

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт инженерной физики и радиоэлектроники
Радиотехники
кафедра

ОТЧЕТ ПО ПРЕДДИПЛОМНОЙ ПРАКТИКЕ

Разработка лабораторного макета для изучения навигационных систем

| | | |
|-------------------------------|---------------|---------------------|
| Руководитель | _____ | <u>А.А.Ерохин</u> |
| | подпись, дата | инициалы, фамилия |
| Руководитель | _____ | <u>С.П.Царев</u> |
| | подпись, дата | инициалы, фамилия |
| Студент РФ19-32Б | _____ | <u>Д.Р.Соловьев</u> |
| номер группы, зачетной книжки | подпись, дата | инициалы, фамилия |

Красноярск 2023

Содержание

| | |
|---|----|
| ВВЕДЕНИЕ..... | 3 |
| Раздел 1. Теоретическая часть | 4 |
| 1.1 Принцип работы системы: | 4 |
| 1.2 Псевдослучайные последовательности: | 7 |
| Раздел 2. Реализация программной части навигационной системы..... | 13 |
| 2.1 Выбор сигнала для измерения расстояния | 13 |
| 2.2 Решение задачи трилатерации: | 16 |
| Заключение | 19 |
| Список используемых источников..... | 20 |
| Приложение А | 21 |
| Приложение Б..... | 22 |

ВВЕДЕНИЕ

Лабораторный макет для изучения навигационных систем позволит изучить принцип, по которому определяется расстояние между спутником и объектом в Глобальных Навигационных Спутниковых Системах. На время преддипломной практики были поставлены следующие цели:

Цели работы: разработать программу, генерирующую требуемый массив отчетов для последующего его использования в качестве сигнала в лабораторном макете. Также необходимо написать программу, рассчитывающую координаты приемника по полученным с помощью устройства данным.

Для выполнения поставленной цели были поставлены следующие задачи:

- выделить наиболее подходящие для реализации макета кодовые последовательности;
- разработать алгоритмы вычисления координат на основе замеров задержки звука. Цель состоит в том, чтобы учебный макет точно определял координаты источников звука на основе измерений задержки звука с динамиков.
- разработать функционал для построения графика на основе найденных значений, что поможет в понимании и анализе данных.

Раздел 1. Теоретическая часть

1.1 Принцип работы системы:

На объект, координаты которого необходимо определить, устанавливается приемно-передающее устройство – терминал.

Для позиционирования терминал подает запрос на спутники. Чем больше спутников ответят на запрос (в идеале – не менее 4), тем точнее будут определены координаты.[4]

Ответный сигнал поступает в терминал, программный комплекс которого анализирует время задержки для разных спутников. На основе анализа ответной информации определяются координаты объекта, на котором установлено приемное оборудование. Пример системы приведен на рисунке 1:

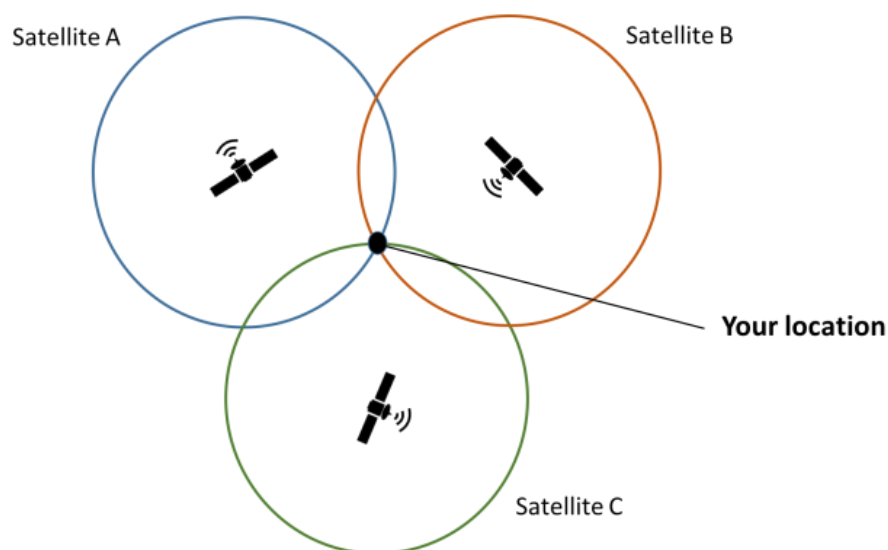


Рисунок 1 – Пример работы навигационной системы

При постоянной работе терминала (т.е. регулярной отправке запросов и анализе ответов) система ГЛОНАСС может определять не только положение, но и скорость движения объекта. При движении точность позиционирования снижается, но все равно остается достаточной для того, чтобы навигационное оборудование могло выполнить привязку координат объекта к электронной карте местности и построить маршрут. [1,2]

Глобальная навигационная спутниковая система (Global Navigation Satellite System – ГНСС) – это спутниковые системы (наиболее распространены GPS и ГЛОНАСС), используемые для определения местоположения в любой точке земной поверхности с применением специальных навигационных или геодезических приемников. ГНСС-технология нашла широкое применение в геодезии, городском и земельном кадастре, при инвентаризации земель, строительстве инженерных сооружений, в геологии и т.д.

ГЛОНАСС – это российская разработка, которая обеспечивает точное позиционирование объекта в пространстве с минимальной погрешностью. Для определения координат используется специальное оборудование, которое при поддержке наземной инфраструктуры связывается с сетью спутников, выведенных на околоземную орбиту.

