

РАЗРАБОТКА КОМПЬЮТЕРНОЙ МОДЕЛИ ЦИФРОВОГО КАНАЛА СВЯЗИ С ASK В ПРОГРАММНОЙ СРЕДЕ LABVIEW

А. Е. КУРОЧКИН, ТУН КХАНТ

Белорусский государственный университет информатики и радиоэлектроники
(г. Минск, Республика Беларусь)

E-mail: kurochkin@bsuir.by

Аннотация. Представлены результаты разработки компьютерной модели цифрового канала связи с ASK в LabVIEW.

Annotation. The results of developing a computer model of a digital communication channel with ASK in LabVIEW are presented.

Введение

Цель разработки – создать компьютерную модель цифрового канала связи с амплитудной манипуляцией (ASK) для передачи текстовой информации, не применяя штатные средства LabVIEW [1]. Дополнительное условие – оценка качества передачи с помощью глазковой диаграммы. Амплитудная манипуляция нашла применение в волоконно-оптических линиях связи (ВОЛС), так как они гораздо меньше подвержены амплитудным помехам. При этом под амплитудой в оптическом сигнале понимается сила света. Таким образом, наличие или отсутствие светового импульса будет соответствовать двум значениям цифрового потока: 1 или 0, причём манипуляция обеспечивается путём прямой модуляции тока лазерного диода. Последовательность бинарных символов может быть представлена различными способами в виде последовательности оптических импульсов. Самым простым способом является однополярный NRZ (Non Return to Zero) - без возврата к нулю.

Разработка интерфейса модуля в LabVIEW

На рисунке 1 представлена передняя панель (Front Panel) разработанного модуля.

