# CT303: Digital Communications

## Prof. Manish Kumar

## Lab - 6


## Name : Jeet Daiya

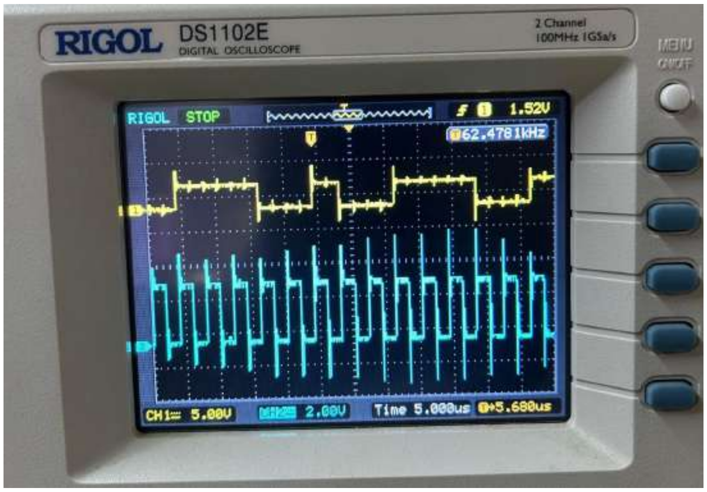
## ID: 202301017 Lab Group: 1

**Group:**

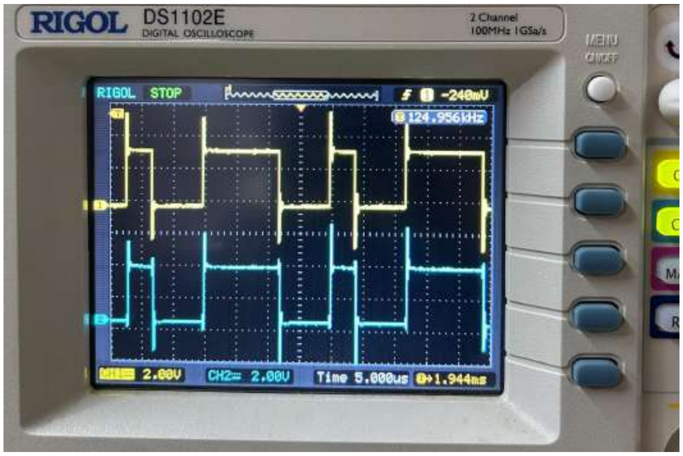
Dhruvil Mehta - 202301061
Jeet Daiya - 202301017
Yug Patel - 202301051
Jay Rathod - 202301006
Vansh Padaliya - 202301065

- ## **Experiment numbers 1 :**

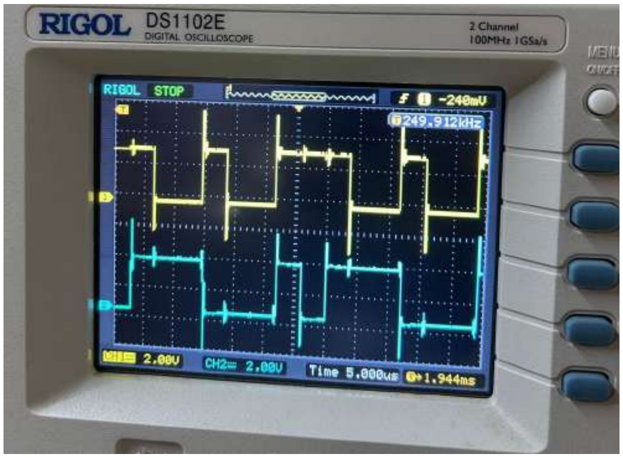
CH 1: DATA CLK (266 KHz) & CH 2: SERIAL DATA (00011011)