Основой системы ГЛОНАСС являются 24 космических аппарата, которые движутся в трёх орбитальных плоскостях по 8 аппаратов в каждой плоскости, наклоненных к экватору под углом $64,8^\circ$, с высотой орбит 19100 км и периодом обращения 11 ч 15 мин 44 с. Выбранная структура орбитальной группировки обеспечивает движение всех космических аппаратов по единой трассе на поверхности Земли с ее повторяемостью через 8 суток. Такие характеристики обеспечивают высокую устойчивость орбитальной группировки системы ГЛОНАСС, что практически позволяет обходиться без коррекции орбит космических аппаратов в течение всего срока их активного существования. [4]

Для определения расстояния в навигационных системах применяют дальномерные коды (псевдослучайные последовательности).[1]

В случае учебного макета терминалом является микрофон, а динамики – выполняют роль спутников, относительно которых и будет определяться координата микрофона. Блок-схема работы лабораторного стенда представлена на рисунке 2.

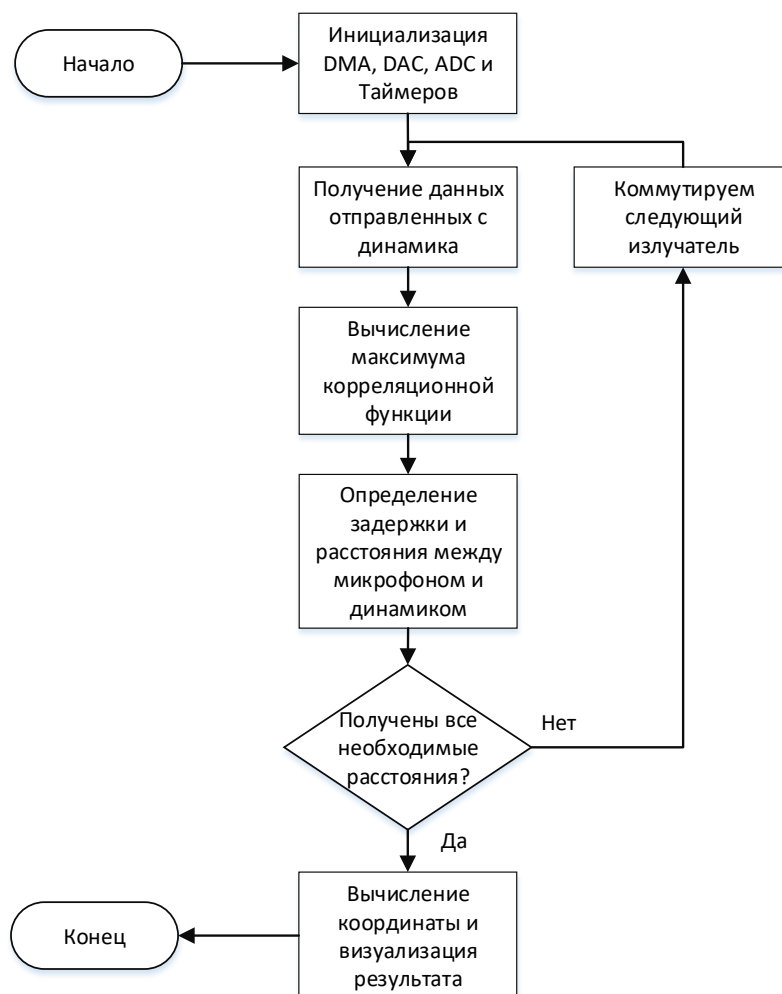


Рисунок 2 – Блок-схема работы лабораторного стенда

1.2 Псевдослучайные последовательности:

Псевдослучайные последовательности (ПСП) играют важную роль в навигации, особенно в глобальных навигационных спутниковых системах (ГНСС) таких, как GPS и ГЛОНАСС. Эти системы используют сигналы, создаваемые спутниками, для определения местоположения приемника на Земле.

Каждый спутник ГНСС генерирует сигналы, которые содержат коды, известные как кодовые последовательности. Коды представляют собой бинарные последовательности нулей и единиц, которые кажутся случайными. Однако эти последовательности скорее являются псевдослучайными, то есть они создаются с помощью детерминированных алгоритмов, которые могут генерировать такие последовательности.

Приемник ГНСС использует сигналы от нескольких спутников для определения своего местоположения. Приемник сравнивает фазу сигнала, получаемого от спутника, с фазой сигнала, создаваемого внутренней кодовой последовательностью приемника, чтобы определить эту разницу.

ПСП также используются для устранения ошибок измерения в ГНСС. Измерения могут быть искажены различными факторами, такими как многолучевое распространение и эффекты атмосферы. Однако, если заранее известны кодовые последовательности, сгенерированные спутником, и коды, сгенерированные приемником, то приемник может устранить эти ошибки и получить более точное измерение расстояния до спутника.

В целом, псевдослучайные последовательности очень важны для навигации и ГНСС. Они позволяют определять расстояние до спутников и устранять ошибки измерений, что делает системы ГНСС более точными и надежными. Для передачи псевдослучайной последовательности используется фазовая манипуляция.

Фазовая манипуляция (ФМ) – это вид модуляции, в котором информация кодируется изменением фазы несущего сигнала. При ФМ используются различные дискретные значения фазы, каждое из которых представляет определенное состояние символа или бита данных. [7,8]

Рассмотрим некоторые из них:

М-последовательности

М-последовательности, или последовательности максимальной длины, являются особыми фазоманипулированными сигналами. Они имеют ряд характеристик, которые делают их полезными в навигационных системах:

1. Периодичность: М-последовательность является периодической с периодом, состоящим из N импульсов или символов.