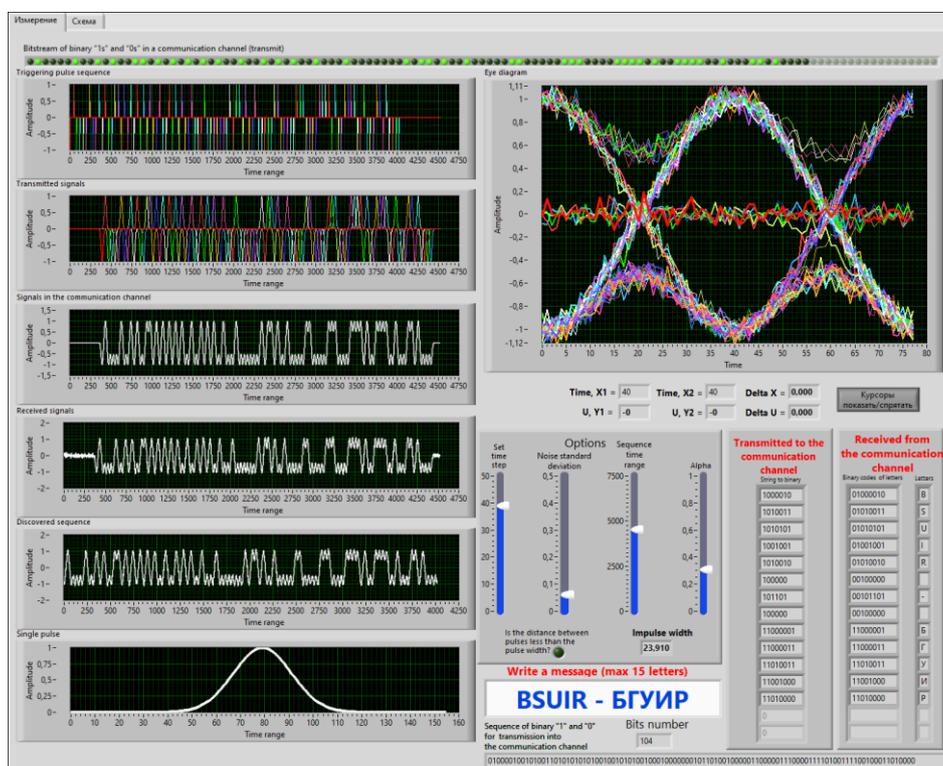


Рис. 1. Интерфейс программного модуля

На панели размещены: семь элементов “Waveform Graph” палитры Graph для отображения одиночного импульса Гаусса, первичной последовательности импульсов запуска, сформированной последовательности гауссовых импульсов, импульсов на входе линии связи, импульсов на выходе линии связи, импульсов на входе демодулятора, глазковой диаграммы, а также светодиодный массив на основе элемента “Round LED” для отображения цифрового потока и строковый элемент String для отображения последовательности передаваемых ASCII кодов элементов цифрового потока. Для изменения временных параметров передаваемой последовательности импульсов, уровня помех в линии связи, временного диапазона анализа, параметров импульса Гаусса на панели размещены четыре элемента “Vertical Pointer Slide” палитры Numeric. Для отображения ASCII кодов на входе и выходе линии связи, а также декодированных символов на панели размещены три индикатора соответствующих массивов на основе элемента “Numeric Indicator” палитры Numeric. Для ввода передаваемого текста (не более 15 символов) служит элемент “String Control” палитры “String&Path”.

Разработка блок-схемы модуля

На рисунке 2 представлена разработанная в LabVIEW блок-схема модуля. Номерами от 1 до 9 на блок-схеме обозначены основные функциональные группы элементов, с помощью которых выполняются перечисленные ниже действия.

1. Формирование исходной последовательности нулей (0) и единиц (1):
 - используя функцию “String To Byte Array Function”, преобразуем строку в массив беззнаковых байтов, где каждый байт в массиве имеет ASCII-код соответствующего символа в строке;
 - с помощью цикла “For Loop”, используя функции “Number To Boolean Array Function” и “Reverse 1D Array Function” формируем реверсивный массив логических переменных;
 - с помощью функций “Array Size Function”, которая возвращает количество элементов в каждом измерении массива, “Multiply Array Elements Function”, возвращающую произведение всех элементов числового массива, и “Reshape Array Function” формируем последовательность логических переменных;
 - функцией “Boolean To (0,1) Function” формируем последовательность нулей (0) и единиц (1).
2. Формирование передаваемой последовательности нулей (0) и единиц (1) в виде кодов ASCII и символов для их отображения на индикаторах:
 - с помощью функции “String Length Function” определяем длину передаваемого текста;
 - с помощью циклов “For Loop” и функций “Multiply Function”, “Add Function”, “Index Array Function”, “Boolean To (0,1) Function”, “Index Array Function”, “Number To Decimal String Function”, “Concatenate Strings Function” разбираем строку символов на октеты и формируем символьные образы кодов ASCII;
