Mini Project Report- Applied SDLC and Software Testing

Team 15

Digital Modulation Tools

L&T Technology and Services

Contents

Introduction

Digital Modulation Tools is a simple console application designed to give the user a simulation of their data and the possible modulations and encoding. The bit stream given can be modulated in various modulation techniques like - ASK, FSK, PSK, etc. It is developed using C programming language.

Digital Modulation Concepts and Types:

Analog to Digital Modulation Techniques

Amplitude Shift Keying

ASK is a type of modulation where the digital signal is represented as a change in amplitude. In order to carry out amplitude shift keying, we require a carrier signal and a binary sequence signal. It is also known as On-Off keying. This is because the carrier waves switch between 0 and 1 according to the high and low level of the input signal.

Frequency Shift Keying

Frequency Shift Keying FSK is the digital modulation technique in which the frequency of the carrier signal varies according to the digital signal changes. FSK is a scheme of frequency modulation.
The output of a FSK modulated wave is high in frequency for a binary High input and is low in frequency for a binary Low input. The binary 1s and 0s are called Mark and Space frequencies.

Binary Phase Shift Keying

Phase Shift Keying PSK is the digital modulation technique in which the phase of the carrier signal is changed by varying the sine and cosine inputs at a particular time. This is also called as 2-phase PSK or Phase Reversal Keying. In this technique, the sine wave carrier takes two phase reversals such as 0° and 180°.
BPSK is basically a Double Side Band Suppressed Carrier DSBSC modulation scheme, for message being the digital information.

Quadrature Amplitude Modulation

The creation of symbols that are some combination of amplitude and phase can carry the concept of transmitting more bits per symbol further. This method is called quadrature amplitude modulation (QAM).
While QAM is enormously efficient of spectrum, it is more difficult to demodulate in the presence of noise, which is mostly random amplitude variations.
Linear power amplification is also required.

Phase Shift Keying

Digital to Digital Modulation Techniques

NRZ

- NRZ stands for Non-return zero.
- In NRZ encoding, the level of the signal can be represented either positive or negative.

NRZ-L

- In NRZ-L encoding, the level of the signal depends on the type of the bit that it represents.
- If a bit is 0 or 1, then their voltages will be positive and negative respectively. Therefore, we can say that the level of the signal is dependent on the state of the bit.

NRZ-I

- NRZ-I is an inversion of the voltage level that represents 1 bit.
- In the NRZ-I encoding scheme, a transition occurs between the positive and negative voltage that represents 1 bit.
- In this scheme, 0 bit represents no change and 1 bit represents a change in voltage level.

RZ

- RZ stands for Return to zero.
- There must be a signal change for each bit to achieve synchronization. However, to change with every bit, we need to have three values: positive, negative and zero.
- RZ is an encoding scheme that provides three values, positive voltage represents 1, the negative voltage represents 0, and zero voltage represents none.
- In the RZ scheme, halfway through each interval, the signal returns to zero.
- In RZ scheme, 1 bit is represented by positive-to-zero and 0 bit is represented by negative-to-zero.

Bi-phase

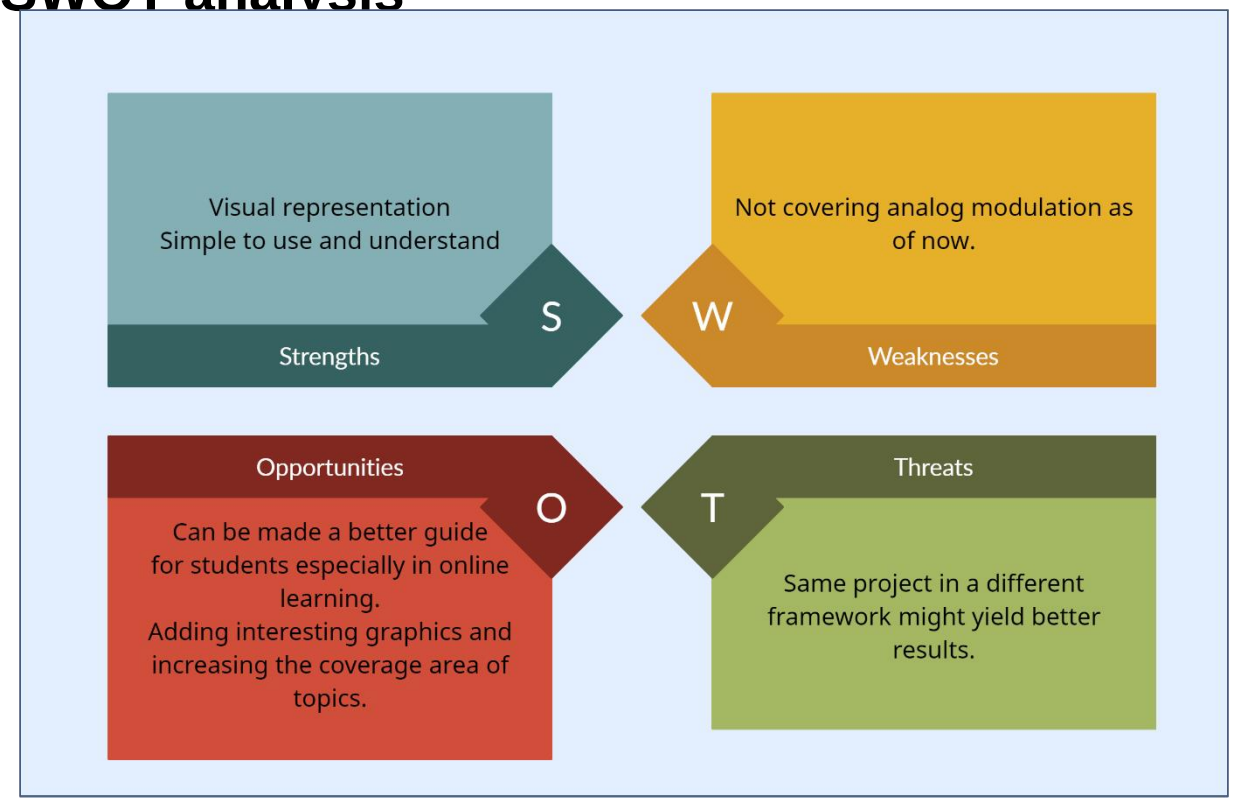
Manchester Encoding

- It changes the signal at the middle of the bit interval but does not return to zero for synchronization.
- In Manchester encoding, a negative-to-positive transition represents binary 1, and positive-to-negative transition represents 0.
- Manchester has the same level of synchronization as RZ scheme except that it has two levels of amplitude.

Differential Manchester Encoding

- It changes the signal at the middle of the bit interval and here the presence or absence of the transition at the beginning of the interval determines the bit.
- A transition means binary 0 and no transition means binary 1.
- Two signal changes represent 0 and one signal change represent 1.

SWOT analysis



4W's and 1H

Who:

- Students and curious learners who wish to see how digital signals are encoded.

What:

- A console based simulating tool covering digital modulation concepts.

When:

- On encountering troubles in understanding digital modulation concepts.

Where:

- A handy tool that can be accessed from anywhere .

How:

- One-to-one mapping will be used for displaying the modulated waveforms and bitstreams and also a database of circuits used for the respective purpose will be prepared which would be displayed as and when needed and thus guiding the user to a complete extent.

Detail requirements

High Level Requirements:

ID	Description	Category	Status
HLR01	User Interface	Technical	Done
HLR02	Digital to Analog Modulation Techniques	Technical	Done
HLR02	Digital to Digital Modulation Techniques	Technical	Done

Low level Requirements:

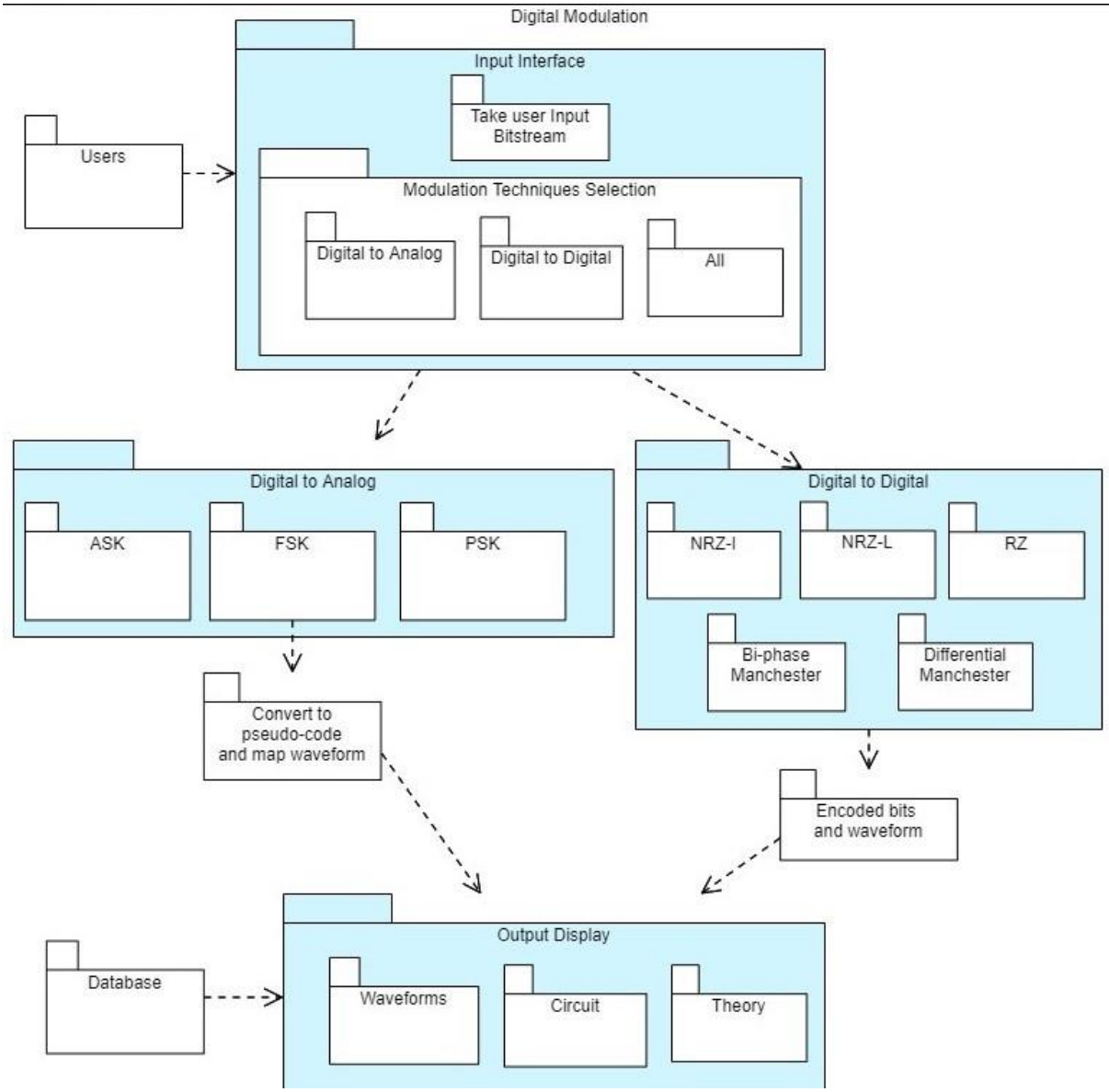
ID	Description	HLR ID	Status
LR00	Input Interface	HLR01	Done
LR01	Output Display	HLR01	Done
LR02	Displaying theory of circuits	HLR01	Done
LR03	Amplitude Shift Keying	HLR02	Done