2. Боковые пики автокорреляционной функции М-последовательностей равны $-1/N$.

3. Псевдослучайность: М-последовательность в общем случае состоит из нескольких видов импульсов, которые распределены в периоде равномерно. Это делает М-последовательности псевдослучайными, поскольку они имеют свойства случайных последовательностей. Это особенно полезно в навигационных системах, где требуется иметь сложную, но предсказуемую последовательность для различения и идентификации сигналов.

4. Малые боковые пики: при усечении М-последовательности, то есть взятии непериодической последовательности длиной в период N , боковые пики автокорреляционной функции приближаются к $1/\sqrt{N}$. Это означает, что при увеличении N величина боковых пиков уменьшается, что делает М-последовательности более эффективными.

Из-за перечисленных свойств М-последовательности они широко используются в навигационных системах. Например, системы GPS и

ГЛОНАСС используют М-последовательности для идентификации спутниковых сигналов. [8]

М-последовательности обладают хорошими автокорреляционными характеристиками, но их использование в качестве псевдослучайной последовательности для лабораторного макета нежелательно по следующим причинам:

- Долгая обработка данных: М-последовательности обычно имеют длину, состоящую из множества символов. Использование такой длинной последовательности может привести к неэффективному использованию ресурсов устройства и требовать больше вычислительной мощности

- Переполнение памяти микроконтроллера: Использование М-последовательностей может требовать значительного объема памяти для хранения этих последовательностей и выполнения вычислений с ними. Ограниченные ресурсы памяти микроконтроллера могут повлиять на функциональность учебного макета.

Код Баркера

Код Баркера, также известный как последовательность Баркера, является одним из популярных типов псевдослучайных последовательностей, которые могут применяться в лабораторных макетах с участием радиосвязи или навигации. Вот некоторые преимущества его использования:

1. Код Баркера имеет относительно небольшое количество элементов или символов по сравнению с другими типами псевдослучайных последовательностей. При использовании занимают меньше памяти и требуют меньше вычислительных ресурсов для их обработки. На рисунке 3 изображена временная диаграмма сигнала, модулированного кодовой последовательностью Баркера из пяти символов.

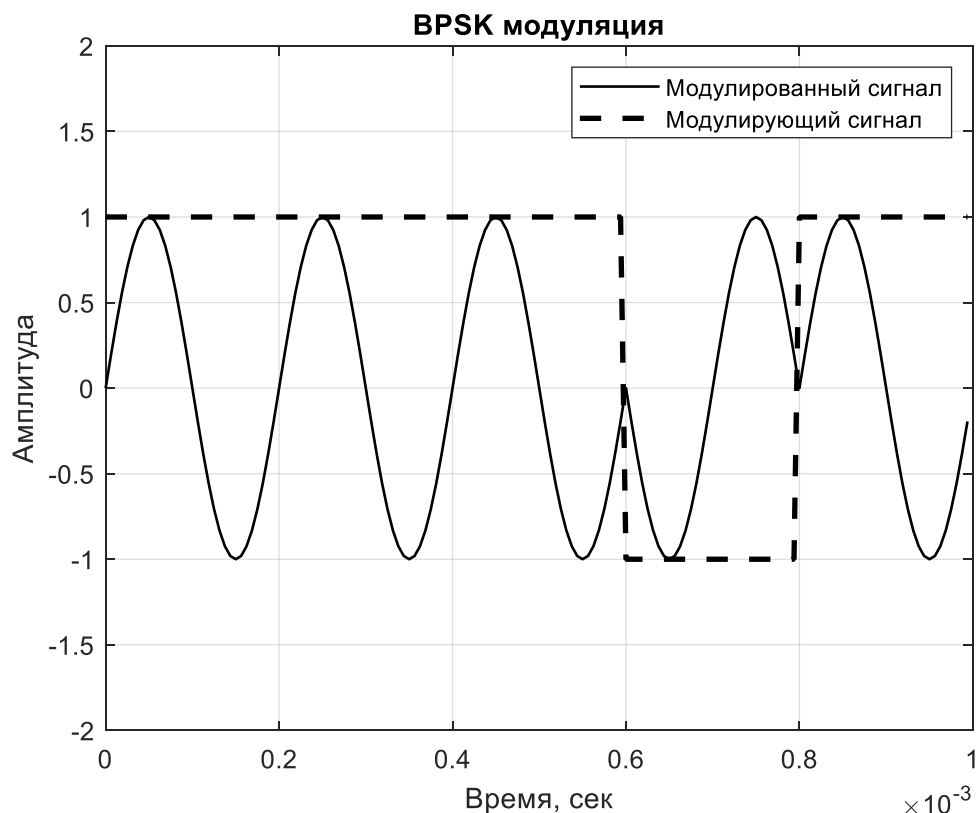


Рисунок 3 – Временная диаграмма сигнала, модулированного кодовой последовательностью Баркера

2. Код Баркера обладает отличными автокорреляционными свойствами. Это значит, что при сравнении принятого сигнала с оригинальным можно точно определить задержку сигнала. Последовательности Баркера имеют минимальный уровень боковых лепестков автокорреляционной функции $1/N$. На рисунке 4 показана автокорреляционная характеристика сигнала, модулированного кодовой последовательностью Баркера.