 - с помощью дополнительного цикла “For Loop”, константы “format string” и функций “Format Into String”, “Concatenate Strings” форматируем и отображаем передаваемое сообщение на индикаторе строки String.
3. Формирование последовательности первичных импульсов с помощью цикла “For Loop”, используя элемент “Impulse Pattern VI” палитры “Signal Generation”.
4. Формирование шаблона гауссова импульса с помощью структуры “Formula Node” и кода на языке программирования C.
5. Формирование последовательности гауссовых импульсов с помощью цикла “For Loop” и элемента “FIR Filter VI” палитры Filters. Входная последовательность первичных импульсов с помощью КИХ-фильтра, заданного КИХ-коэффициентами гауссова импульса, на выходе приобретает форму последовательности одиночных гауссовых импульсов в соответствии с передаваемой информацией: 0 – импульса нет, 1 – импульс есть.
6. Формирование временной последовательности гауссовых импульсов с помощью функции “Initialize Array” и цикла “For Loop”, используя суммирование одиночных гауссовых импульсов по амплитуде с учётом времени появления импульсов в последовательности.
7. Формирование модели реального канала связи с помощью функции “Add Function” и элемента “Gaussian White Noise VI” палитры “Signal Generation”. Выходной процесс – сумма последовательности гауссовых импульсов и псевдослучайного белого шума с распределением по Гауссу.
8. Формирование глаз-диаграммы с помощью функций “Array Subset Function”, “Divide Function” и “Reshape Array Function”.
9. Реализация демодулятора октетов:
 - с помощью цикла “For Loop” задаём анализ всех переданных октетов, содержащих коды ASCII;

- с помощью дополнительного цикла “For Loop” на каждой итерации предыдущего цикла анализируем каждый бит текущего октета с помощью функций “Index Array Function”, “Greater? Function” и “Select Function”;
- с помощью цикла “For Loop” из каждого октета с помощью функций Index Array Function, Number To Decimal String Function, Concatenate Strings Function формируем бинарные коды символов и отображаем результат на индикаторе массива октетов String в виде ASCII кодов;
- с помощью цикла “For Loop” каждый числовой октет нулей и единиц с помощью функции “Equal? Function” преобразуем в октет логических констант True и False;
- с помощью функций “Reverse 1D Array”, “Boolean Array To Number”, “To Unsigned Byte Integer” формируем символы, соответствующие принятым ASCII кодам, и отображаем на индикаторе массива символов String.

