cdot 3 \\ 3 + i \cdot 3 \\ -3 - i \cdot 3 \\ 3 + i \cdot 3 \\ -3 - i \cdot 3 \end{pmatrix} \quad sync := \begin{pmatrix} 3 + i \cdot 3 \\ -3 - i \cdot 3 \\ 3 + i \cdot 3 \\ -3 - i \cdot 3 \\ 3 + i \cdot 3 \\ -3 - i \cdot 3 \\ 3 + i \cdot 3 \\ -3 - i \cdot 3 \end{pmatrix}$$

$$vkf := rxIQ_0 \cdot sync_0 + rxIQ_1 \cdot sync_1 + rxIQ_2 \cdot sync_2 + rxIQ_3 \cdot sync_3 + rxIQ_4 \cdot sync_4 + rxIQ_5 \cdot sync_5 + rxIQ_6 \cdot sync_6 + rxIQ_7 \cdot sync_7$$

$$vkf = 144i$$

$$|vkf| = 144$$

Рисунок 4. ВКФ сигнала при полном совпадении.

Как видно из данного рисунка максимальное значение ВКФ равно 144. Теперь, зная максимально возможное значение, можно реализовать алгоритм, который будет находить этот максимум. Но так как у нас в системе имеется источник шума, то наш максимум может несколько отличаться. Чтобы определиться со значение порога, найдем значение ВКФ с предыдущей группой символов (Рисунок 5).

$$rxIQ := \begin{pmatrix} -3 + i \cdot 3 \\ 3 + i \cdot 3 \\ -3 - i \cdot 3 \\ 3 + i \cdot 3 \\ -3 - i \cdot 3 \\ 3 + i \cdot 3 \\ -3 - i \cdot 3 \\ 3 + i \cdot 3 \end{pmatrix} \quad sync := \begin{pmatrix} 3 + i \cdot 3 \\ -3 - i \cdot 3 \\ 3 + i \cdot 3 \\ -3 - i \cdot 3 \\ 3 + i \cdot 3 \\ -3 - i \cdot 3 \\ 3 + i \cdot 3 \\ -3 - i \cdot 3 \end{pmatrix}$$

$$vkf := rxIQ_0 \cdot sync_0 + rxIQ_1 \cdot sync_1 + rxIQ_2 \cdot sync_2 + rxIQ_3 \cdot sync_3 + rxIQ_4 \cdot sync_4 + rxIQ_5 \cdot sync_5 + rxIQ_6 \cdot sync_6 + rxIQ_7 \cdot sync_7$$

$$vkf = -18 - 126i$$

$$|vkf| = 127.279$$

Рисунок 5. ВКФ сигнала при сдвиге на один символ.

Теперь зная эти два значения, выберем оптимальный порог: $144 - 127,3 = 16,7$. Следовательно, порог равен $144 - 16,7/2 = 135,65$.