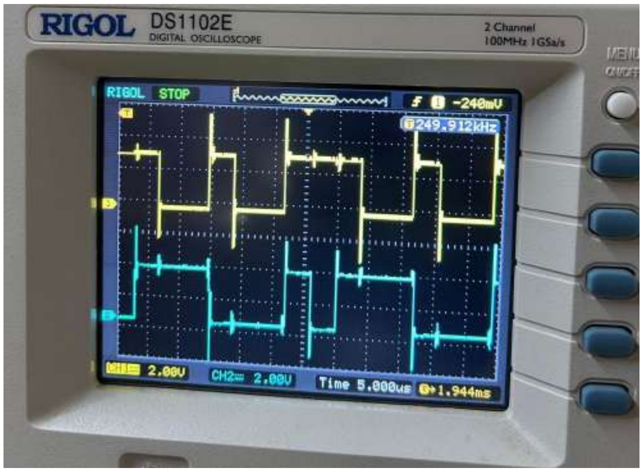

CH 1:DATA IN& CH 2:NRZ-L



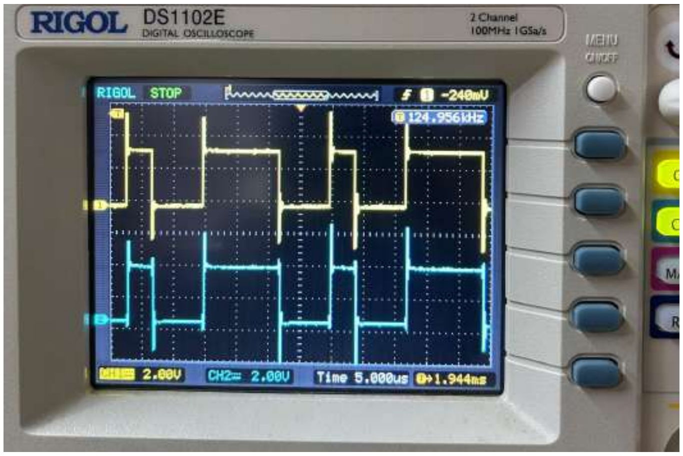
CH 1: DATA IN& CH 2: NRZ-M



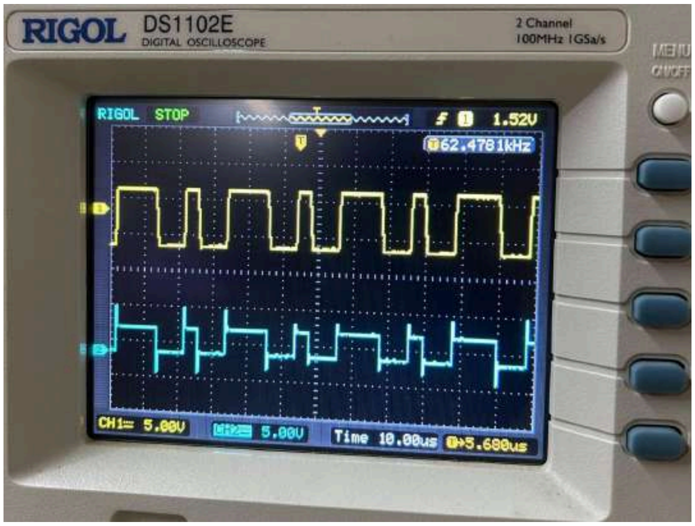

CH 1: DATA IN& CH 2: NRZ-M



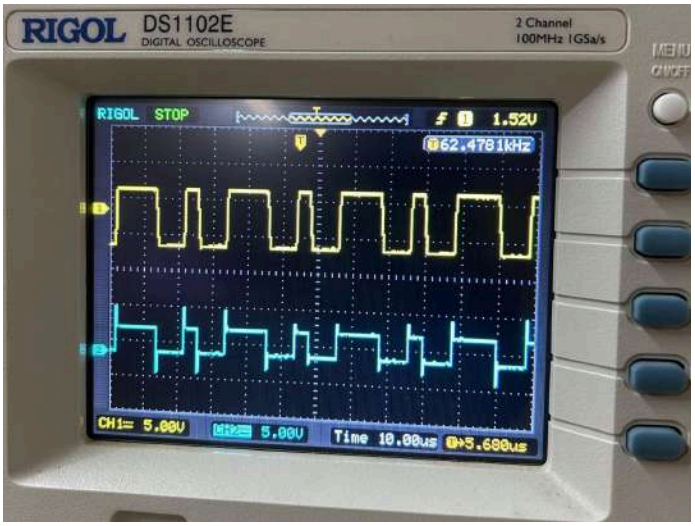
CH 1: DATA IN& CH 2: NRZ-S



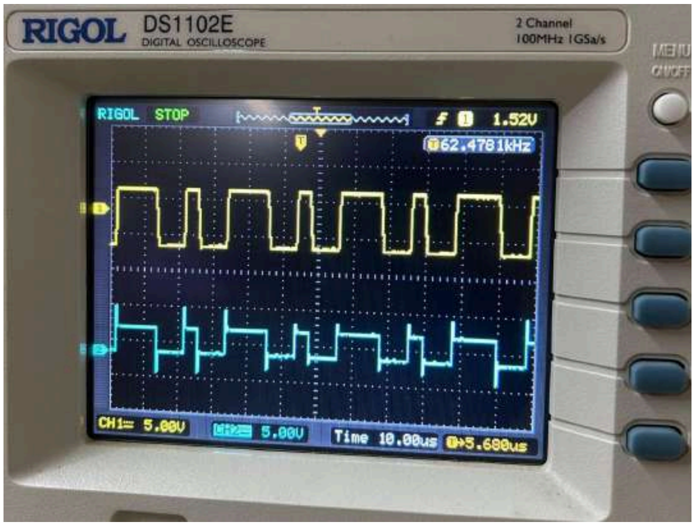

CH 1:DATA IN& CH 2: OUT1



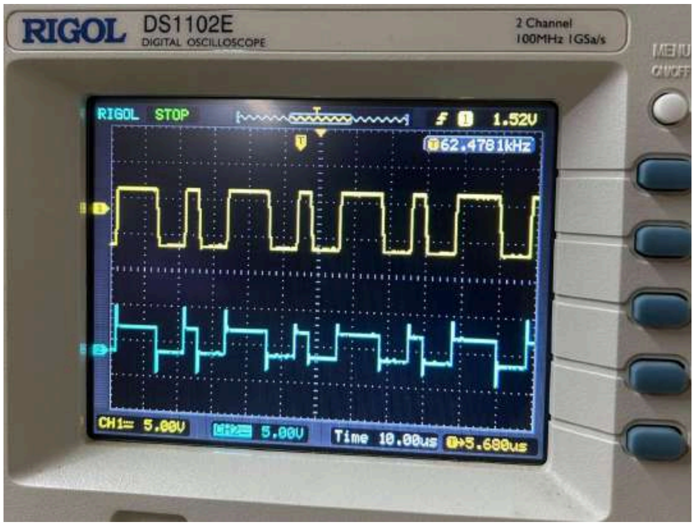