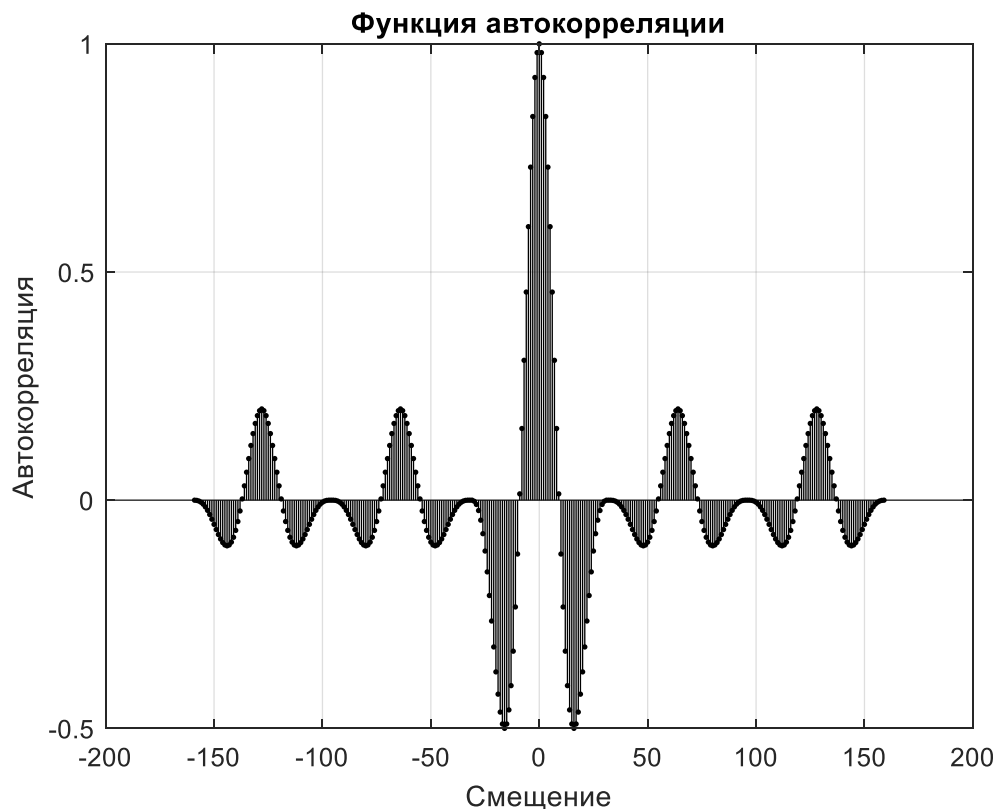


Рисунок 4 – Автокорреляционная характеристика сигнала, модулированного кодовой последовательностью Баркера

3. Устойчивость к помехам: Коды Баркера обладают хорошей способностью сопротивляться помехам. Их структура позволяет легко отличать их от других сигналов или шума. Это особенно полезно в условиях с шумом или многолучевым распространением сигнала. [7]

Код Баркера представляет собой компактную и эффективную псевдослучайную последовательность и использование для определения расстояния сигнала упрощает реализацию системы, а низкий уровень боковых лепестков автокорреляционной функции позволит точнее определять задержку.

Помимо Баркера и М-последовательностей, в радиосвязи и навигации широко применяются следующие типы псевдослучайных последовательностей:

1. Голдовские последовательности: Голдовские последовательности получаются путем комбинирования двух М-последовательностей с использованием операции XOR. Они обладают хорошими корреляционными свойствами и используются в системах CDMA (Code Division Multiple Access), таких как GSM и GPS.

2. Каскадные последовательности: Каскадные последовательности, также известные как Соломоновские последовательности, являются комбинацией нескольких последовательностей различной длины, полученных из семейства Фибоначчи или других базовых последовательностей. Они используются в радиолокации, системах связи и других приложениях.

3. Шумовые последовательности: Шумовые последовательности, также известные как псевдослучайные шумы, генерируются с использованием статистических алгоритмов или физических источников шума. Они обладают случайными свойствами и широко используются в системах радиосвязи для модуляции и демодуляции сигналов.

4. Коды Хэдемарка: Коды Хэдемарка являются математическими конструкциями, представляющими собой матрицы с определенными свойствами. Они используются в радиолокации, спектральном разделении сигналов и других приложениях, где требуется устойчивость к помехам и хорошая спектральная эффективность.[8]

Это лишь некоторые из популярных типов псевдослучайных последовательностей, используемых в радиосвязи и навигации. Каждый тип последовательности имеет свои особенности и применения в различных системах и технологиях.

Раздел 2. Реализация программной части навигационной системы

2.1 Выбор сигнала для измерения расстояния

Во время работы над макетом возникла необходимость в исследовании корреляционных свойств, а также в изменении параметров сигнала, модулированного последовательностью Баркера, с помощью которого мы определяем задержку.

По причине экономии места, отчеты излучаемого сигнала хранятся в постоянной памяти и во время работы устройства не изменяются, было решено записать их в массив констант при загрузке прошивки в устройство. Чтобы избежать ручного формирования массива, была написана специальная программа на языке MATLAB (код программы прикреплен в Приложении А).

Перед запуском программы, необходимо написать параметры требуемого сигнала в соответствующие переменные (Рисунок 5):

```
% кодируемая последовательность
code = barker2;
% количество отсчетов одного периода синуса
m = 12;
% длительность импульса в периодах синусоиды
num_of_periods_per_bit = 1;
% частота сигнала в Гц
fc = 5000;
```

Рисунок 5 – Параметры для настройки генерируемого с помощью программы сигнала

Во время выполнения кода программа рассчитает отчеты синусоидального сигнала, сформирует модулирующий сигнал и, путем перемножения отсчетов, смоделирует требуемый фазово-манипулированный сигнал. Полученный сигнал и его параметры записываются в текстовый файл.