ID	Description	HLR ID	Status
LR04	Frequency Shift Keying	HLR02	Done
LR05	Phase Shift Keying	HLR02	Done
LR06	Quadrature Amplitude Modulation	HLR02	Done
LR07	Non Return to Zero - I	HLR03	Done
LR08	Non Return to Zero - L	HLR03	Done
LR09	Return to Zero	HLR03	Done
LR10	Biphase Manchester	HLR03	Done
LR11	Differential Manchester	HLR03	Done

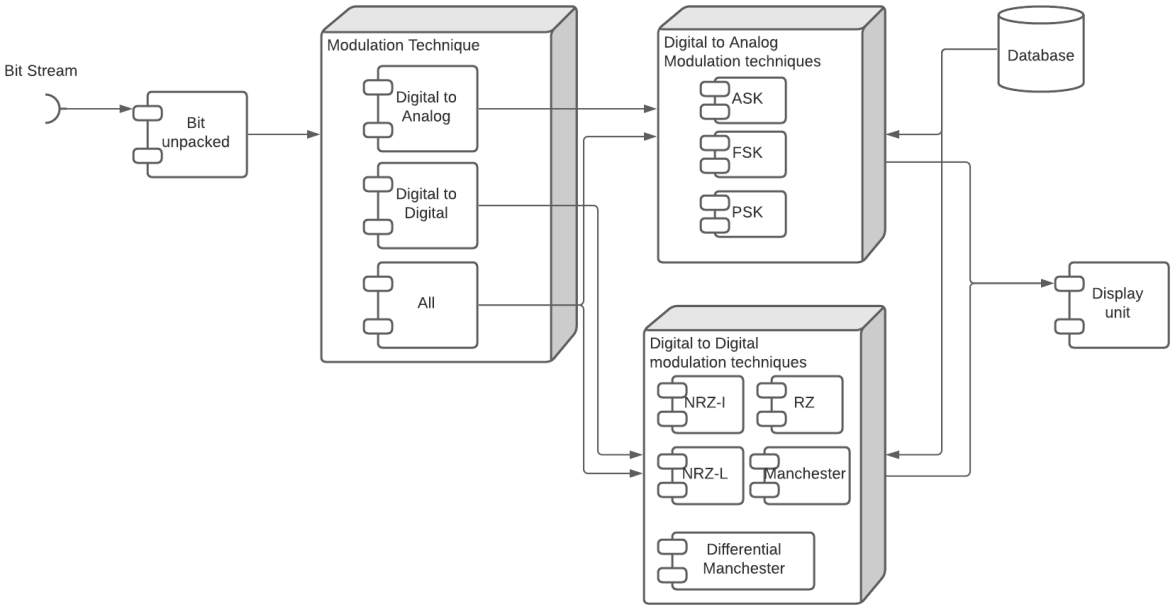
Design

High Level Requirements UML Diagrams

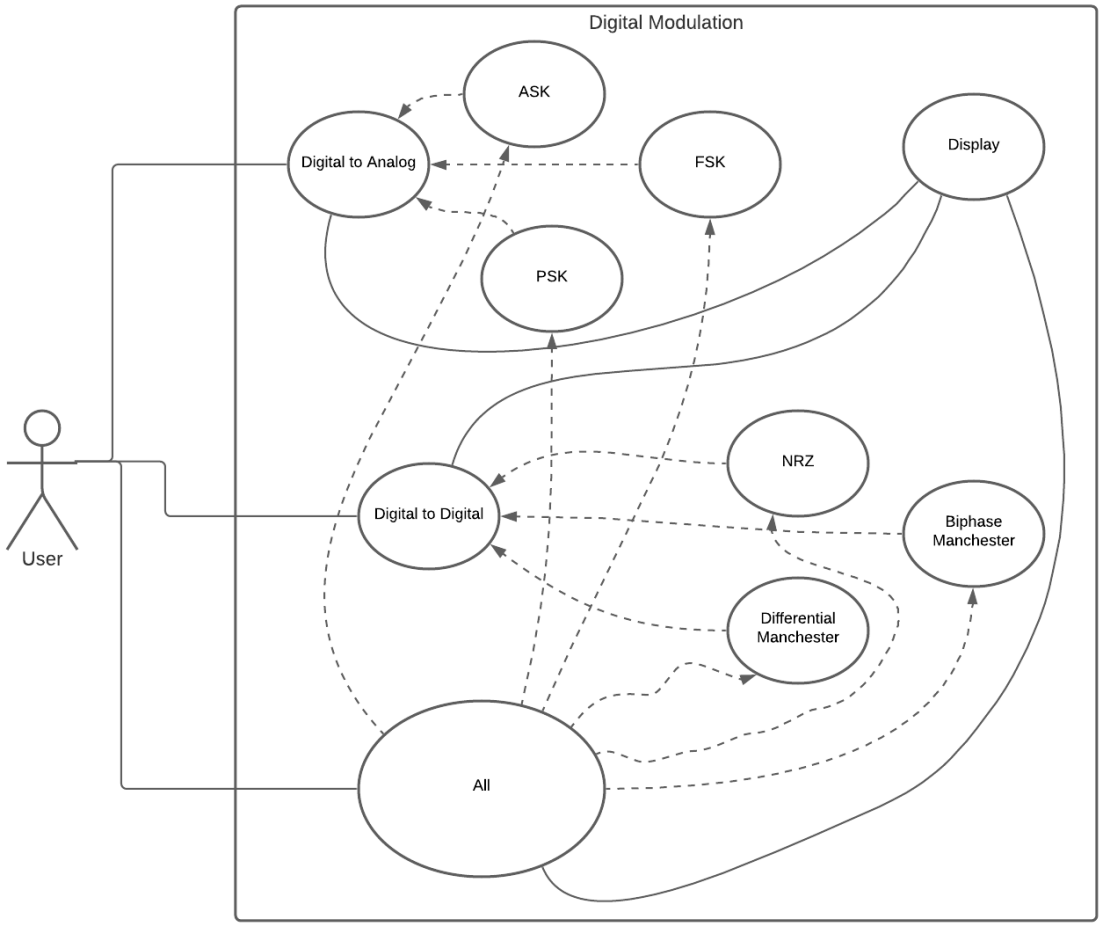
Package Diagram



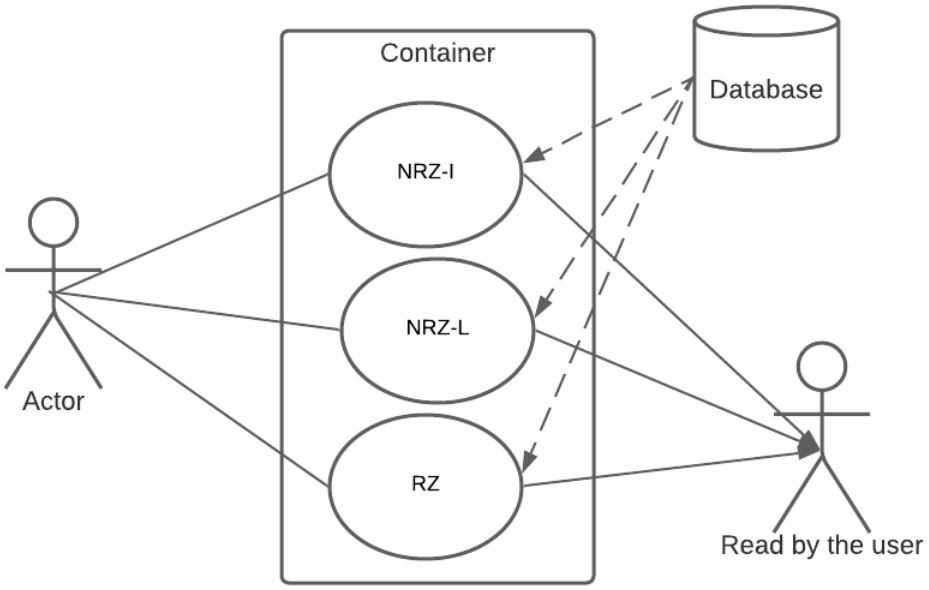
Component Diagram


Bit Stream
Bit unpacked
Modulation Technique
Digital to Analog
Digital to Digital
All
Digital to Analog Modulation techniques
ASK
FSK
PSK
Database
Display unit
Digital to Digital modulation techniques
NRZ-I
RZ
NRZ-L
Manchester
Differential Manchester