CH 1: DATA IN& CH 2: OUT2



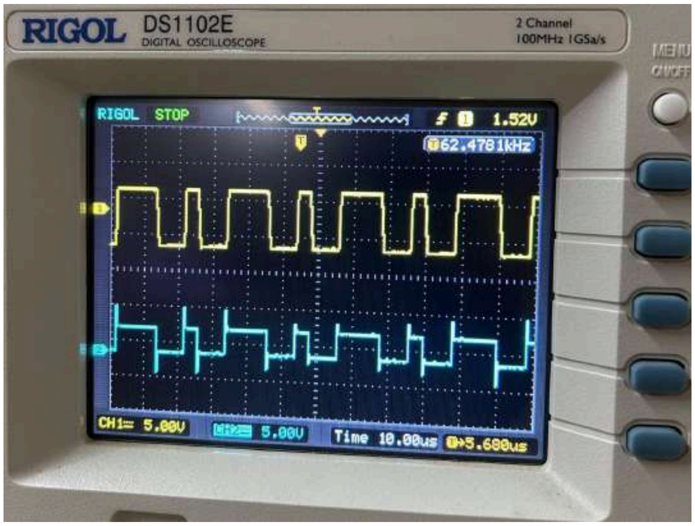

CH 1: DATA IN& CH 2: OUT3



CH 1: IN10& CH 2:OUT10

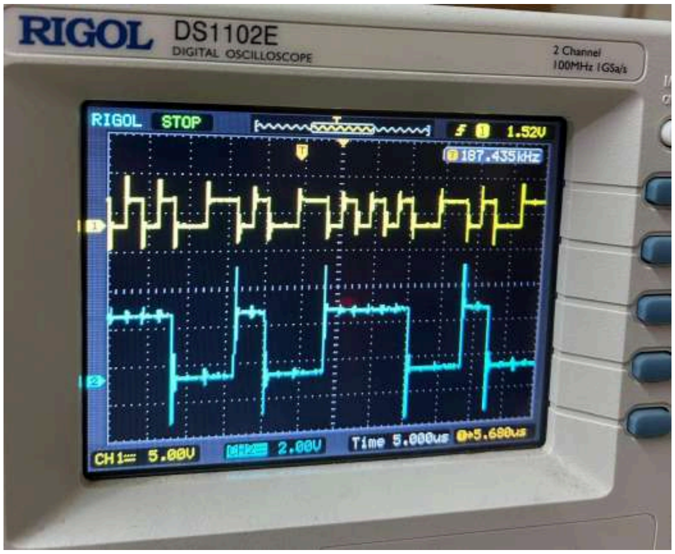



CH 1: IN10& CH 2: OUT11

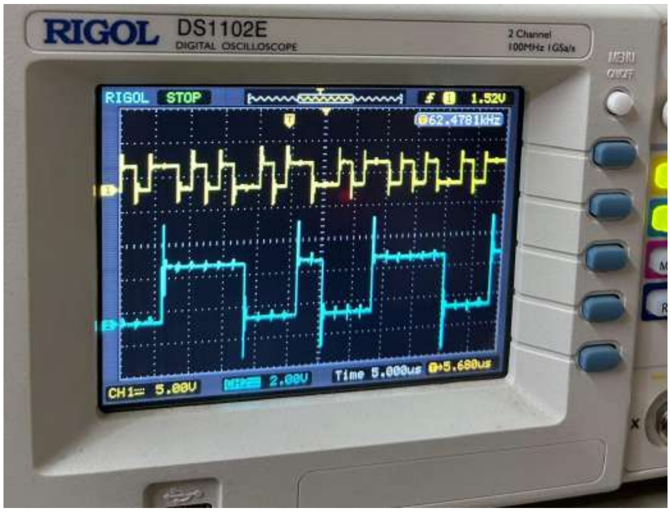


- ## Experiment numbers 2 :

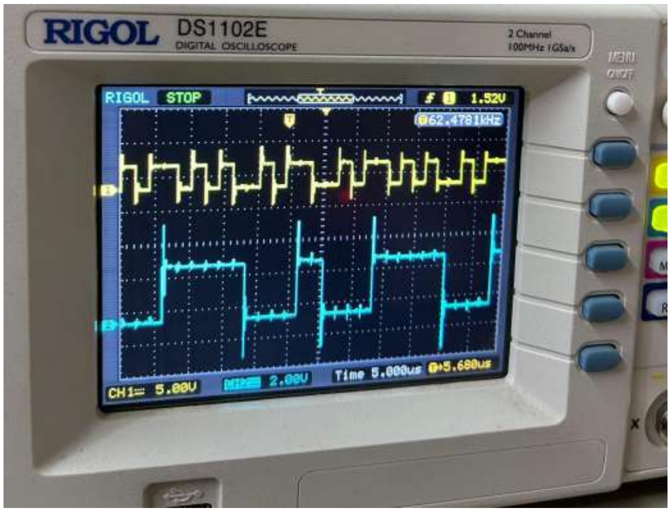
CH 1: DATA CLK (266 KHz) & CH 2: SERIAL DATA (00011011)