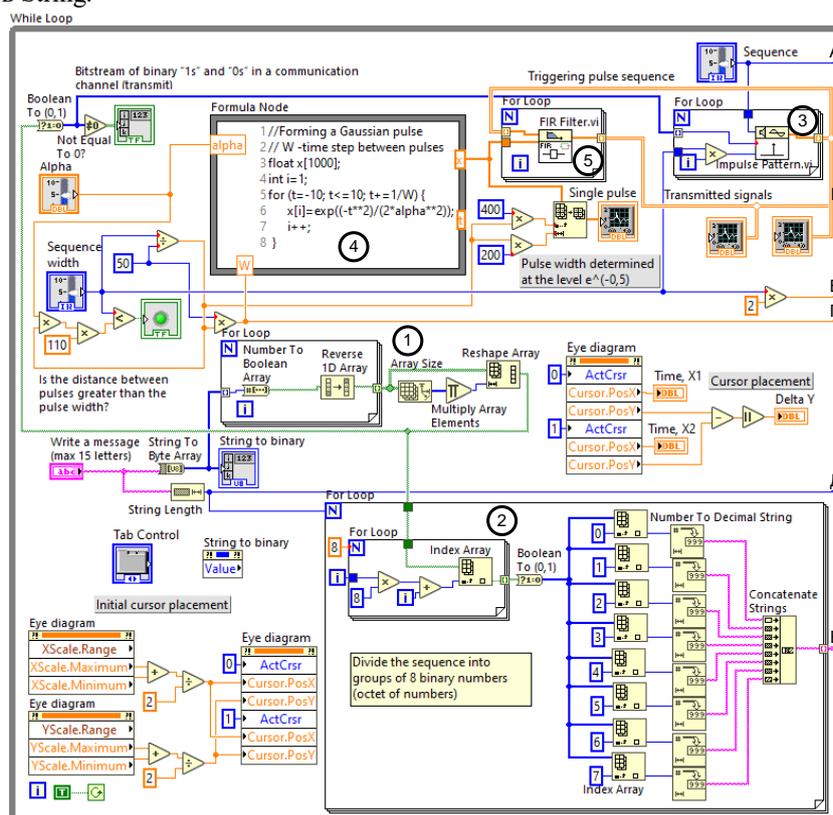


Рис. 2. Блок-схема модуля

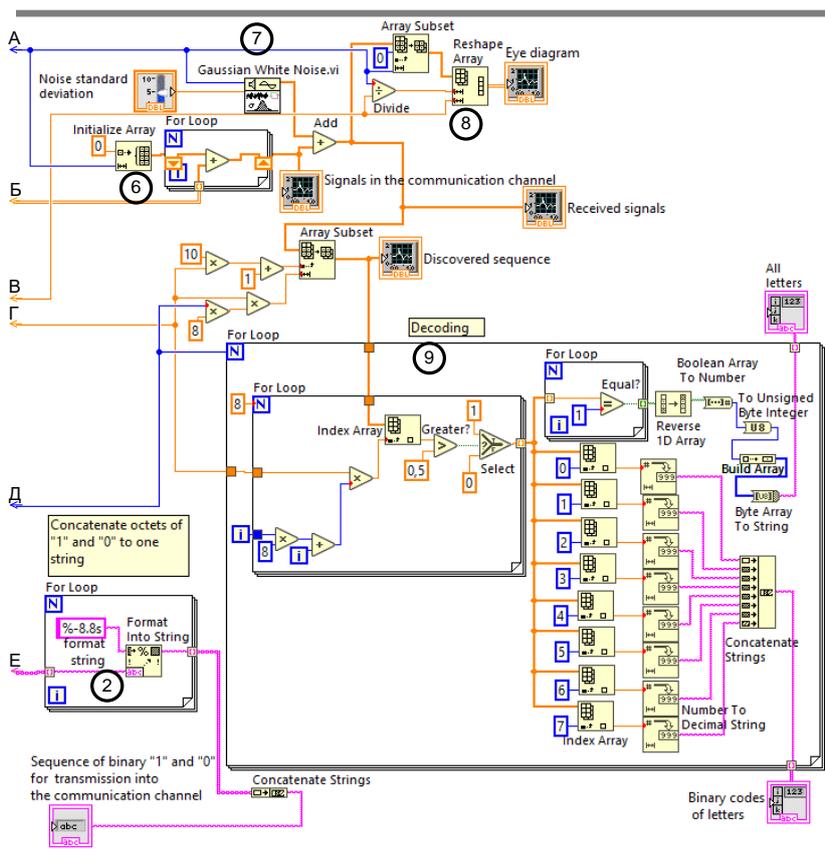
В ВОЛС для описания изменения формы импульса в оптической среде часто используется гауссов импульса. Огибающую импульса гауссовой формы с точностью до постоянного множителя можно записать в виде $U(t) = \exp(-t^2 / 2\alpha^2)$. Постоянная α имеет смысл половины длительности импульса, определяемой на уровне $e^{-1/2}$ от амплитуды импульса (рисунок 1, к). Таким образом, полная длительность импульса $t_{и}=2\alpha$. В блок-схеме для формирования импульса гауссовой формы использована структура “Formula Node” и следующий код:

```
//Forming a Gaussian pulse; W -time step between pulses
float x[1000];
int i=1;
for (t=-10; t<=10; t+=1/W) {
    x[i]=exp((-t**2)/(2*alpha**2));
    i++;}