Тогда отрывок программы синхронизации выглядит следующим образом.

```
k = 1; %переменная для определения номера первого инф. символа
for j = 1:(length(rxIQ)-7) % цикл перебирает все символы
vkf = 0;
m = 1;
for i = k:(k+7) %цикл вычисляет ВКФ с последовательностью
%символов, начиная с j
vkf = vkf + rxIQ(i) * st.sync(m);
```

```

    m = m + 1;

end;

vkfplot(j+1)= abs(vkf);%накапливает значения ВКФ для ее построения

if abs(vkf) <135.65 %сравнивает с максимально возможным значением

    %если больше, то цикл прерывается

    k = k + 1;

else

    break

end;

end;

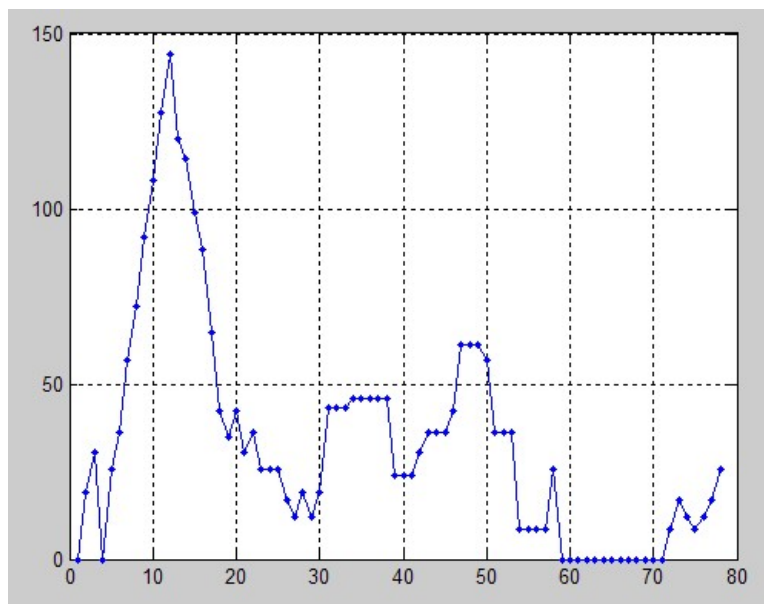
figure(2);

plot(vkfplot, '.-'); grid % график ВКФ

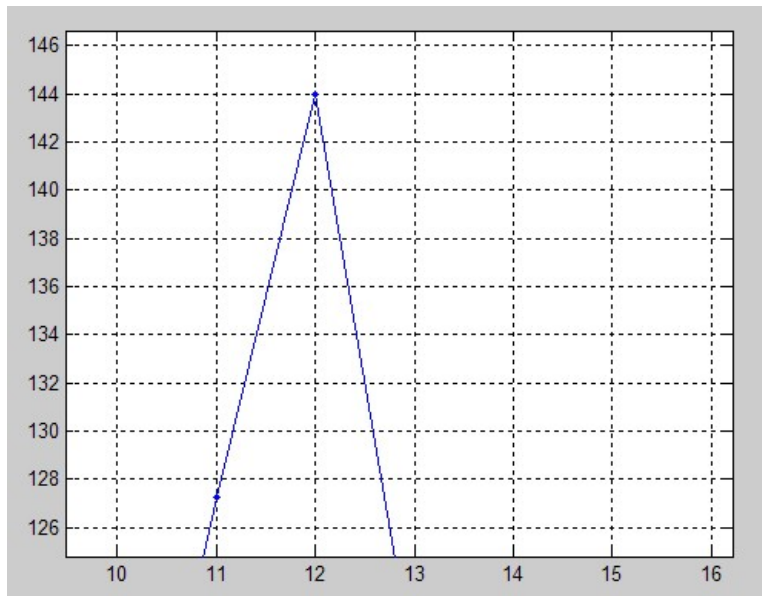
hold all;

```

Данная программа определяет номер первого информационного пакета. Также для наглядности можно построить график ВКФ для всей символьной последовательности. Необходимо только увеличить порог больше 144, чтобы не сработало условие. Посмотрим ВКФ без воздействия шума (Рисунок 6) и с ним (Рисунок 7).

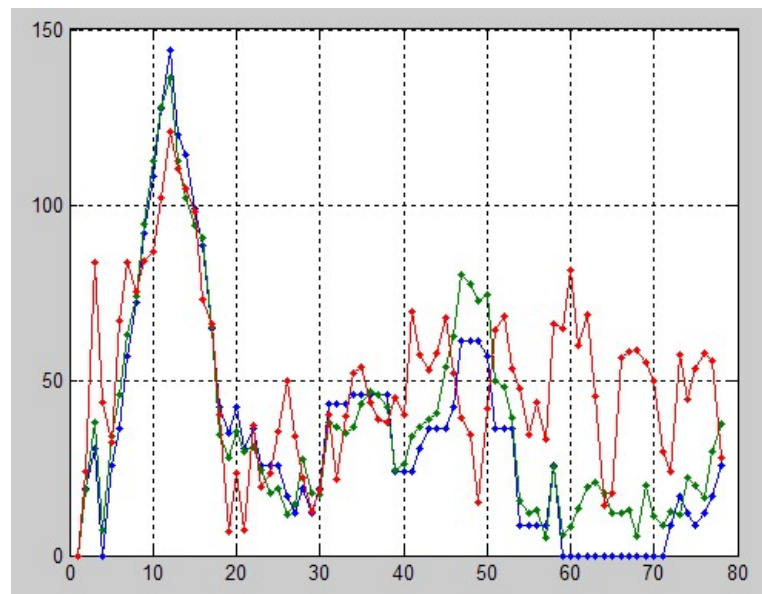


a)