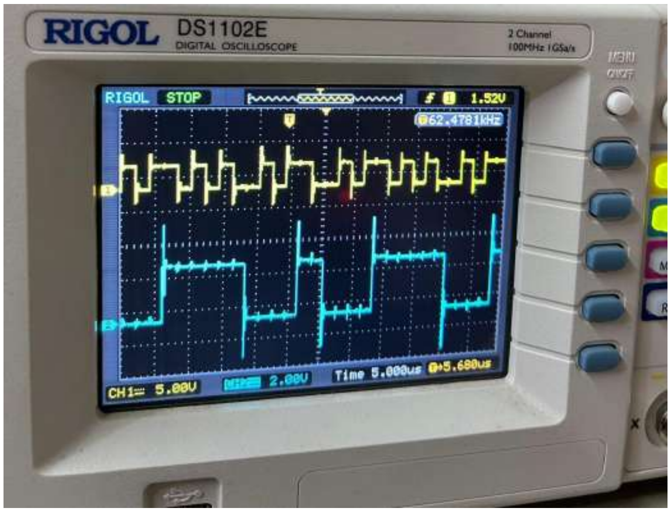

CH 1: DATA IN& CH 2: BIO-L



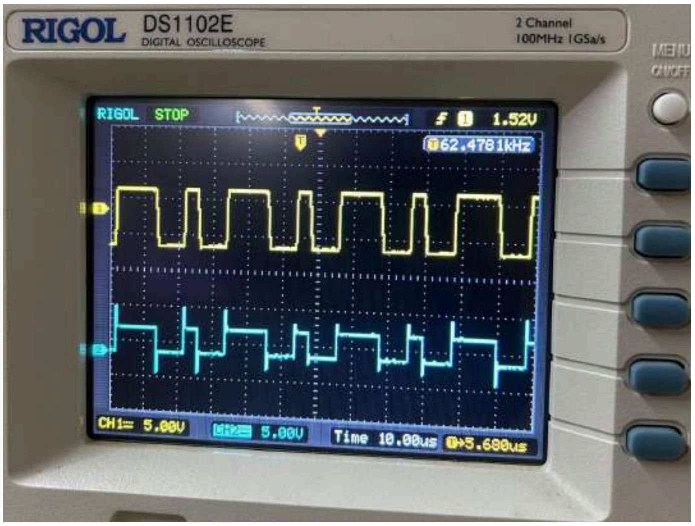
CH 1: DATA IN & CH 2: BIO-M



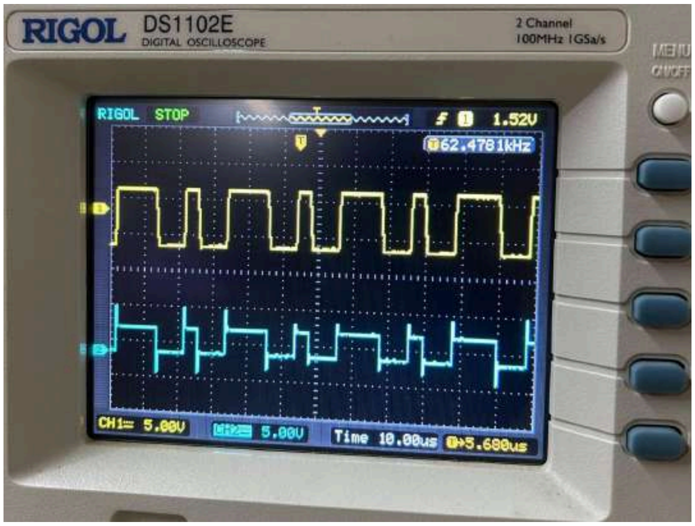

CH 1: DATA IN& CH 2: BIO-S



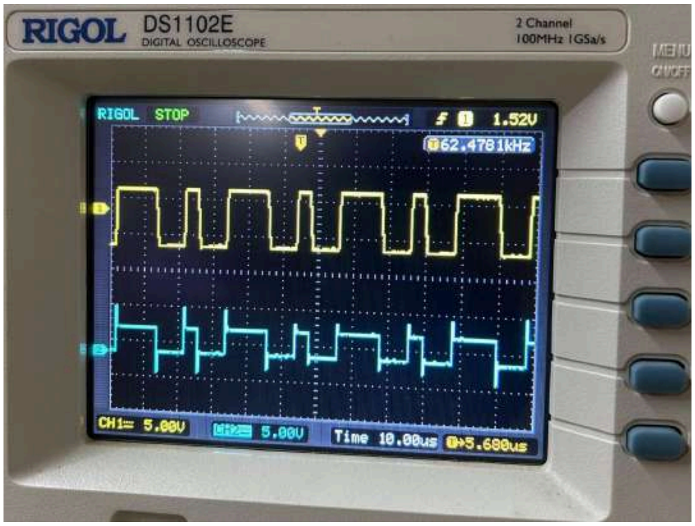
CH 1: DATA IN& CH 2: OUT5




CH 1: DATA IN& CH 2: OUT6

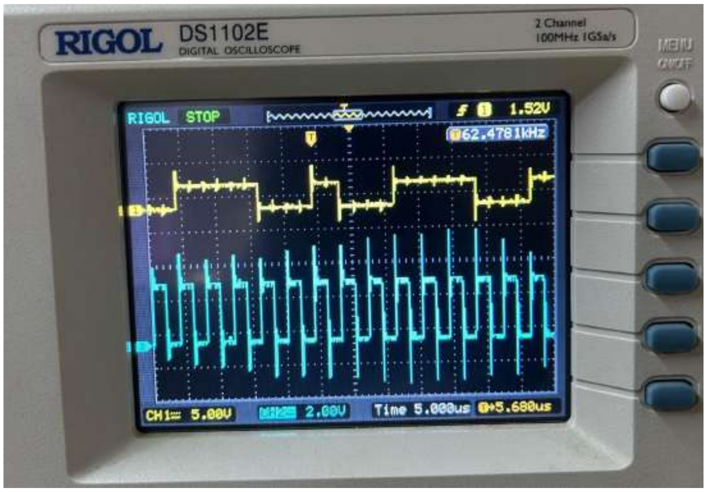


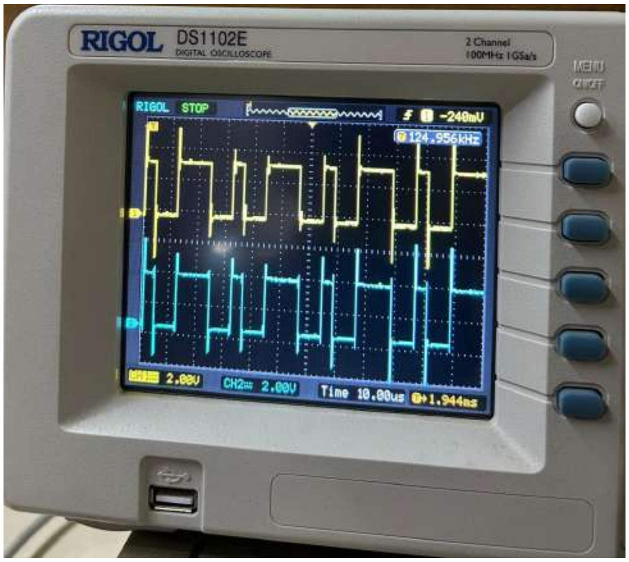