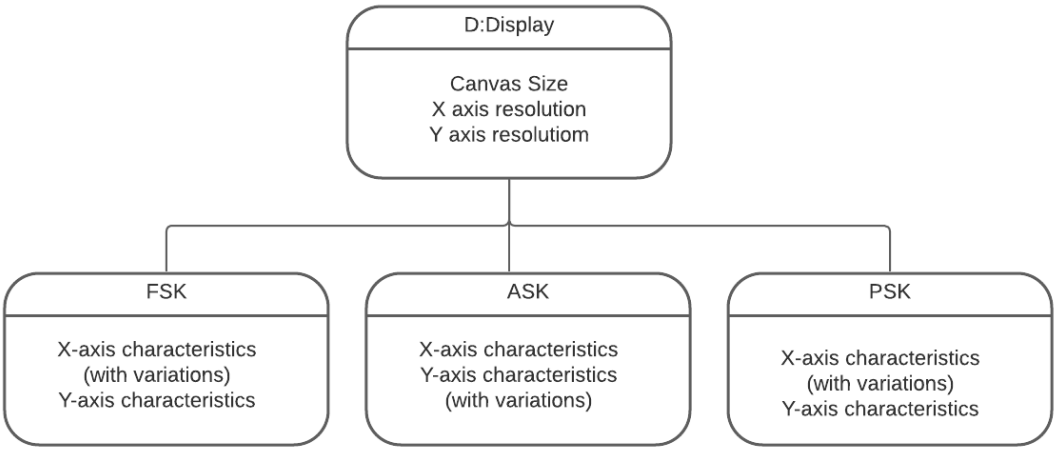
Use Case


Digital Modulation
ASK
Digital to Analog
FSK
Display
PSK
NRZ
Digital to Digital
Biphase Manchester
Differential Manchester
All
User