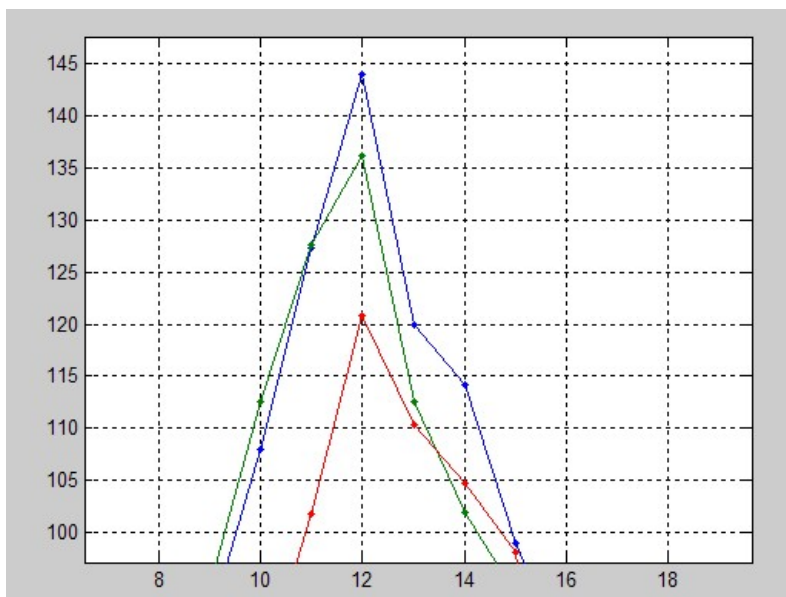


б)

Рисунок 6. Графики ВКФ без шума.



а)



б)

Рисунок 7. Графики ВКФ с шумом. Синяя – без шума, Зеленая – ОСШ= 20дБ, Красная – ОСШ= 0дБ.

Как видно из рисунка 6, максимум получается при $k = 12$, и равен он 144, при $k = 11$ ВКФ равна 127,3. Программа выдает значение k , полученное при последнем значении ВКФ, меньшем 144, и к нему прибавляет длину последовательности тактовой синхронизации. Получается $11 + 8 = 19$. с 19 символа начинается демодуляция и последующее декодирование. Но как видно из рисунка 7, из-за шума значение ВКФ изменяется, и при большом шуме значение ВКФ может очень сильно измениться и произойдет срыв синхронизации.

Анализ поставленной задачи. Передача текстового сообщения.

Необходимо передать текстовое сообщение, например: *“Основные цели работы не достигнуты: отсутствует иллюстрация процесса временной синхронизации и не рассмотрен процесс передачи текстового сообщения. Предлагаю в качестве передаваемого сообщения при моделировании использовать текущий комментарий”*.

Из рисунка 2 видно, что размер передаваемой информации в каждом пакете равен 192 бита, следовательно, для передачи сообщения большей длины необходимо его разбивать на пакеты.