CH 1: DATA IN& CH 2: OUT7



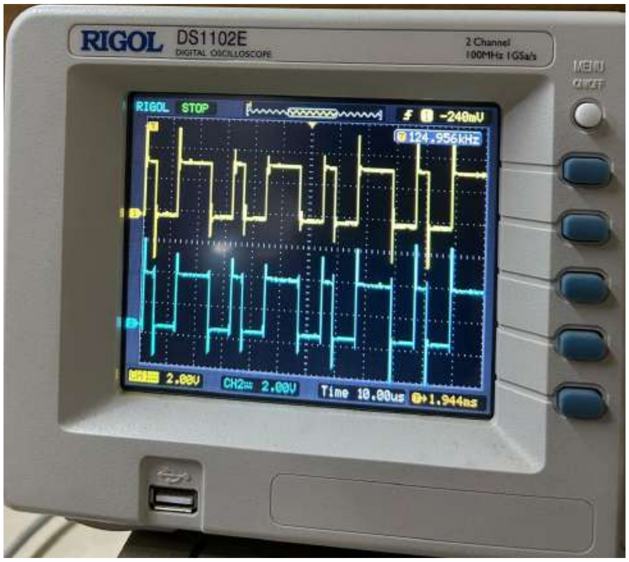

- ## Experiment numbers 3 :

CH 1: DATA CLK (266 KHz) & CH 2: SERIAL DATA (00011011)

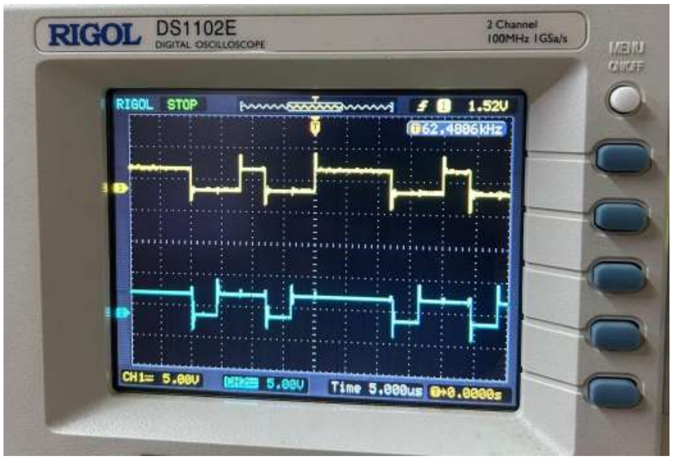




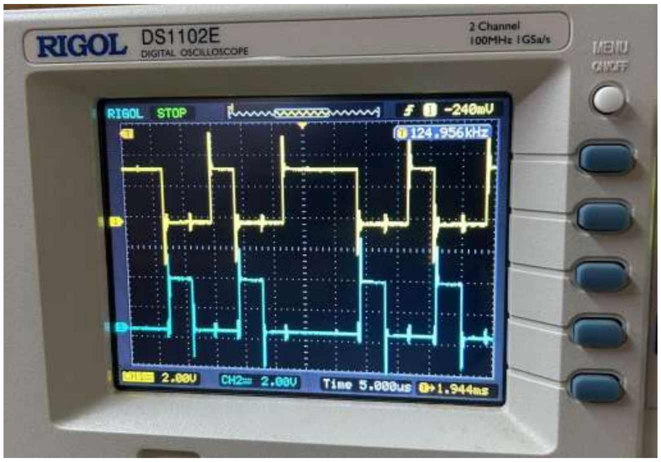

CH 1: DATA IN& CH 2: URZ



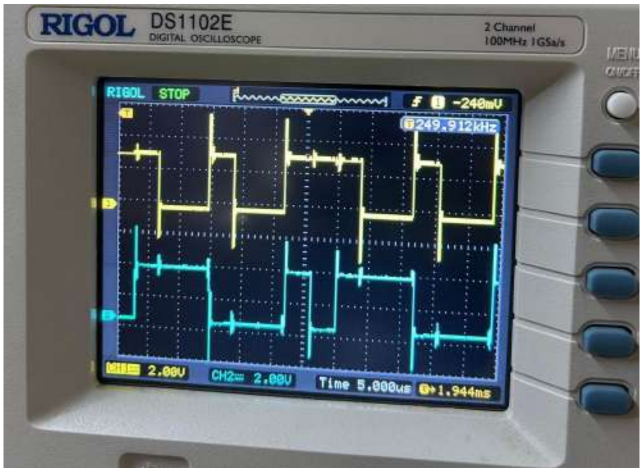
CH 1: DATA IN& CH 2: OUT4




CH 1: IN8 & CH 2: OUT8




CH 1: IN8 & CH 2: OUT9




- Problem 6.8 :