Предварительно амплитуда сигнала домножается на 2048 (половина от максимального значения 12-ти разрядного ЦАП микроконтроллера) и инкрементируется на это же значение для того, чтобы убрать отрицательный полупериод.

Одновременно с этим вычисляется автокорреляция сигнала и строится временная диаграмма (Рисунок 6) и автокорреляционная функция (Рисунок 7):

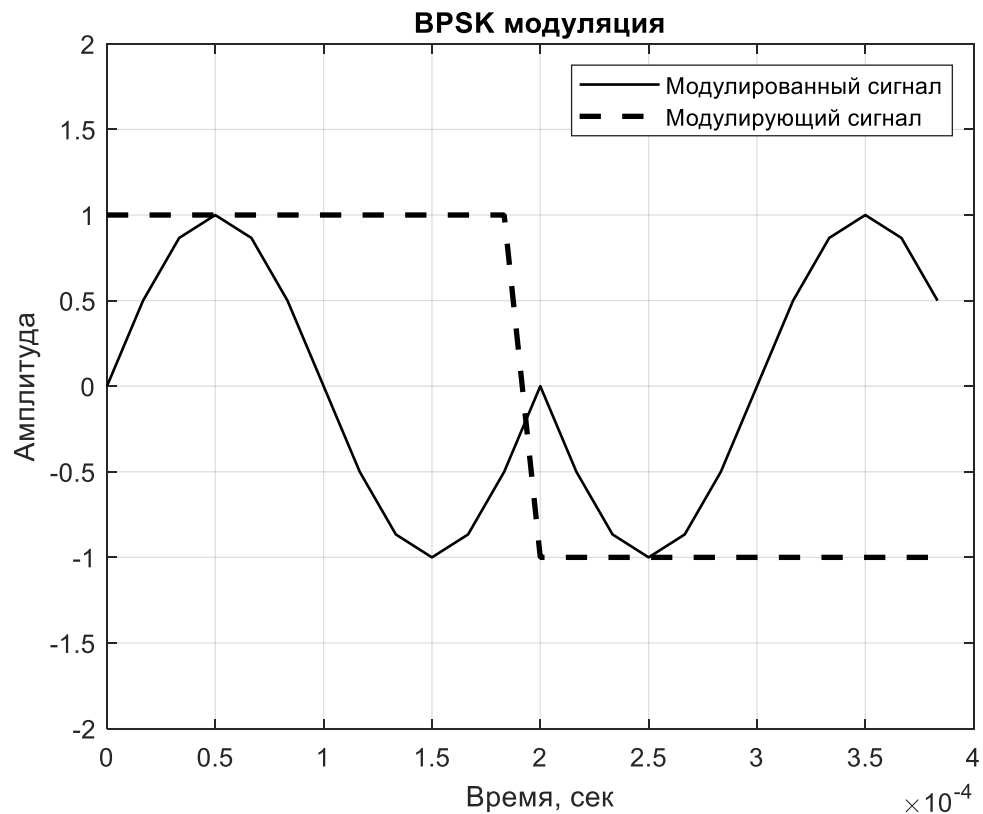


Рисунок 6 – Временная диаграмма последовательности Баркера из двух символов и сигнала, получившегося в результате выполнения программы

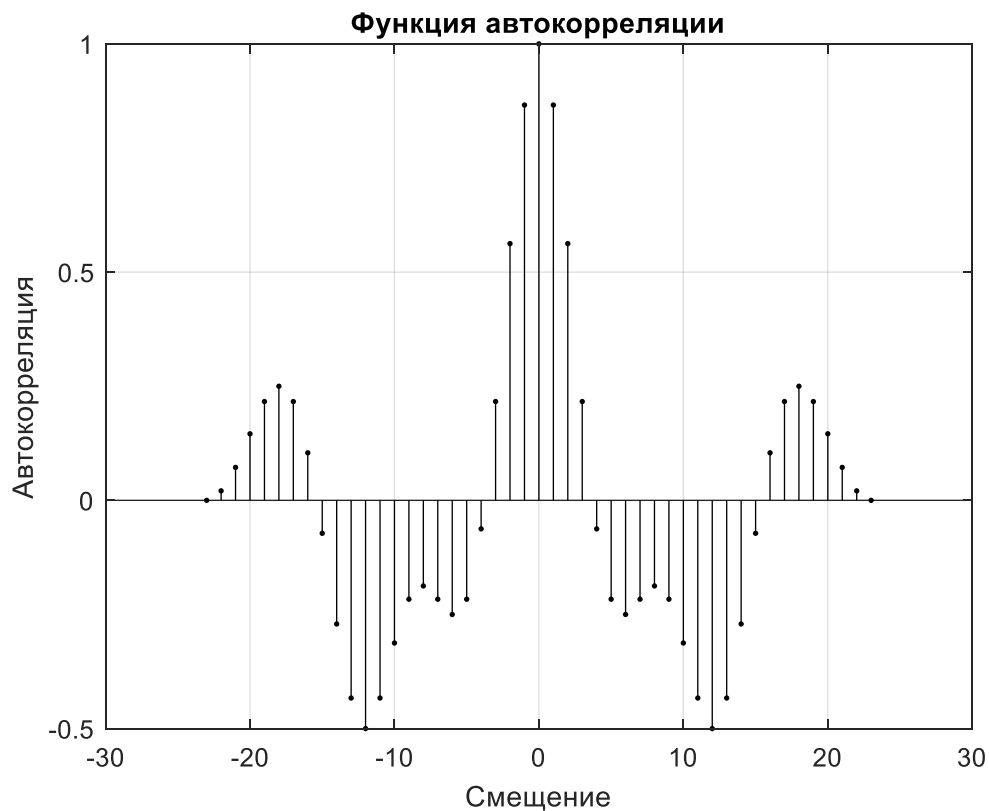


Рисунок 7 – Автокорреляционная функция сигнала, получившегося в результате выполнения программы