Длина данного сообщения равна 244 символа, каждый из которых представлен 16 битами. Следовательно, его длина в битах – 3904. Чтобы найти число пакетов, необходимых для передачи, разделим 3904 на 192. Получим результат 20,3333. То есть нам необходимо 20 полных пакетов и один пакет, заполненный только на треть. Для того чтобы не портить структуру передаваемого пакета дополним последний нулями, чтобы его размер тоже был 192 бита. Получается, что всего понадобится 21 пакет. Далее организуем цикл передачи и приема 21 пакета, причем результат приема каждого из пакетов будем добавлять в строковую переменную, чтобы по завершению передачи всех пакетов восстановить сообщение (Рисунок 8).

```
Количество символов текста: 244
Количество бит: 3904
количество пакетов: 21
```

```
Поле информации: 192
Поле DATA: 192
Поле Сообщение_L3: 206
Поле Сообщение_L2: 223
Поле "Кодовое слово": 255
Поле "Сообщение_L1": 336
Система синхронизована, прием информации со следующего символа
Первый информационный символ под номером 19
Длина: 12
Тип сообщения: traffic
Номер пакета: 1
Ошибки отсутствуют
Текстовое сообщение: Основные цел
```

```
Поле информации: 192
Поле DATA: 192
Поле Сообщение_L3: 206
Поле Сообщение_L2: 223
Поле "Кодовое слово": 255
Поле "Сообщение_L1": 336
Система синхронизована, прием информации со следующего символа
Первый информационный символ под номером 19
Длина: 12
Тип сообщения: traffic
Номер пакета: 15
Ошибки отсутствуют
Текстовое сообщение: передаваемо
```

```

Поле информации: 64
Поле DATA: 192
Поле Сообщение_L3: 206
Поле Сообщение_L2: 223
Поле "Кодовое слово": 255
Поле "Сообщение_L1": 336
Система синхронизована, прием информации со следующего символа
Первый информационный символ под номером 19
Длина: 4
Тип сообщения: traffic
Номер пакета: 21
Ошибки отсутствуют
Текстовое сообщение: рий.

```

Принятое сообщение: Основные цели работы не достигнуты: отсутствует иллюстрация процесса временной синхронизации и не рассмотрен процесс передачи текстового сообщения. Предлагаю в качестве передаваемого сообщения при моделировании использовать текущий комментарий.

Рисунок 8. Отрывки результата выполнения программы при отсутствии шума.

Теперь добавим шум, но прежде, чтобы добиться реального результата, необходимо определить энергию сигнала. Для этого воспользуемся стандартным атрибутом функции `awgn`, который называется `'measured'`. С помощью него сначала измеряется мощность сигнала, а затем на основании `snr` накладывается шум.

Результат работы программы при ОСШ = 15дБ (Рисунок 9).

```

Поле информации: 192
Поле DATA: 192
Поле Сообщение_L3: 206
Поле Сообщение_L2: 223
Поле "Кодовое слово": 255
Поле "Сообщение_L1": 336
Система синхронизована, прием информации со следующего символа
Первый информационный символ под номером 19
Длина: 12
Тип сообщения: traffic
Номер пакета: 2
Имеются ошибки
Текстовое сообщение: □ рРаоты =e

```

```

Поле информации: 192
Поле DATA: 192
Поле Сообщение_L3: 206
Поле Сообщение_L2: 223
Поле "Кодовое слово": 255
Поле "Сообщение_L1": 336
Система не синхронизована!!!
Длина: 12
Тип сообщения: traffic
Номер пакета: 0
Имеются ошибки
Текстовое сообщение: □□□□□□□□□□□□

```

```

Поле информации: 192
Поле DATA: 192
Поле Сообщение_L3: 206
Поле Сообщение_L2: 223
Поле "Кодовое слово": 255
Поле "Сообщение_L1": 336
Система не синхронизована!!!
Длина: 12
Тип сообщения: service
Номер пакета: 56
Имеются ошибки
Текстовое сообщение: □с□□□□□□□□□□□□

```

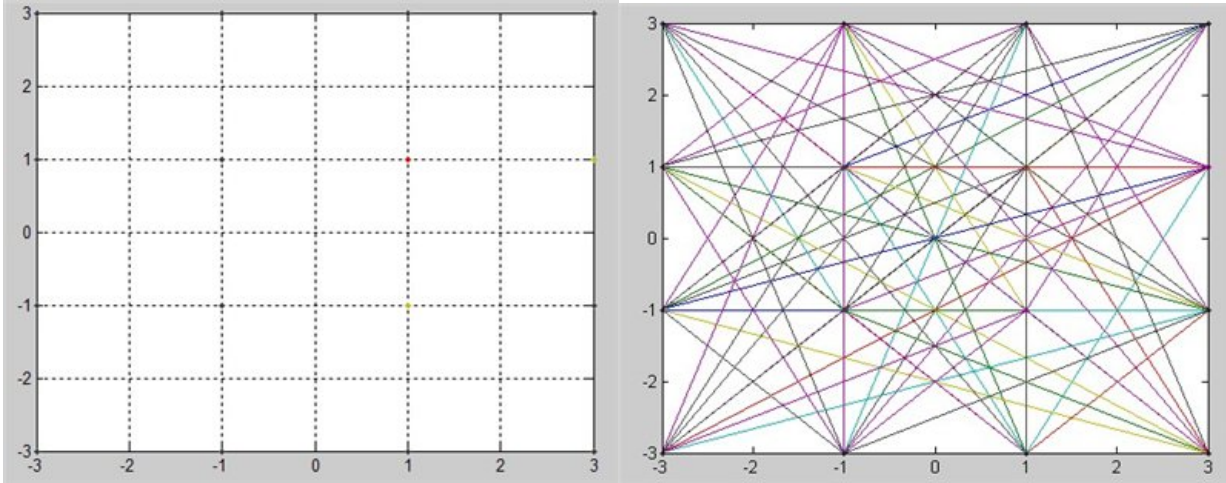
Принятое сообщение: Основные цели работы не достигнуты: отсутствует иллюстрация процесса временной синхронизации и не рассмотрен процесс передачи текстового сообщения. Предлагаю в качестве передаваемого сообщения при моделировании использовать текущий комментарий.