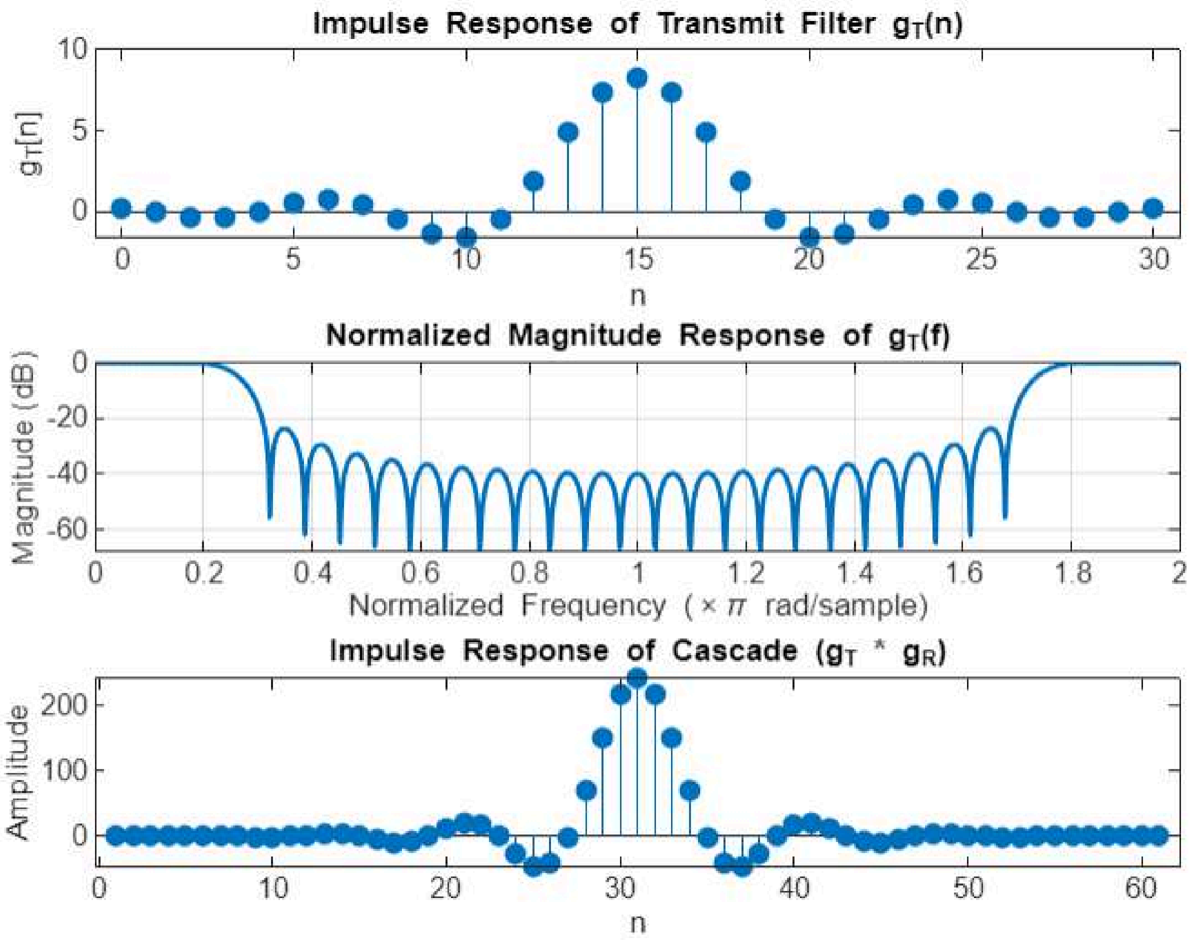
Code :

```
clc; clear; close all;
%% Parameters
N     = 31;         % Filter length (odd for symmetry)
T     = 1;          % Symbol period
alpha = 0.25;       % Roll-off factor
n     = -(N-1)/2:(N-1)/2;  % Time index
%% Compute transmitter filter impulse response g_T
g_T = zeros(1, length(n));  % initialize
for i = 1:length(n)
   for m = -(N-1)/2:(N-1)/2
       % Xrc() is the raised cosine spectrum function
       g_T(i) = g_T(i) + sqrt(xrc(4*m/(N*T), alpha, T)) * ...
                exp(1j*2*pi*m*n(i)/N);
   end
end
```

```
%% Take real part (imag part is numerical error)
g_T = real(g_T);
%% Delay to make linear phase
n2 = 0:N-1;
g_T_shifted = g_T;  % already symmetric, shift not mandatory for
FIR plots
%% Receiver filter (matched filter to g_T)
g_R = g_T;
%% Cascade impulse response
imp_resp_cascade = conv(g_T, g_R);
%% Frequency response of g_T
[GT, W] = freqz(g_T, 1, 1024, 'whole');
magGT_dB = 20*log10(abs(GT)/max(abs(GT)));
%% Plot results
figure;
subplot(3,1,1);
stem(n2, g_T_shifted, 'filled');
title('Impulse Response of Transmit Filter g_T(n)');
xlabel('n'); ylabel('g_T[n]');
subplot(3,1,2);
plot(W/pi, magGT_dB, 'LineWidth', 1.5);
title('Normalized Magnitude Response of g_T(f)');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Magnitude (dB)'); grid on;
subplot(3,1,3);
stem(imp_resp_cascade, 'filled');
```

```
title('Impulse Response of Cascade (g_T * g_R)');

xlabel('n'); ylabel('Amplitude');

%% Function: Raised Cosine Spectrum

function y = xrc(f, alpha, T)

   % Raised cosine frequency response (Xrc(f))

   if abs(f) > (1+alpha)/(2*T)

       y = 0;

   elseif abs(f) > (1-alpha)/(2*T)

       y = (T/2) * (1 + cos((pi*T/alpha) * ...

           (abs(f) - (1-alpha)/(2*T))));

   else

       y = T;

   end

end
```

- Output :
    - Photos :




## Conclusion :

1. **Parameters:**
    - `N = 31` → Filter length (odd so filter is symmetric).
    - `alpha = 0.25` → Roll-off factor of the raised cosine.
    - `T = 1` → Symbol period.