Результат, записанный в текстовый файл, представлен на рисунке 8:

```

bpskDAC.txt – Блокнот
Файл  Правка  Формат  Вид  Справка
Частота сигнала: 5000 Гц
Частота дискретизации: 60000 Гц
Значение ARR (при частоте тактирования шины 80 МГц): 1332
Количество элементов в массиве: 24

2048,3072,3822,4096,3822,3072,2048,1024,274,0,274,1024,2048,
1024,274,0,274,1024,2048,3072,3822,4096,3822,3072

Стр 1, стлб 1    100%    UNIX (LF)    UTF-8
  
```

Рисунок 8 – Содержимое текстового файла с результатами выполнения программы

2.2 Решение задачи трилатерации:

Для определения местоположения используются несколько излучателей (обычно два или три), которые находятся на известном расстоянии друг от друга. Когда источники излучают звуковые сигналы, они распространяется во все стороны, и приходят к приемнику в разное время.

Приемник записывает время прихода сигнала, и затем с помощью трилатерации можно определить расстояние до источников звука. Для этого необходимо провести окружности с центром в каждом источнике и радиусом, равным времени задержки сигнала между источником и приемником. Точка пересечения окружностей будет соответствовать местоположению приемника звука.

Для решения данной задачи нужно найти точку пересечения трёх окружностей, следовательно получаем систему из трёх уравнений окружностей, а именно (Формула 1):

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = r_1^2, \\ (x_2 - x)^2 + (y_2 - y)^2 = r_2^2, \\ (x_3 - x)^2 + (y_3 - y)^2 = r_3^2; \end{cases} \quad (1)$$

где $x_1, y_1, x_2, y_2, x_3, y_3$ — координаты динамиков, r_1, r_2 и r_3 — расстояния до соответствующих динамиков, x и y — координаты микрофона.

В процессе выражения из системы координат микрофона получим (Формула 2):

$$\begin{cases} x = \frac{C * E - F * B}{E * A - B * D}, \\ y = \frac{C * D - A * F}{B * D - A * E}; \end{cases} \quad (1)$$

где $A = 2(x_2 - x_1)$, $B = 2(y_2 - y_1)$, $C = r_1^2 - r_2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2$, $D = 2(x_3 - x_2)$, $E = 2(y_3 - y_2)$, $F = r_2^2 - r_3^2 - x_2^2 + x_3^2 - y_2^2 + y_3^2$.

Написана программа, рассчитывающая координаты приемника (Приложение Б). Результаты обработки данных о местоположении выводятся на экран компьютера (Рисунок 9, 10).

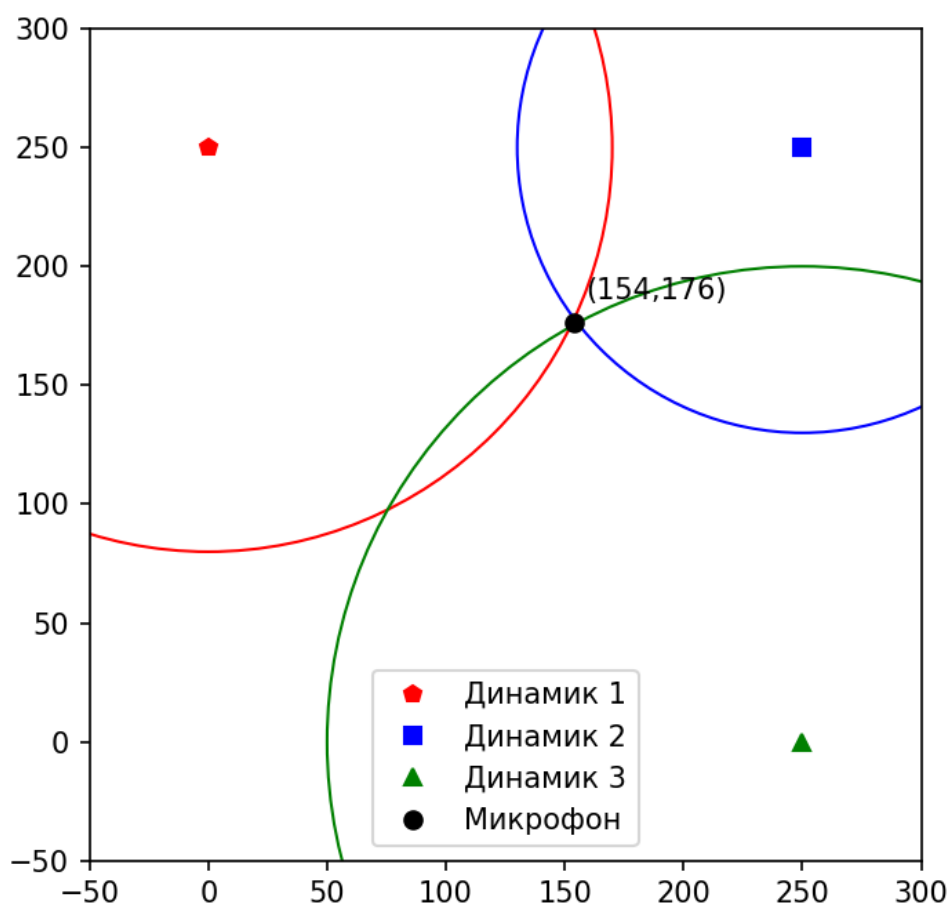


Рисунок 9 – Результат обработки данных при реальной координате микрофона примерно (152, 175)