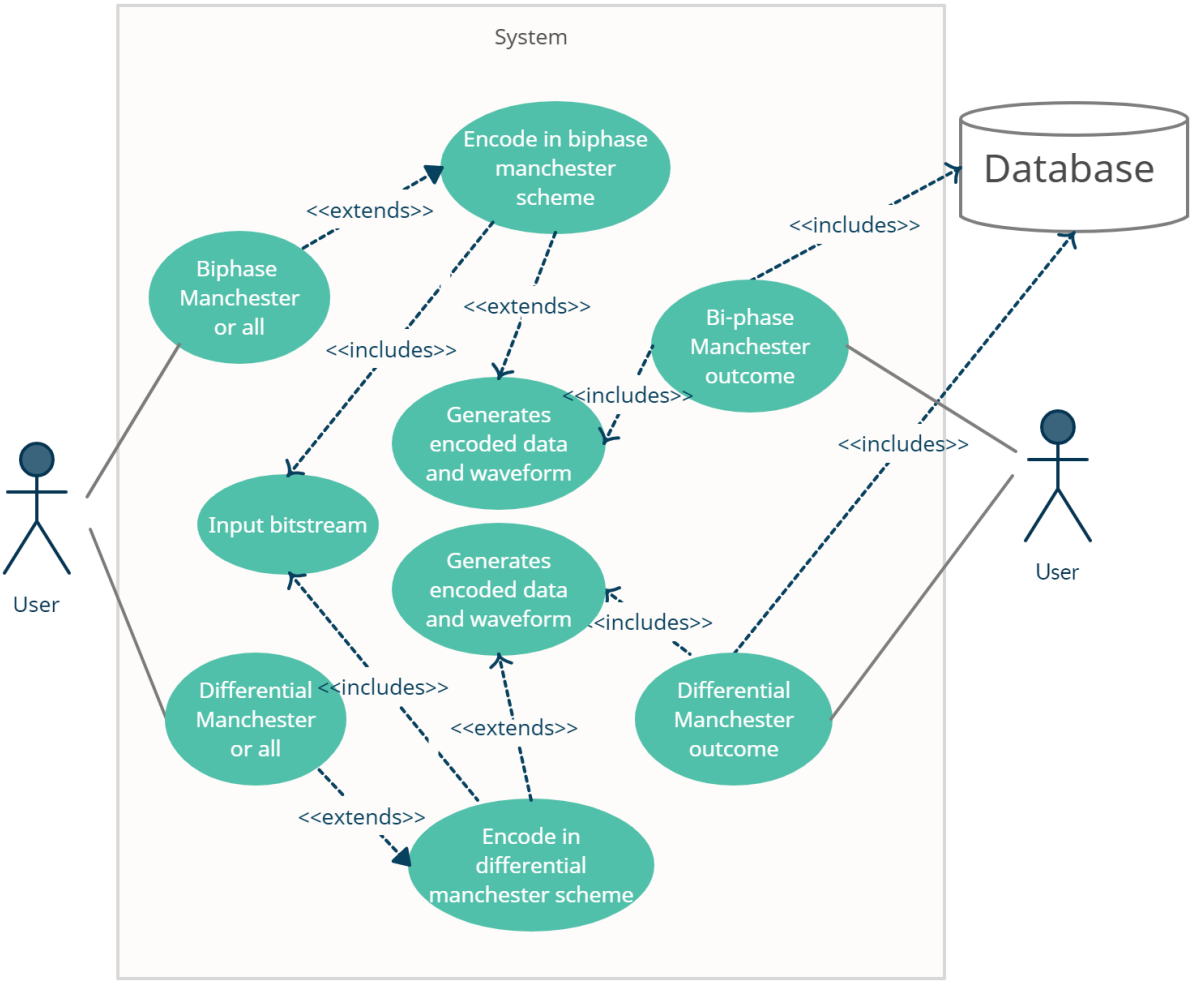
NRZ and RZ Use Case Diagram



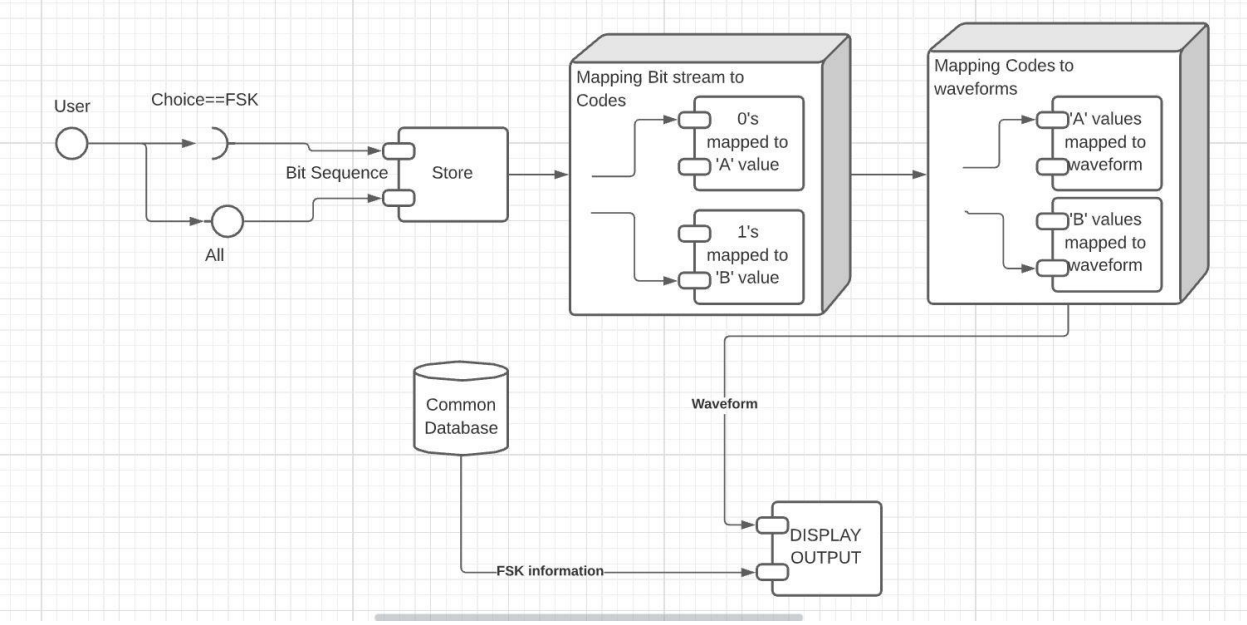
Display Structural Diagram



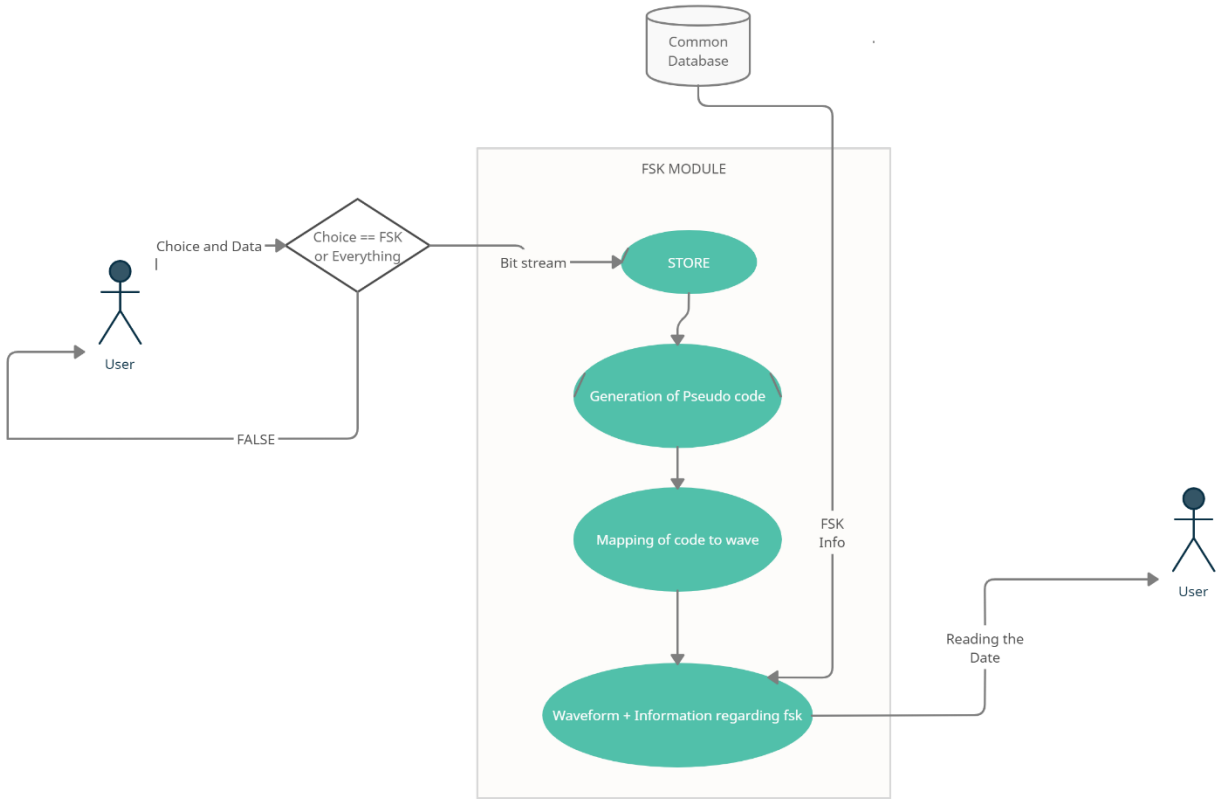
Manchester Use Case Diagram



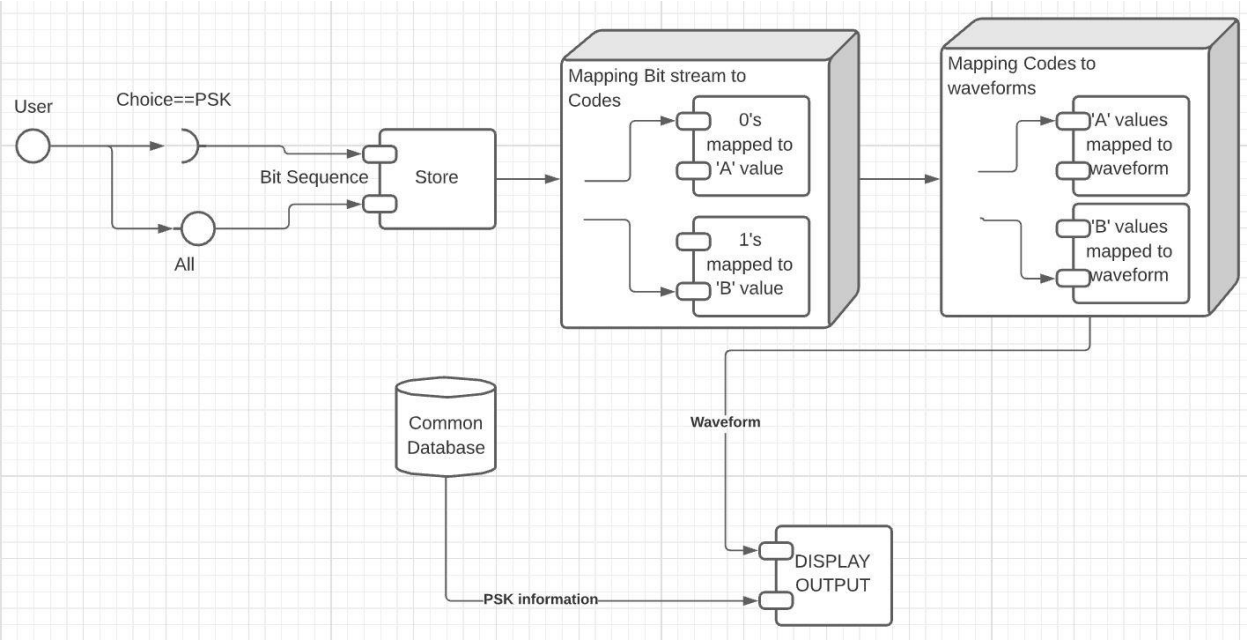
FSK Component Diagram



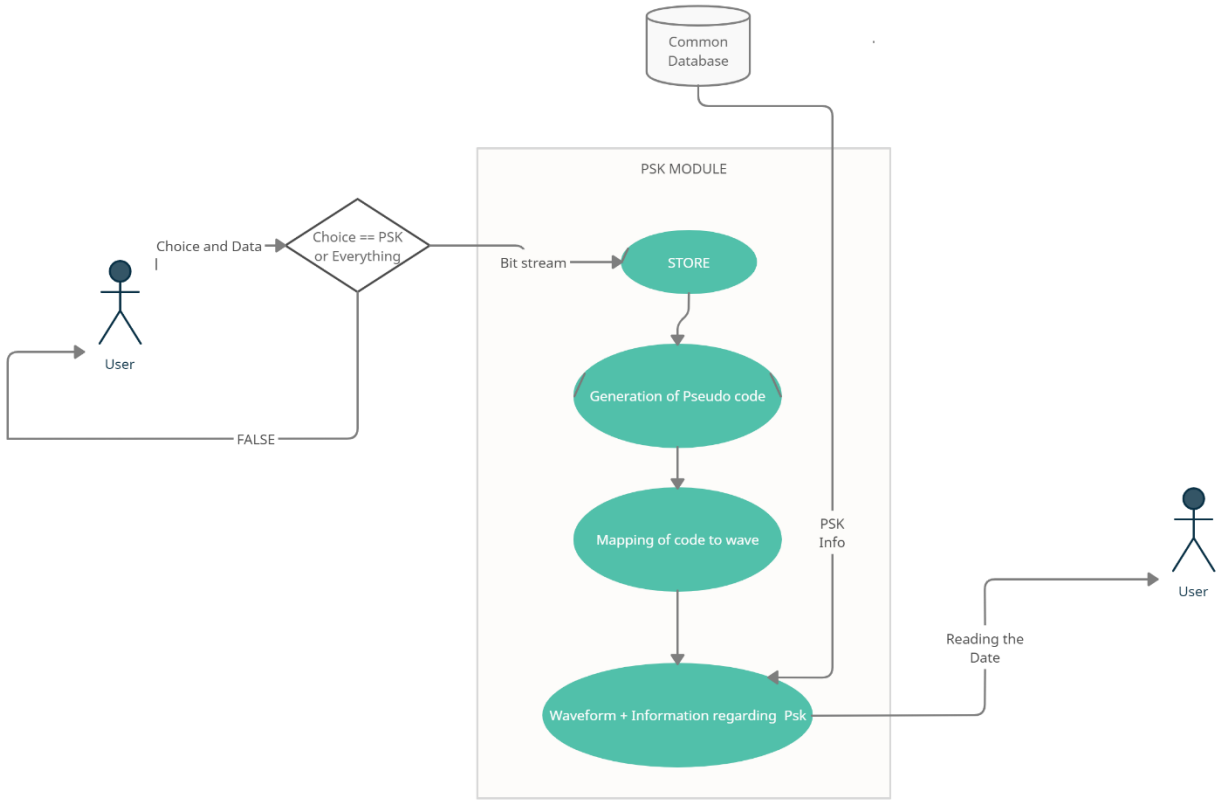