Рисунок 9. Результат работы программы при ОСШ = 15дБ.

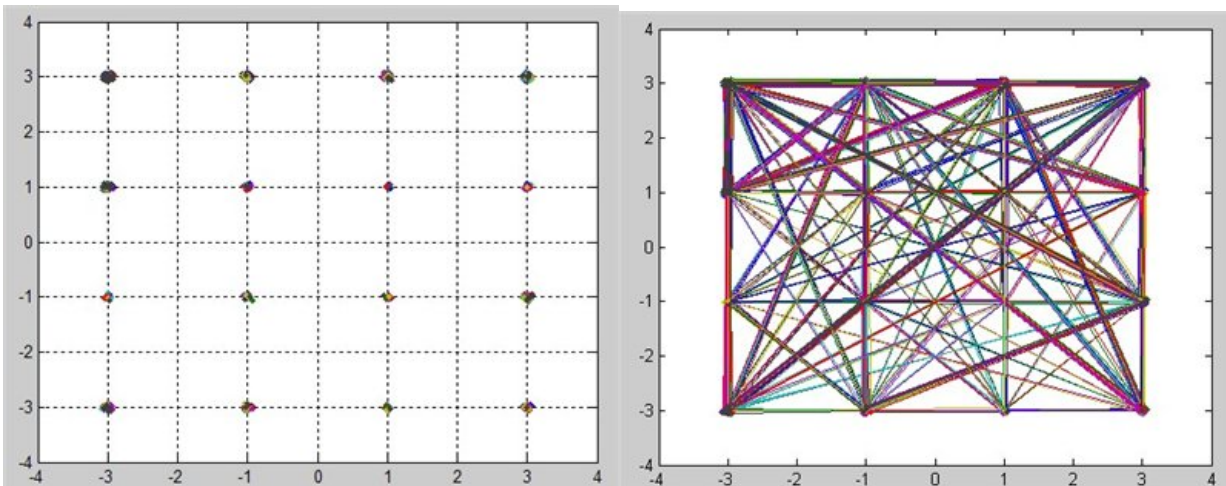
Как мы видим, из-за шума появляются ошибочные пакеты, причем происходят ошибки не только в информационной части, но и в служебной. При сильном шуме может произойти срыв синхронизации и программа завершится.

Также, сравнивая передаваемое и принятое сообщения, вычисляется количество ошибочно принятых символов. При данном ОСШ получилась 41 ошибка.