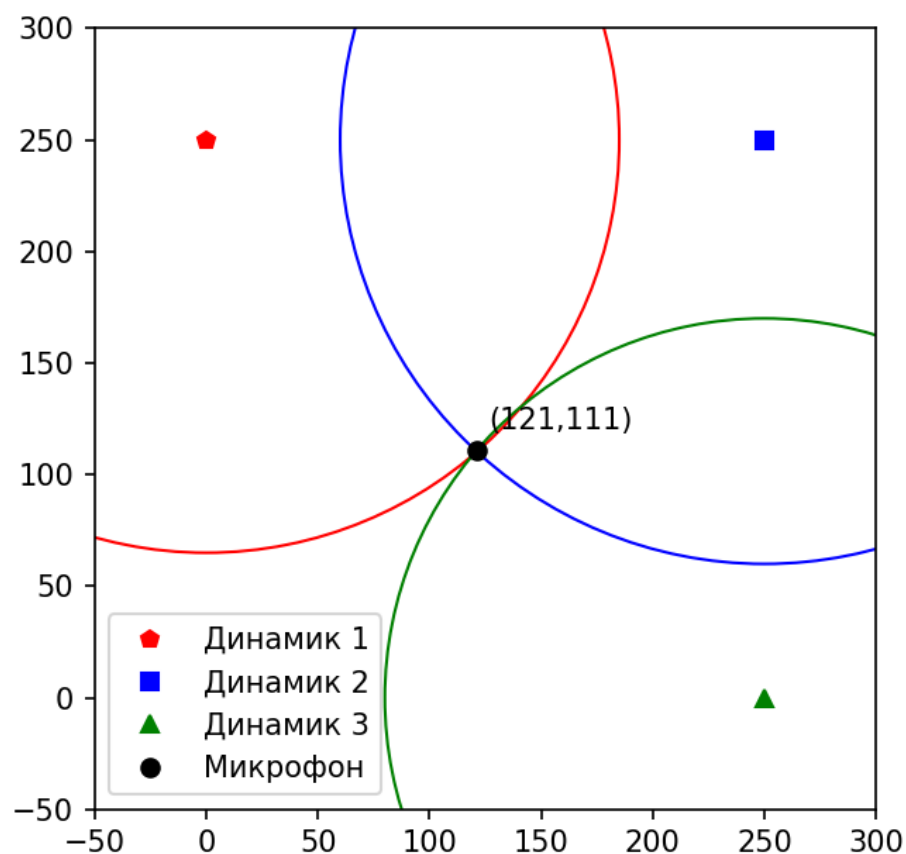


Рисунок 10 – Результат обработки данных при реальной координате микрофона примерно (120, 110)

В окне программы визуализируются следующие данные: заранее известные координаты источников сигнала, окружности с радиусами, равными измеренному с помощью макета расстоянию, точка пересечения окружностей и её координата, соответствующая искомой координате приемника сигнала.

Заключение

Во время преддипломной практики (в рамках работы над лабораторным макетом) была написана программа для генерации массива передаваемого с помощью ЦАП и программа, рассчитывающая координаты приемника по полученным с помощью устройства данным и визуализирующая их на координатной плоскости.

При дальнейшей работе над макетом необходимо правильно разделять передаваемые с контроллера данные, с целью того, чтобы номера динамиков в программе корректно определялись.

Список используемых источников

1. СПУТНИКОВАЯ РАДИОНАВИГАЦИОННАЯ СИСТЕМА "ГЛОНАСС": <https://cyberleninka.ru/article/n/sputnikovaya-radionavigatsionnaya-sistema-glonass-1>.
2. Сайт Роскосмоса: Статья про ГЛОНАСС — российская глобальная навигационная система: <https://www.roscosmos.ru/21923/>.
3. Статья про некогерентный прием сигналов: <https://cyberleninka.ru/article/n/primenenie-ustroystva-nekogerentnoy-demodulyatsii-v-tselom-fazomanipulirovannyh-signalov-v-radiosistemah-upravleniya/viewer>.
4. Прикладной потребительский центра ГЛОНАСС: ГЛОНАСС: <https://www.glonass-iac.ru/guide/glonass.php>.
5. Современные 32-разрядные ARM-микроконтроллеры серии STM32: цифроаналоговый преобразователь.
6. Reference manual на микроконтроллер STM32F40x.
7. Дубинин А.Е. Анализ фазовой модуляции при передаче сигналов Баркера. – Самара : СамГУПС, 2011.
8. Варакин Л. Е. Системы связи с шумоподобными сигналами. — М.: Радио и связь, 1985.
9. Триангуляционная система определения координат источника звука: <https://cyberleninka.ru/article/n/triangulyatsionnaya-sistema-opredeleniya-koordinat-istochnika-zvuka>