FSK Use Case Diagram



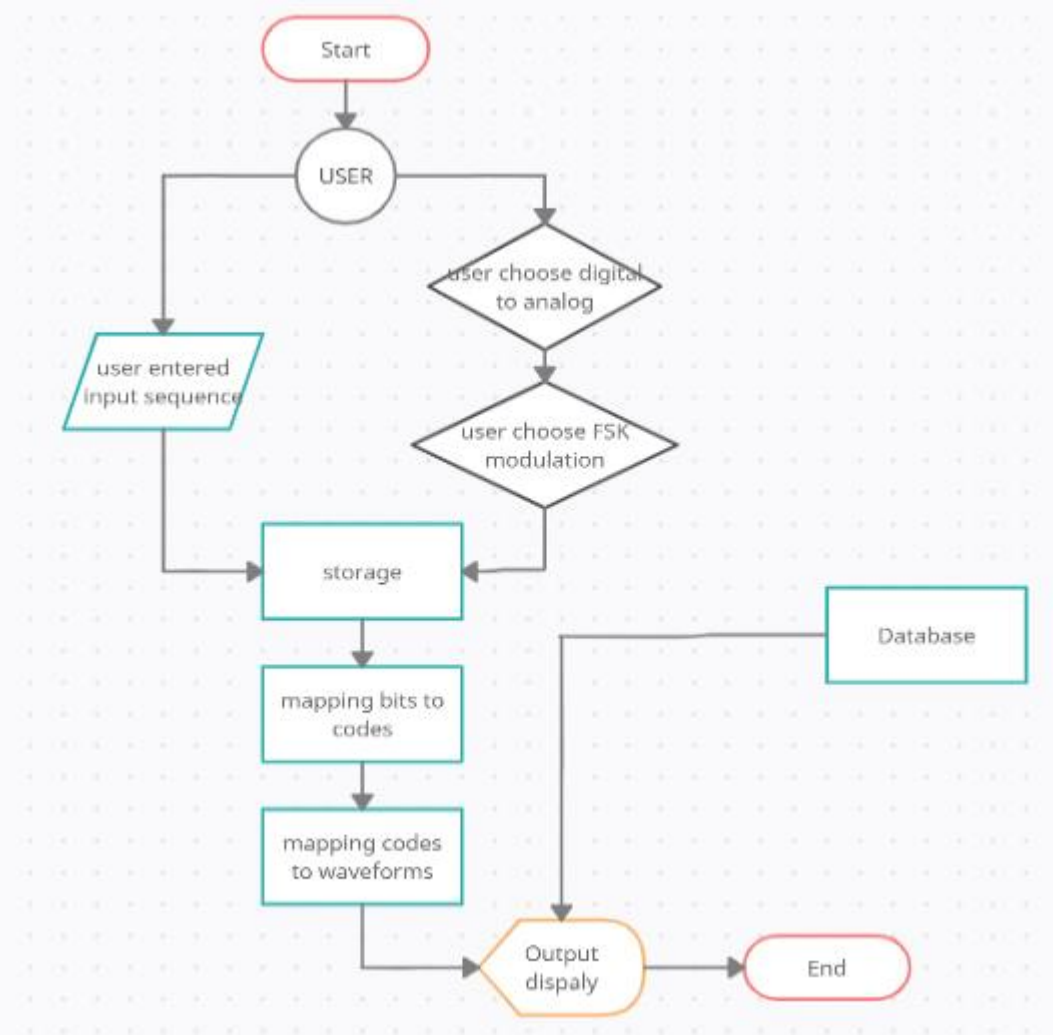
PSK Component Diagram



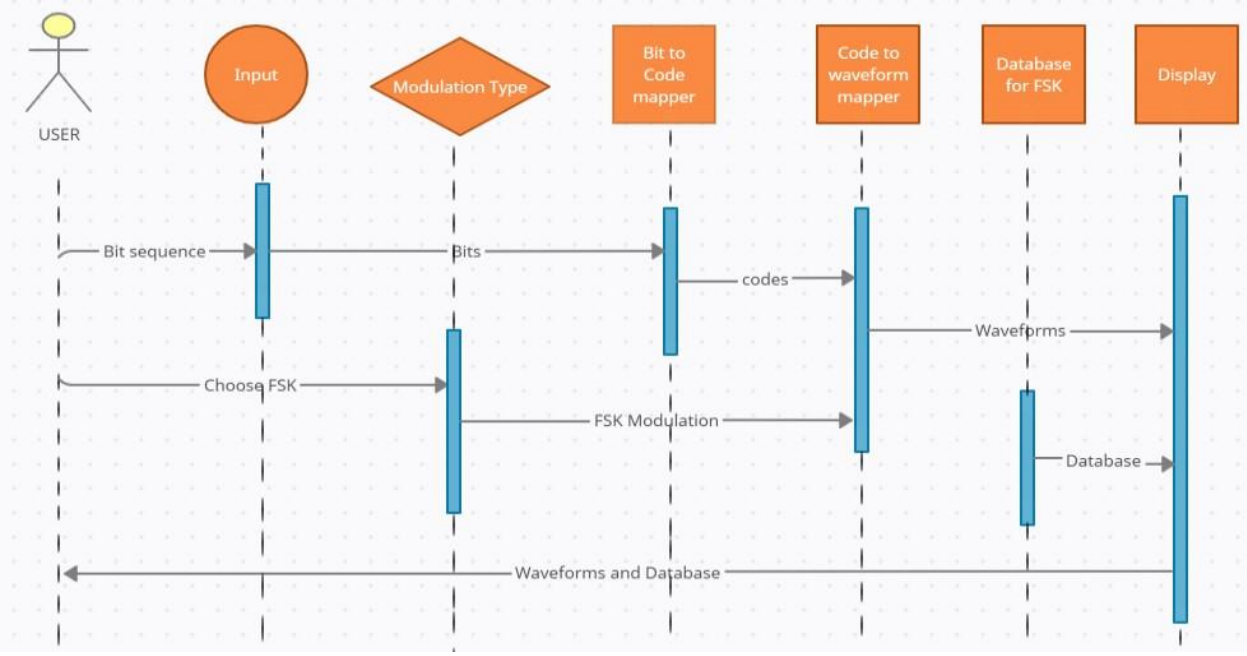
PSK Use Case Diagram



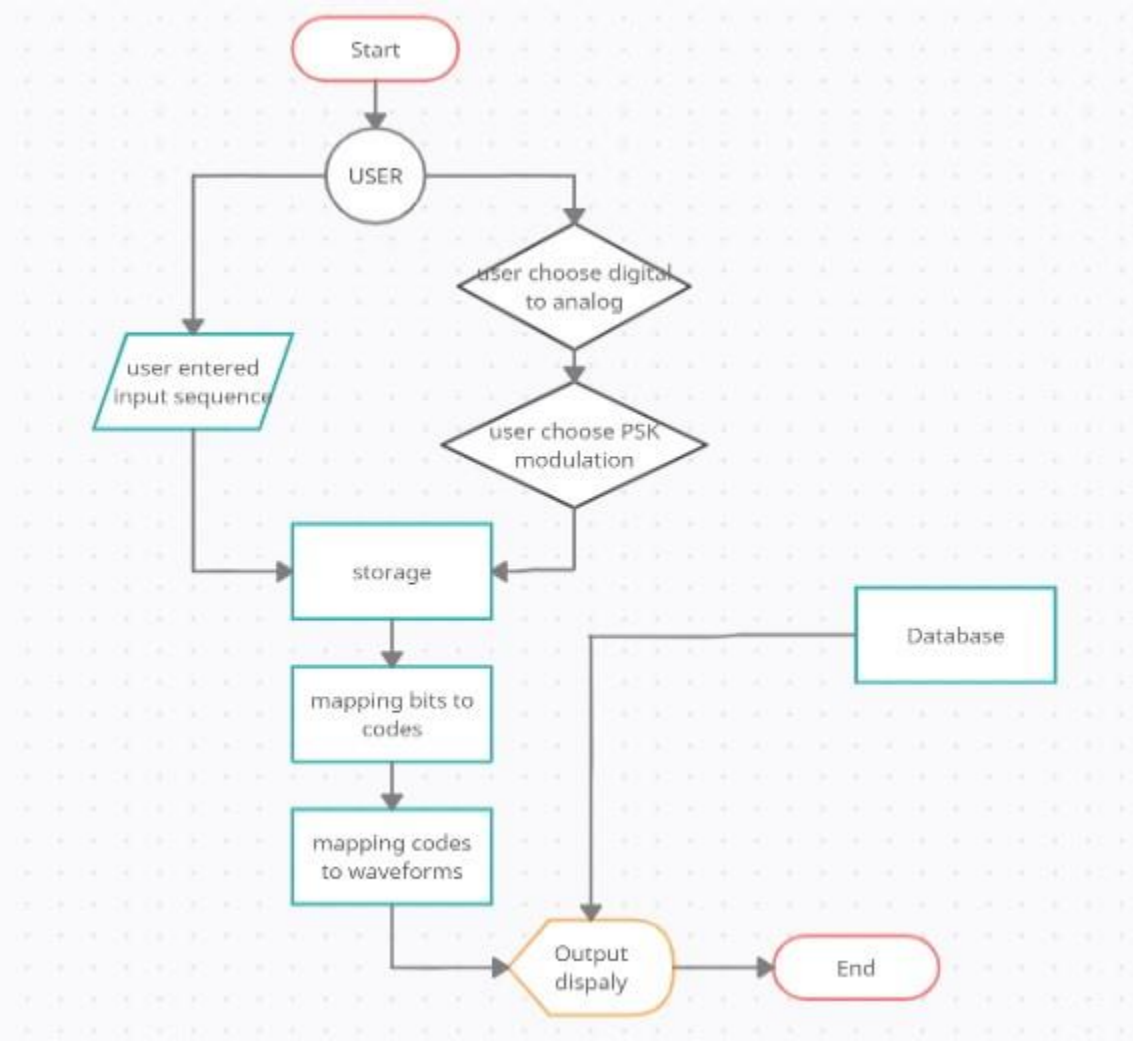
FSK Activity Diagram



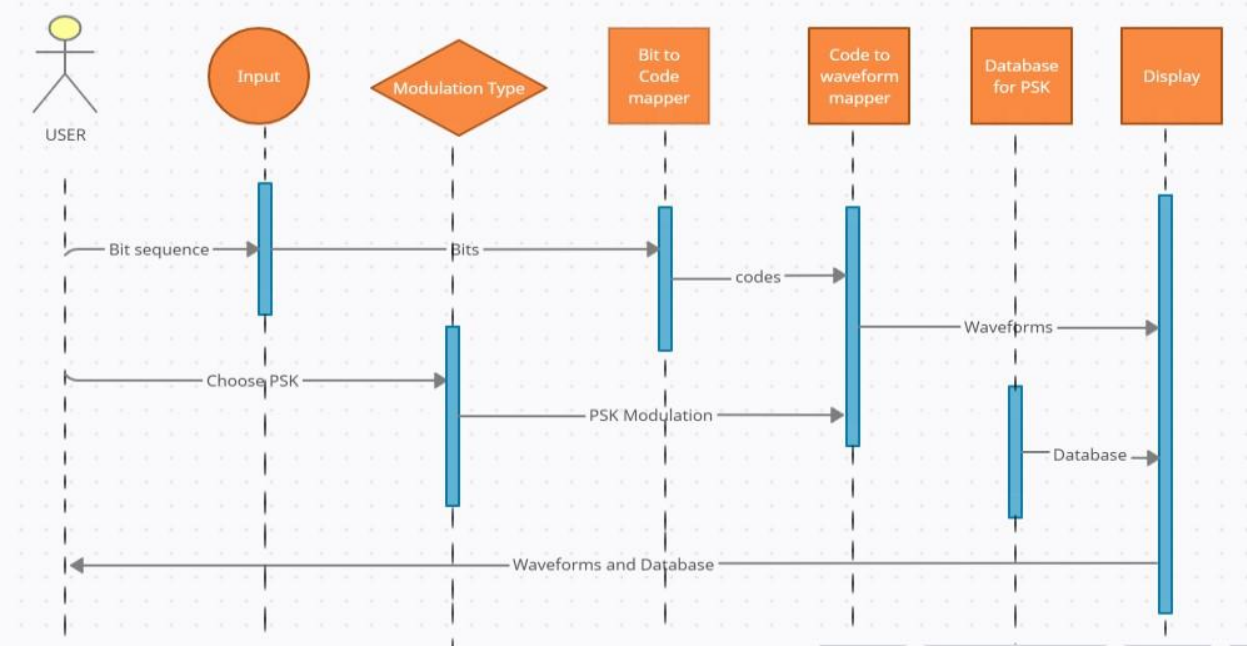
FSK Sequence Diagram



PSK Activity Diagram



PSK Sequence Diagram



TEST PLAN:

Table: High level test plan

Test ID	Description	Exp IN	Exp OUT	Actual Out	Means of Test	Type Of Test
H_01	Display should work	---	display	OPdisp	Tester	Requirement Based
H_02	Waveforms should turn up	---	waveform	Opwaveform	Tester	Scenario Based
H_03	Input Interface	---	interfacedisplay	OPintdisp	Tester	Requirement Based
H_04	Access Database	Integer(1-5)	database	Opdatabase	Tester	Scenario Based

Table: Low level test plan

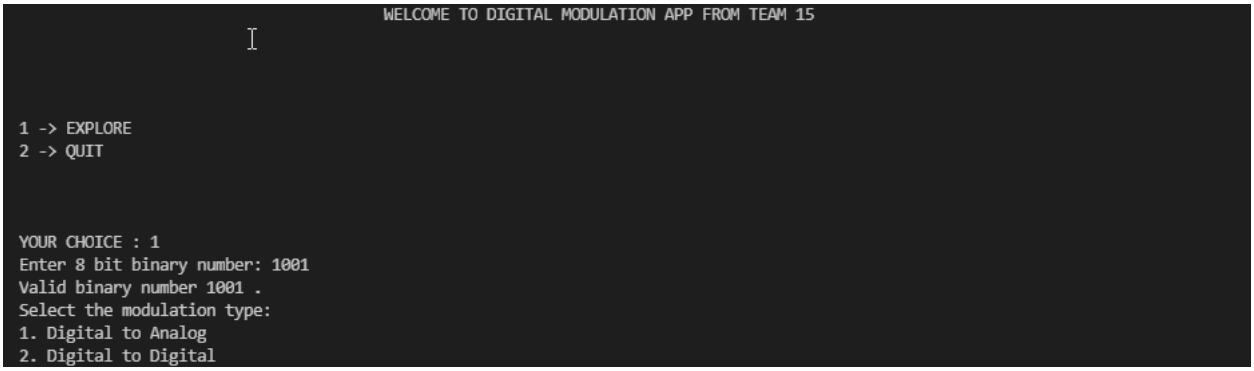
Test ID	Description	Exp I/P	Exp O/P	Actual Out	Means of Test
L_01	ASK	`Enter 8 bit binary number: 10001101`	ASK OP	ASK OP	Tester

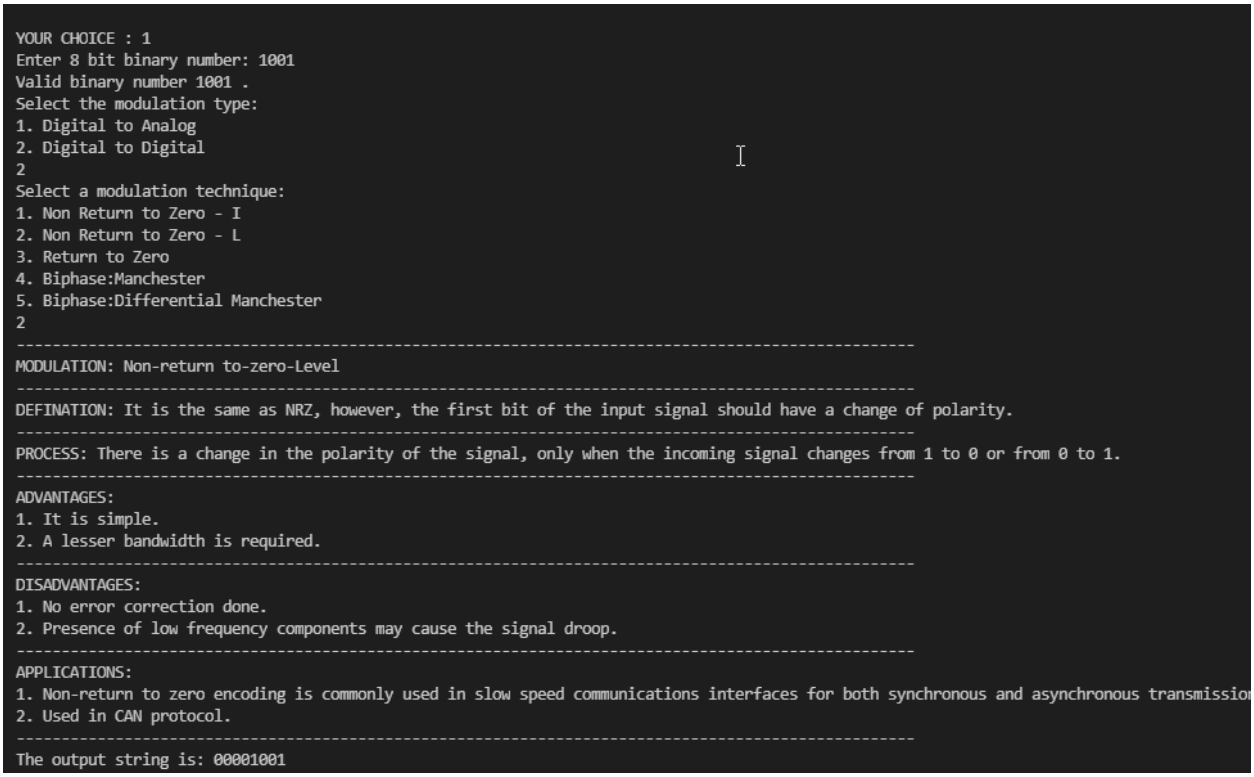
Test ID	Description	Exp I/P	Exp O/P	Actual Out	Means of Test
L_02	FSK	`Enter 8 bit binary number: 10101011`	FSK OP	FSK OP	Tester
L_03	PSK	`Enter 8 bit binary number: 11101001`	PSK OP	PSK OP	Tester
L_04	QAM	`Enter 8 bit binary number: 11101001`	QAM OP	QAMP OP	Tester
L_05	NRZ-L	`Enter 8 bit binary number: 11011110`	NRZL OP	NRZL OP	Tester
L_06	NRZ-L	`Enter 8 bit binary number: 10010011`	NRZL OP	NRZL OP	Tester
L_07	NRZ-L	`Enter 8 bit binary number: 11001`	NRZL OP	NRZL OP	Tester
L_08	NRZ-I	`Enter 8 bit binary number: 11011110`	NRZI OP	NRZI OP	Tester

Test ID	Description	Exp I/P	Exp O/P	Actual Out	Means of Test
L_09	NRZ-I	`Enter 8 bit binary number: 10010011`	NRZI OP	NRZI OP	Tester
L_10	NRZ-I	`Enter 8 bit binary number: 11001`	NRZI OP	NRZI OP	Tester
L_11	RZ	`Enter 8 bit binary number: 11011110`	RZ OP	RZ OP	Tester
L_12	RZ	`Enter 8 bit binary number: 10010011`	RZ OP	RZ OP	Tester
L_13	RZ	`Enter 8 bit binary number: 11001`	RZ OP	RZ OP	Tester
L_14	Biphase Manchester	`Enter 8 bit binary number: 11011110`	Man EOP	Man AOP	Tester
L_15	Differential Manchester	`Enter 8 bit binary number: 11011110`	Diff EOP	Diff AOP	Tester