```

Разработка формирователя глазковой диаграммы

Глазковая диаграмма – это широко распространенный графический способ оценки качества сигналов в цифровых системах передачи информации. Глазковая диаграмма формируется путём наложения множества положительных и отрицательных переходов импульсов из одного состояния в другое в соответствии с передаваемыми битами (0 или 1 на рисунке 3, а–з).



Продолжение рис. 2

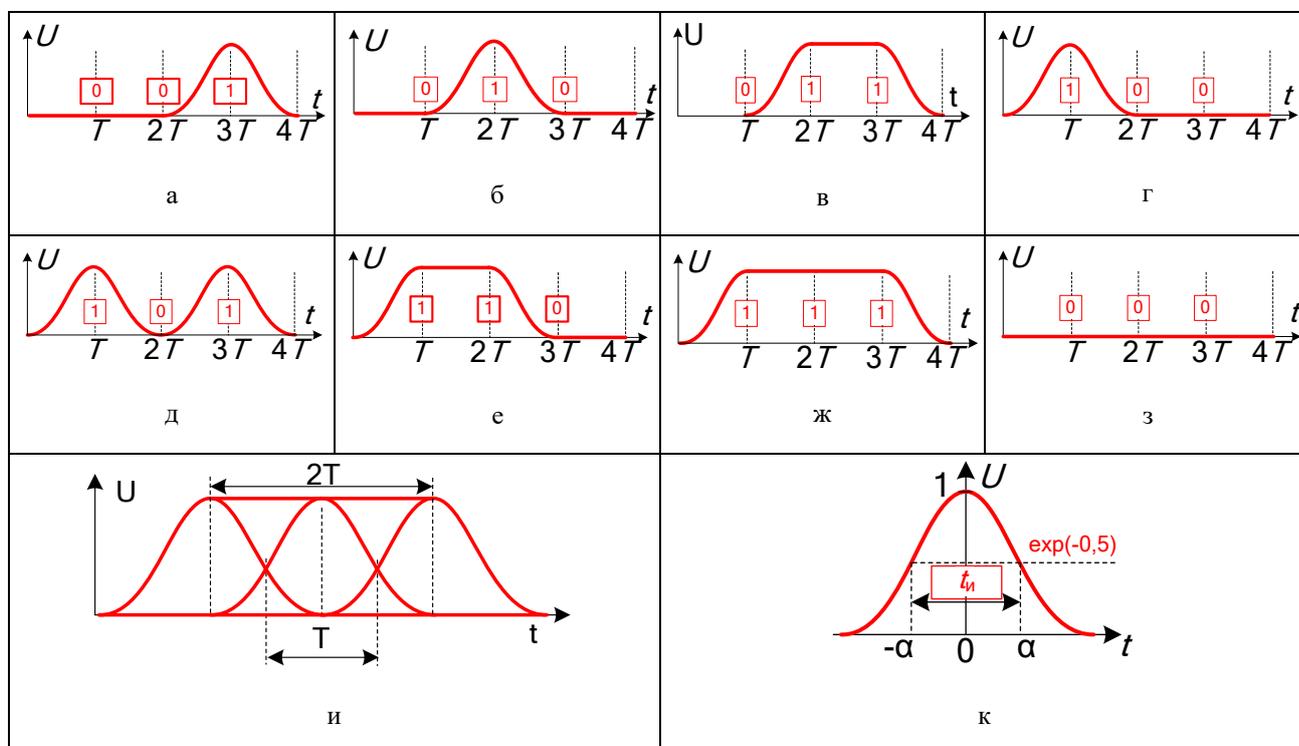


Рис. 3. Формирование глаз-диаграммы

После задержки на время, кратное T , и сложения импульсов получающиеся в результате изображение похоже на раскрытый глаз (рисунок 4, и). Для построения диаграммы достаточно передаваемый сигнал разрезать на части, кратные по времени длительности $2T$ и с помощью функции

Секция 3 «Цифровая обработка сигналов и машинное обучение»

“Reshape Array Function” сформировать новые массивы данных. При этом элемент “Waveform Graph” отображающий диаграмму должен работать в режиме “Autoscale X”.

Параметрами измерения “глаза” при статистическом анализе являются:

- нулевой уровень, среднее значение логического нуля глазковой диаграммы;
- единичный уровень, среднее значение уровня логической единицы глазковой диаграммы;
- битовый период, разница во времени между средними значениями двух соседних точек пересечения;
- амплитуда глаза, разница между средними значениями единичного и нулевого уровней;
- высота глаза, вертикальный раскрыв глаза, т.е. разность амплитуды глаза и стандартных отклонений нулевого и единичного уровней;
- ширина глаза, горизонтальный раскрыв глаза, т.е. разность двух соседних точек пересечения за вычетом стандартных отклонений фронтов;
- точка пересечения глаза, в которой переходы от нуля к единице и от единицы к нулю достигают одинаковой амплитуды (точка пересечения восходящего и спадающего фронтов), выражается в процентах от общей амплитуды глаза и для идеального глаза составляет 50% амплитуды.

В идеальном случае глазковые диаграммы выглядят как прямоугольники. Но каналы связи не идеальны, поэтому из-за амплитудных и фазовых изменений в условиях воздействия помех и шумов переходы по времени не совпадают друг с другом. Различия по времени и амплитуде от бита к биту приводят к уменьшению раскрыва глаза. Так называемая маска глазковой диаграммы для анализа и оценки качества цифрового потока, изображена на рисунке 4. В соответствии с [2], например, для системы PON (Passive Optical Network – пассивная оптическая сеть) при скорости передачи (нисходящее/восходящее направление) 155,52 Мбит/с/155,52 Мбит/с и расчетном значении коэффициента битовых ошибок (BER) не хуже 1×10^{-10} в экстремальном случае затухания оптического тракта и условий дисперсии маска должна соответствовать соотношениям: $x1/x4 - 0,15/0,85$; $x2/x3 - 0,35/0,65$; $y1/y2 - 0,2/0,8$.



Рис. 4. Маска глаз-диаграммы

Заключение

В результате разработки получена удобная учебная версия модели ВОЛС без применения стандартных элементов, имеющихся на палитрах Modulation и Demodulation дополнительного к LabVIEW пакета Modulation Toolkit.

Список использованных источников

1. Евдокимов, Ю.К. LabVIEW для радиоинженера: от виртуальной модели до реального прибора / Ю. Евдокимов, В.Р. Линдваль, Г.И. Щербаков. - Москва : ДМК Пресс, 2009. - 400 с.
2. Рекомендация МСЭ-Т G.983.1 Оптические системы широкополосного доступа, базирующиеся на пассивной оптической сети (PON) [Электронный ресурс] - Режим доступа: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-G.983.1-200501-1!!PDF-R&type=items – Дата доступа: 15.07.2024.
3. Игнатов, А. Методы измерения шума в цифровых схемах [Электронный ресурс] / А. Игнатов. - Режим доступа: https://russianelectronics.ru/files/48468/EK2009_12_74-75.pdf. – Дата доступа: 15.07.2024.