2. **Transmit Filter `g_T(n)`:**
   - Constructed using the **raised cosine frequency response (Xrc)**.
   - Take the real part to remove tiny imaginary numerical errors.
   - Since the filter is symmetric, the linear phase is already satisfied.
3. **Receiver Filter `g_R`:**
   - Same as the transmitter filter (matched filter).
4. **Cascade Response:**
   - Convolution of `g_T` and `g_R` gives the **overall system response** (transmitter + receiver).
5. **Frequency Response:**
   - Computed with `freqz`.
   - Normalized magnitude is shown in **dB**.

---

## 🔹 What the Output Shows

- **Figure (3 Subplots):**
  - **Impulse Response of g_T(n):**
    Looks symmetric → ensures **linear phase** and no distortion in time domain.
  - **Magnitude Response of g_T(f):**
    - Flat in the passband (ideal transmission).

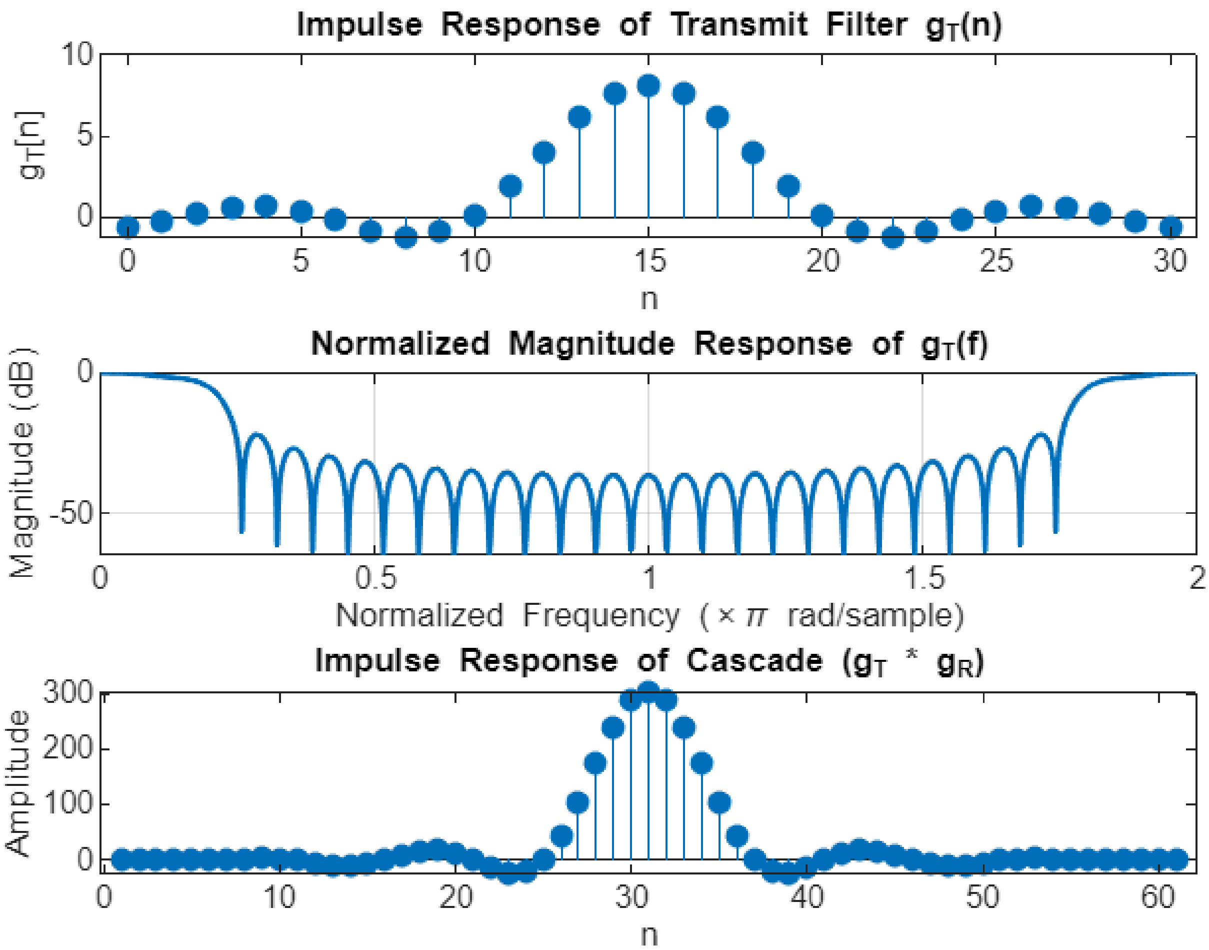
        - Smooth transition band controlled by roll-off factor `α = 0.25`.
        - Attenuates outside passband ➜ avoids ISI (Inter-Symbol Interference).
    - **Cascade Impulse Response (g_T * g_R):**
        - Peaks sharply at center, small sidelobes ➜ behave like a **Nyquist filter**.
        - Ensures **ISI-free transmission** when sampled at symbol intervals.