Results

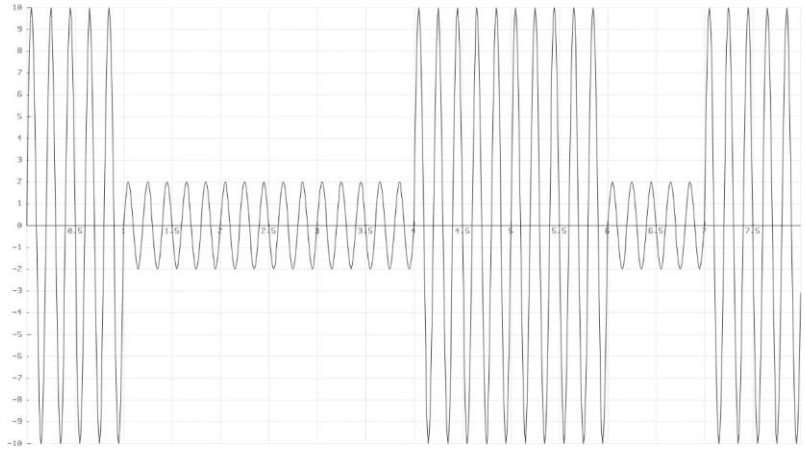
Implementation Screen

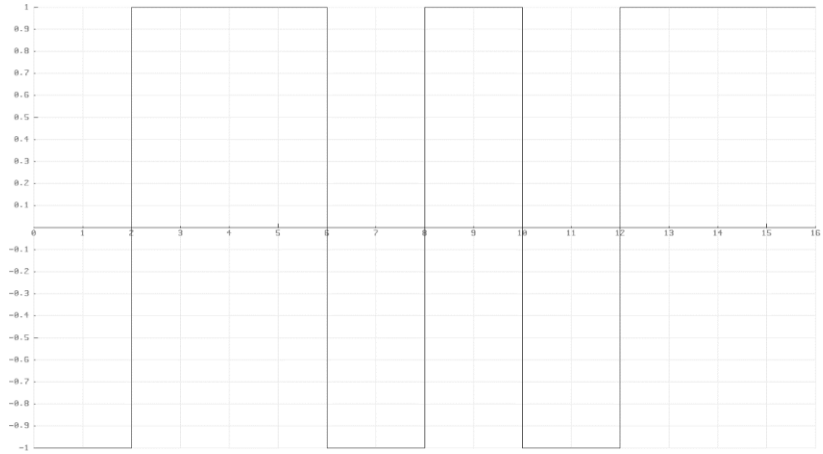


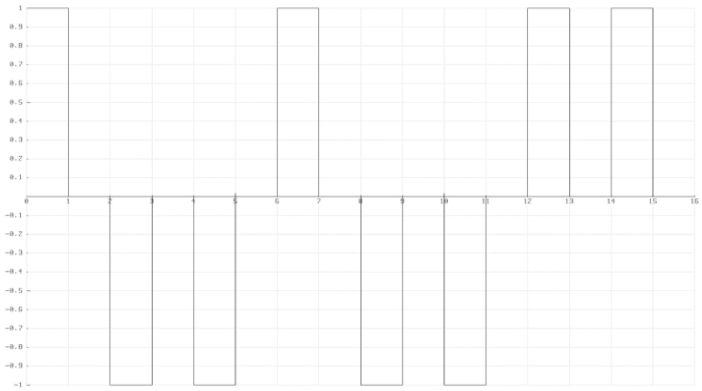


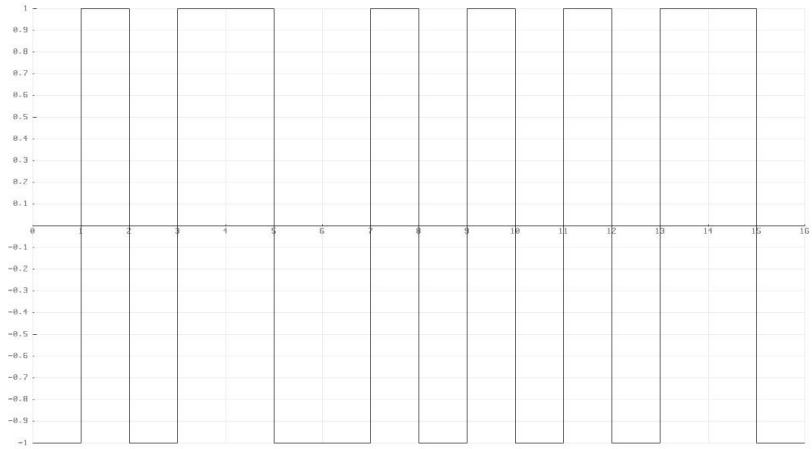
```
                                WELCOME TO DIGITAL MODULATION APP FROM TEAM 15




1 -> EXPLORE
2 -> QUIT
          I


YOUR CHOICE : 2
                                WELCOME TO DIGITAL MODULATION APP FROM TEAM 15




ARE YOU SURE YOU WANT TO QUIT? :
```