Приложение А

Код программы на языке MATLAB:

```
clear, clc, close all
load('BarkerCodes.mat');
% кодируемая последовательность
code = barker2;
% количество отсчетов одного периода синуса
m = 12;
% длительность импульса в периодах синусоиды
num_of_periods_per_bit = 1;
% частота сигнала в Гц
fc = 5000;
fs = fc*m;
ts = 0 : 1/fs : (m*length(code)*num_of_periods_per_bit)/fs-1/fs;
N = length(ts);
sinus = sin(2*pi*fc*ts);
% длина одного бита в отсчётах
n_for_bit = m*num_of_periods_per_bit;
% формируем модулирующий сигнал
fm = zeros(1,N);
for i=1:length(code)
    for j=n_for_bit*(i-1)+1:n_for_bit*i
        fm(j) = code(i);
    end
end
x = sinus.*fm;
filename = 'bpskDAC.txt'; % Имя файла
fid = fopen(filename, 'w'); % Открываем файл для записи
% Запись частоты сигнала, частоты дискретизации и значение ARR в файл
fprintf(fid, 'Частота сигнала: %d Гц\n', fc);
fprintf(fid, 'Частота дискретизации: %d Гц\n', fs);
ARR = round((80*10^6)/(fs))-1;
fprintf(fid, 'Значение ARR (при частоте тактирования шины 80 МГц): %d \n',
ARR);
fprintf(fid, 'Количество элементов в массиве: %d \n\n', length(x));
% запись отсчетов для 12 разрядного DAC в текстовый файл
dlmwrite(filename, round(2048*x)+2048,'-append', 'delimiter', ',');
fclose(fid);
% построение графиков
plot(ts,x,'black','LineWidth',1), grid on, hold on;
plot(ts,fm,'--black','LineWidth',2), grid on;
title('BPSK модуляция');
ylim([-2 2]);
xlabel('Время, сек'), ylabel('Амплитуда');
legend({'Модулированный сигнал','Модулирующий сигнал'});
% Вычисление автокорреляционной функции с помощью функции myAutocorr
autocorr = Autocorr(x);
% Создание оси времени
time = -(length(x)-1):(length(x)-1);
% Построение графика автокорреляционной функции
figure;
stem(time, autocorr, '.', 'black', 'LineWidth', 0.5);
xlabel('Смещение');
ylabel('Автокорреляция');
title('Функция автокорреляции');
grid on;
```

Приложение Б

Код на языке Python, определяющий координату и визуализирующий ее на координатной плоскости:

```
import math
import serial
import matplotlib.pyplot as plt
from matplotlib.patches import Circle
import time

# Известные точки и их координаты
point1 = (0, 250)
point2 = (250, 250)
point3 = (250, 0)

# Подключение к UART порту
ser = serial.Serial('COM30', 9600, timeout=0) # Замените 'COM1' на ваше имя
порта и укажите правильную скорость передачи данных
# Функция для вычисления координат неизвестной точки
def trilaterate(p1, p2, p3, d1, d2, d3):
    x1, y1 = p1
    x2, y2 = p2
    x3, y3 = p3
    A = 2 * (x2 - x1)
    B = 2 * (y2 - y1)
    C = d1**2 - d2**2 - x1**2 + x2**2 - y1**2 + y2**2
    D = 2 * (x3 - x2)
    E = 2 * (y3 - y2)
    F = d2**2 - d3**2 - x2**2 + x3**2 - y2**2 + y3**2
    x = (C*E - F*B) / (E*A - B*D)
    y = (C*D - A*F) / (B*D - A*E)
    return x, y

# График
fig, ax = plt.subplots()
# Отображение известных точек
ax.plot(point1[0], point1[1], 'rs', label='Динамик 1')
ax.plot(point2[0], point2[1], 'bo', label='Динамик 2')
ax.plot(point3[0], point3[1], 'g^', label='Динамик 3')
# Отображение орбит
circle1 = Circle(point1, 0, fill=False, color='r')
circle2 = Circle(point2, 0, fill=False, color='b')
circle3 = Circle(point3, 0, fill=False, color='g')
ax.add_patch(circle1)
ax.add_patch(circle2)
ax.add_patch(circle3)
# Отображение неизвестной точки
unknown_point, = ax.plot([], [], 'ko', label='Микрофон')
# Настройка графика
```

```

ax.set_aspect('equal')
ax.set_xlim([-50, 300])
ax.set_ylim([-50, 300])
ax.legend()
# Функция для обновления графика
def update_plot(distances):
    # Вычисление координат неизвестной точки
    result = trilaterate(point1, point2, point3, distances[0], distances[1],
distances[2])
    # Обновление орбит и координат неизвестной точки
    circle1.set_radius(distances[0])
    circle2.set_radius(distances[1])
    circle3.set_radius(distances[2])
    unknown_point.set_data(result[0], result[1])
    # Обновление графика
    plt.draw()
    plt.pause(0.01)
# Чтение и обновление расстояний от UART порта каждую секунду
while True:
    # Чтение расстояний от UART порта
    distances = []
    while len(distances) < 3:
        if ser.in_waiting > 0:
            # Чтение данных из порта
            data = ser.readline()
            data = data.decode('utf8')
            try:
                distance = float(data) # Преобразовать прочитанные данные в
число
                distances.append(distance)
            except ValueError:
                continue
            print(data)
    # Обновление графика
    update_plot(distances)
    # Пауза в 1 секунду
    time.sleep(0.5)
# Закрыть соединение с UART портом
ser.close()

```