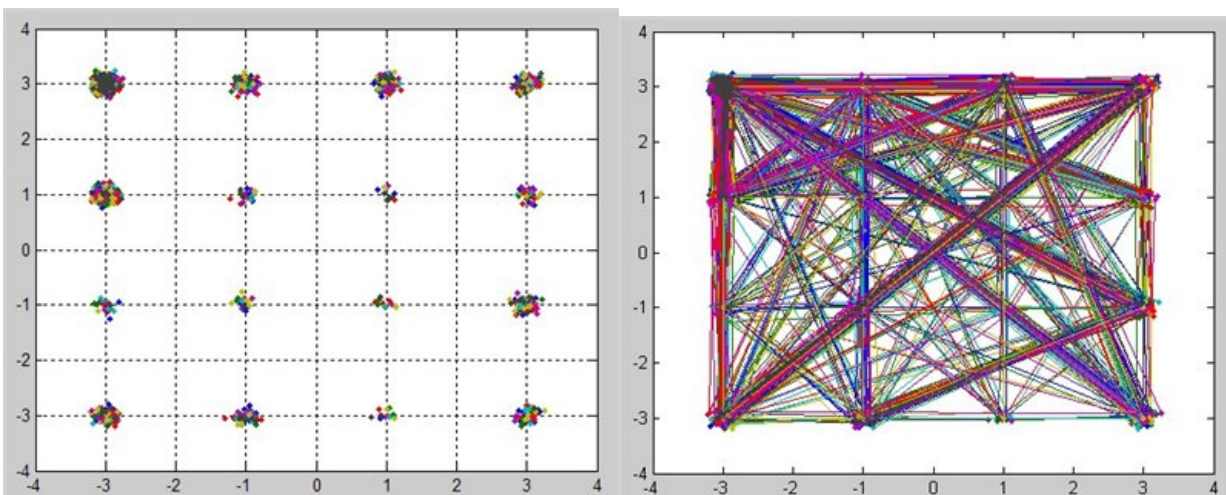
Проведем исследование сигнального созвездия и трассы изменения сигнала от ОСШ (Рисунок 10).



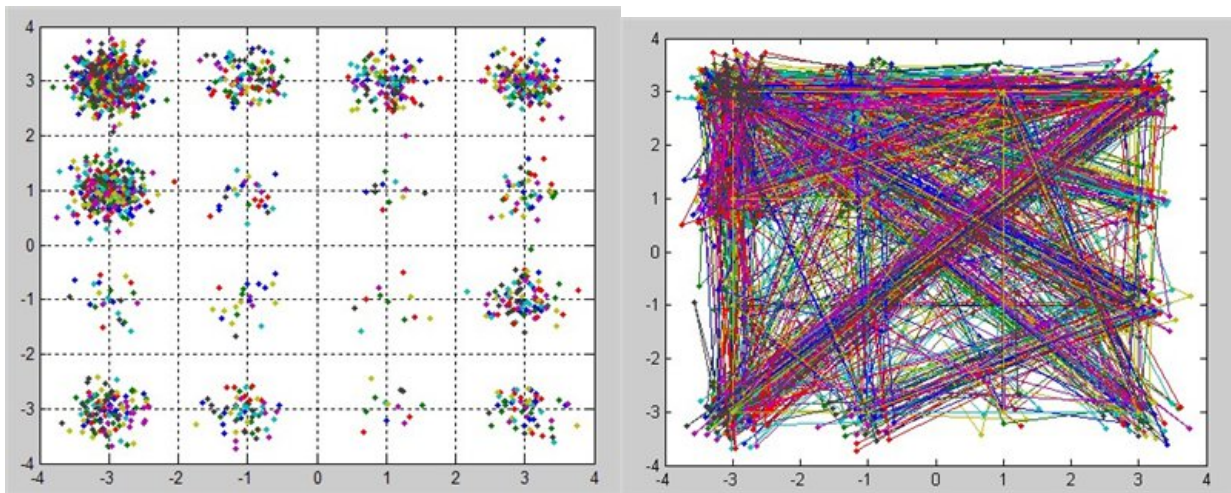
а)



б)



в)



г)

Рисунок 10. а) без шума; б) ОСШ= 40дБ; в) ОСШ=30дБ; г) ОСШ=20дБ.

Выводы: в данной работе удалось выполнить поставленные задачи по реализации тактовой синхронизации и передаче текстового сообщения.

Список используемой литературы:

1. Радиомодем. Тактовая и пакетная синхронизация. Статья 1.
2. Радиомодем. Тактовая и пакетная синхронизация. Статья 2.
3. Радиомодем. Тактовая и пакетная синхронизация. Статья 3.
4. Пояснения к выполнению самостоятельной работы(2)
5. Скляр Б. Цифровая связь. Теоретические основы и практическое применение. М.: Вильямс, 2003г.
6. Исходный код