Display Waveforms Examples

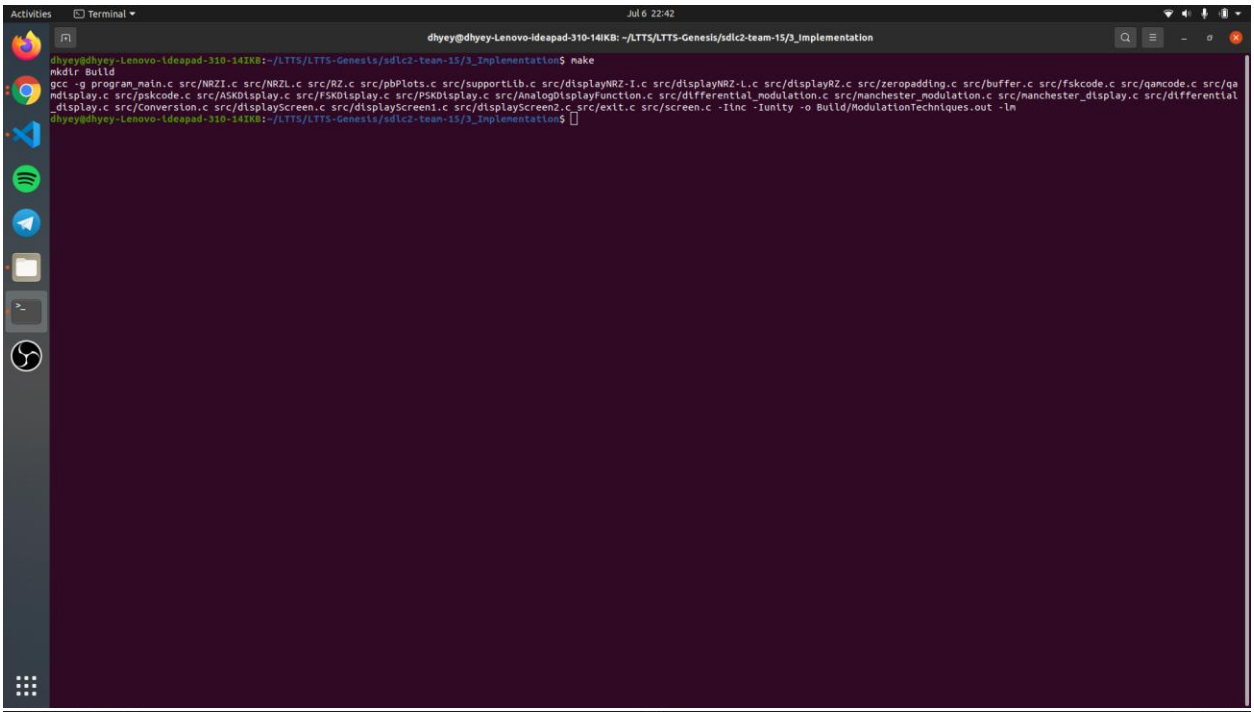


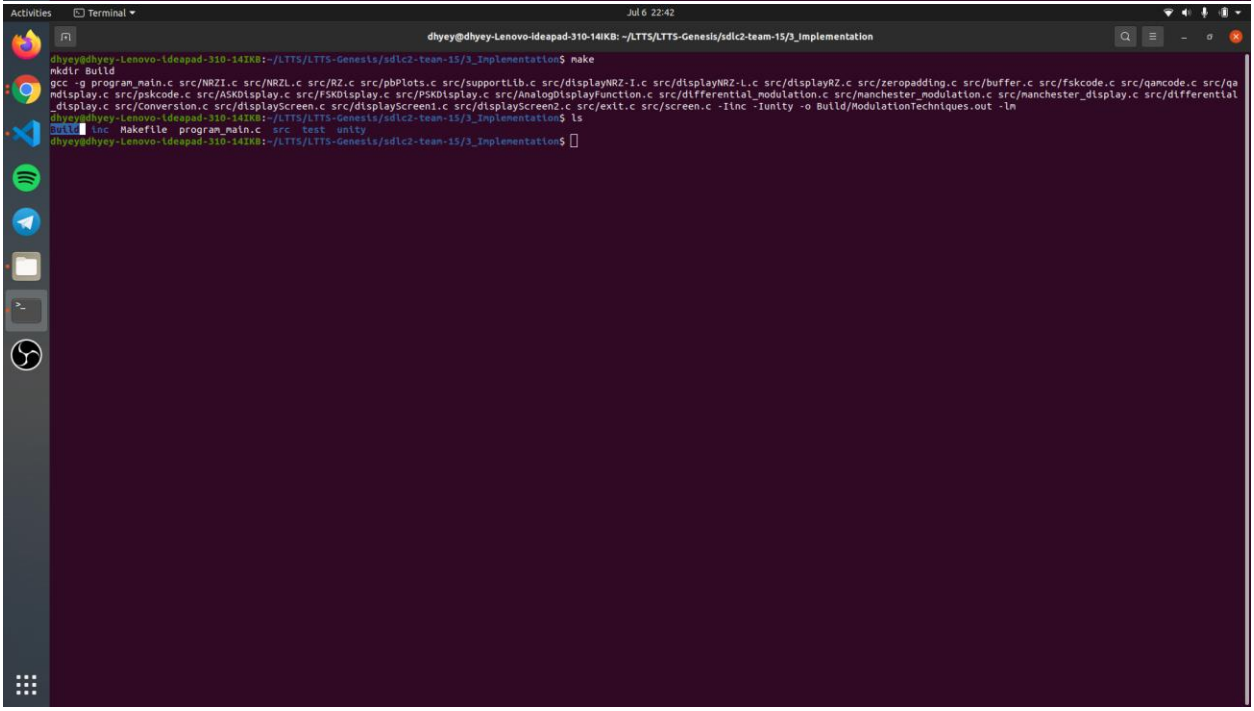




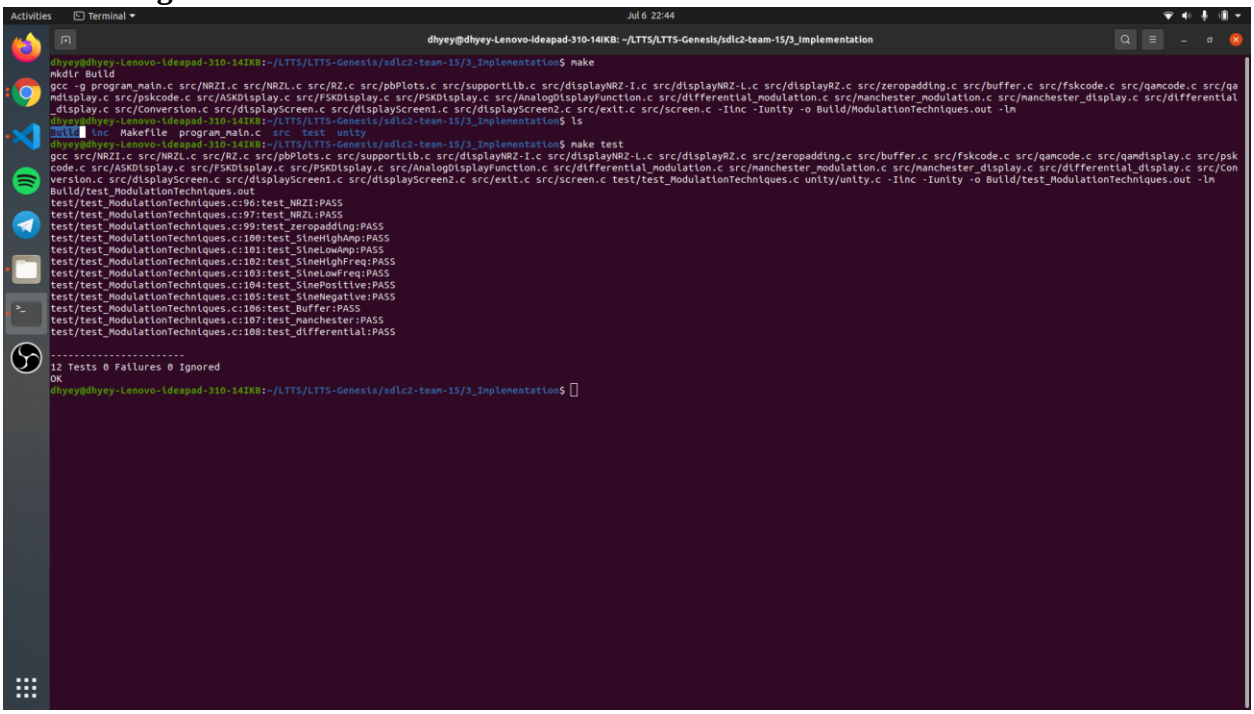


Build

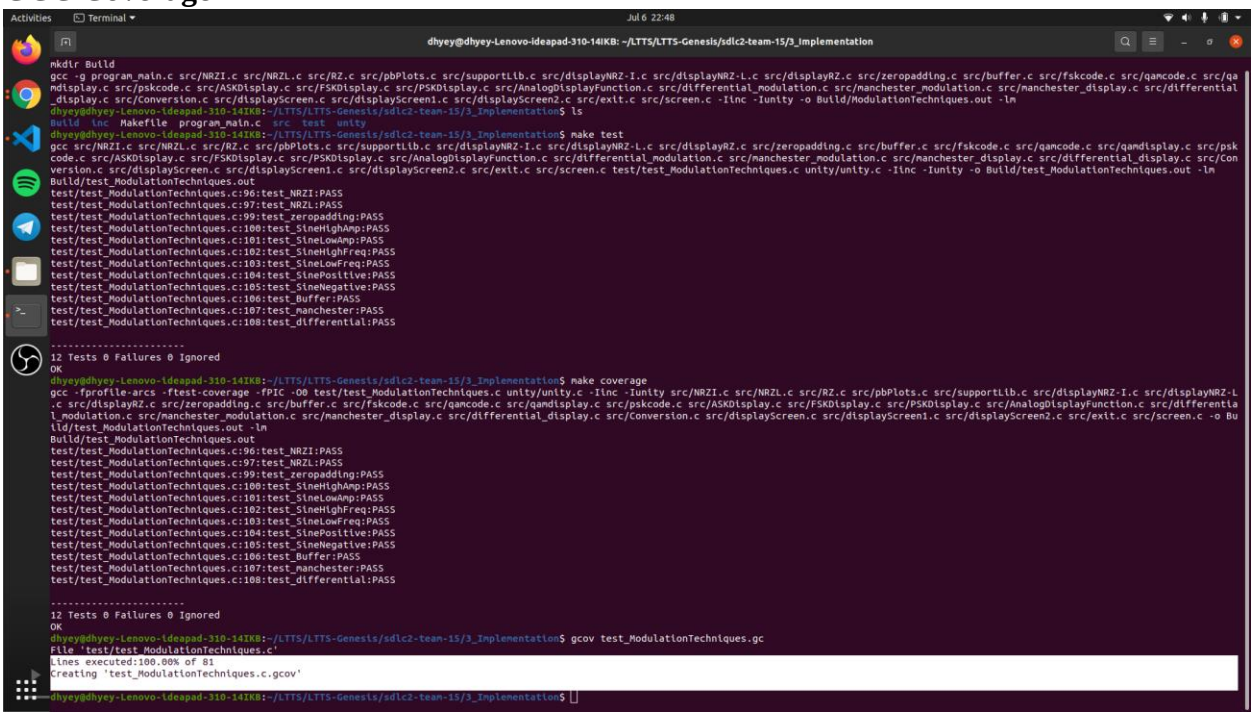

Activities
Terminal
Jul 6 22:42
dhyey@dhyey-Lenovo-ideapad-310-14IKB: ~/LTTS/LTTS-Genesis/sdlc2-team-15/3_Implementation
dhyey@dhyey-Lenovo-ideapad-310-14IKB:~/LTTS/LTTS-Genesis/sdlc2-team-15/3_Implementation$ make
mkdir Build
gcc -g program_main.c src/NRZI.c src/NRZL.c src/RZ.c src/pbPlots.c src/supportLib.c src/displayNRZ-I.c src/displayNRZ-L.c src/displayRZ.c src/zeropadding.c src/buffer.c src/fskcode.c src/qamcode.c src/qamdisplay.c src/pskcode.c src/ASKDisplay.c src/FSKDisplay.c src/PSKDisplay.c src/AnalogDisplayFunction.c src/differential_modulation.c src/manchester_modulation.c src/manchester_display.c src/differential_display.c src/Conversion.c src/displayScreen.c src/displayScreen1.c src/displayScreen2.c src/exit.c src/screen.c -Iinc -Iunity -o Build/ModulationTechniques.out -lm
dhyey@dhyey-Lenovo-ideapad-310-14IKB:~/LTTS/LTTS-Genesis/sdlc2-team-15/3_Implementation$


Activities
Terminal
Jul 6 22:42
dhyey@dhyey-Lenovo-ideapad-310-14IKB: ~/LTTS/LTTS-Genesis/sdlc2-team-15/3_Implementation
dhyey@dhyey-Lenovo-ideapad-310-14IKB:~/LTTS/LTTS-Genesis/sdlc2-team-15/3_Implementation$ make
mkdir Build
gcc -g program_main.c src/NRZI.c src/NRZL.c src/RZ.c src/pbPlots.c src/supportLib.c src/displayNRZ-I.c src/displayNRZ-L.c src/displayRZ.c src/zeropadding.c src/buffer.c src/fskcode.c src/qamcode.c src/qamdisplay.c src/pskcode.c src/ASKDisplay.c src/FSKDisplay.c src/PSKDisplay.c src/AnalogDisplayFunction.c src/differential_modulation.c src/manchester_modulation.c src/manchester_display.c src/differential_display.c src/Conversion.c src/displayScreen.c src/displayScreen1.c src/displayScreen2.c src/exit.c src/screen.c -Iinc -Iunity -o Build/ModulationTechniques.out -lm
dhyey@dhyey-Lenovo-ideapad-310-14IKB:~/LTTS/LTTS-Genesis/sdlc2-team-15/3_Implementation$ ls
Build Inc Makefile program_main.c src test unity
dhyey@dhyey-Lenovo-ideapad-310-14IKB:~/LTTS/LTTS-Genesis/sdlc2-team-15/3_Implementation$

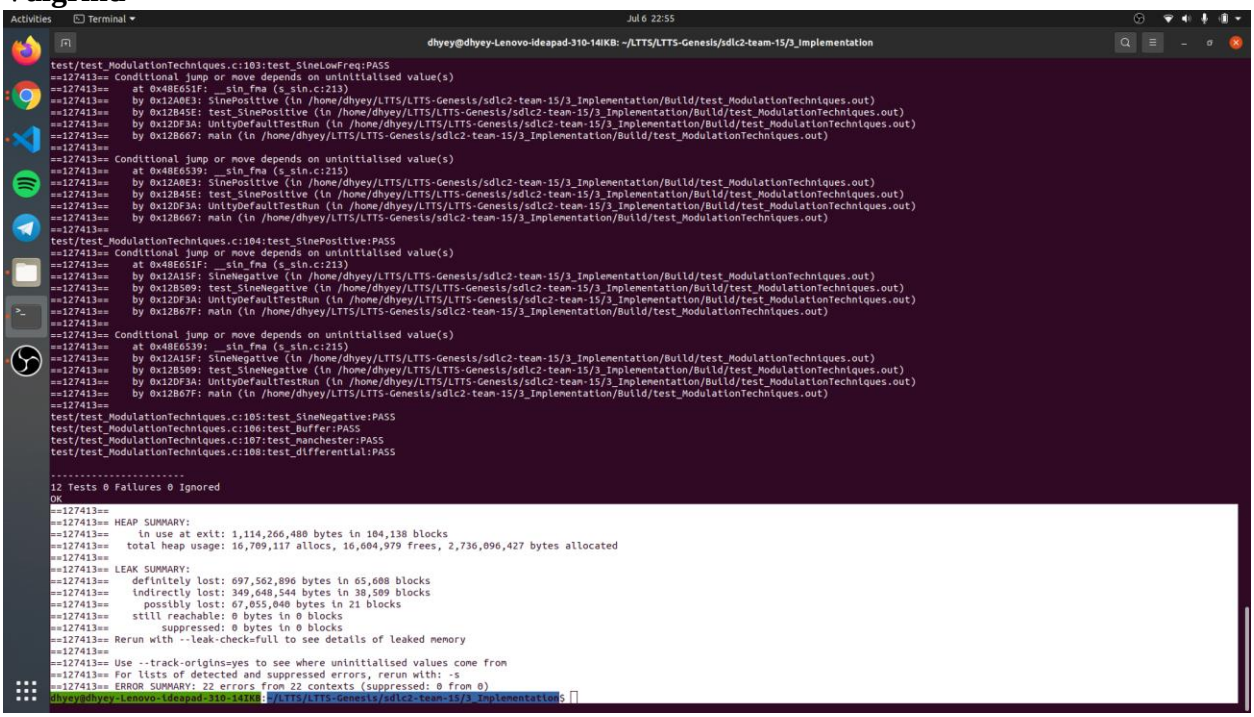
Unit Testing



GCC Coverage



Valgrind



References

- Different Modulation Techniques
- How to plot in C