- Problem 6.9 :
    - Code :

```
clc; clear; close all;
%% Parameters
N = 31;          % Filter length (odd for symmetry)
T = 1;           % Symbol period
W = 1/(2*T);     % Bandwidth
n = -(N-1)/2:(N-1)/2;   % Time index
%% Compute transmitter filter impulse response g_T
g_T = zeros(1, length(n));  % initialize
for i = 1:length(n)
   for m = -(N-1)/2:(N-1)/2
       f_val = (4*m)/(N*T);  % frequency grid
       if abs(f_val) <= W
           g_T(i) = g_T(i) + sqrt((1/W) * cos((2*pi*m)/(N*T*W))) *
...
                     exp(1j*2*pi*m*n(i)/N);
       end
```

```
   end
end
%% Take real part (remove numerical error)
g_T = real(g_T);
%% Receiver filter (matched filter to g_T)
g_R = g_T;
%% Cascade impulse response
imp_resp_cascade = conv(g_T, g_R);
%% Frequency response of g_T
[GT, Wfreq] = freqz(g_T, 1, 1024, 'whole');
magGT_dB = 20*log10(abs(GT)/max(abs(GT)));
%% Plot results
figure;
subplot(3,1,1);
stem(0:N-1, g_T, 'filled');
title('Impulse Response of Transmit Filter g_T(n)');
xlabel('n'); ylabel('g_T[n]');
subplot(3,1,2);
plot(Wfreq/pi, magGT_dB, 'LineWidth', 1.5);
title('Normalized Magnitude Response of g_T(f)');
xlabel('Normalized Frequency (\times\pi rad/sample)');
ylabel('Magnitude (dB)'); grid on;
subplot(3,1,3);
stem(imp_resp_cascade, 'filled');
title('Impulse Response of Cascade (g_T * g_R)');
xlabel('n'); ylabel('Amplitude');
```

- Output :
    - Photos :




## 🔹 Conclusion :

1. **Parameters:**
    - `N = 31` → Filter length, odd for symmetry.
    - `T = 1` → Symbol period.

- `W = 1/(2T)` → Bandwidth of the filter.
- `n` → Discrete-time index vector.

2. **Transmit Filter `g_T(n)`:**
   - Constructed by summing over frequency samples (`m`).
   - Spectrum is shaped using a **cosine weighting function** inside the passband.
   - Impulse response is real and symmetric → ensures **linear phase**.
3. **Receiver Filter `g_R`:**
   - Same as the transmit filter (matched filter).
4. **Cascade Response:**
   - Convolution of `g_T` with `g_R` gives the **overall system impulse response**.
5. **Frequency Response:**
   - Computed using `freqz`.
   - Normalized magnitude plotted in **dB**.

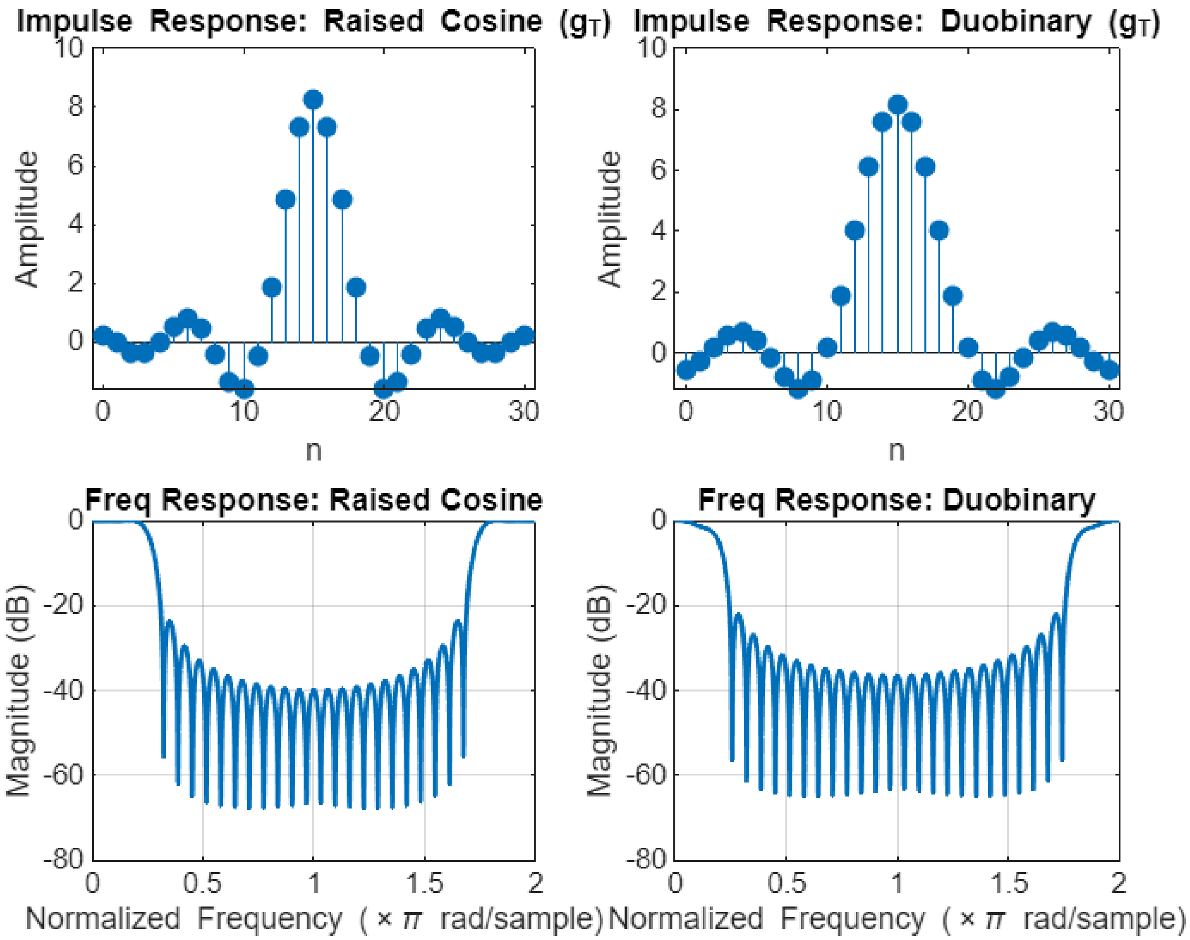
---

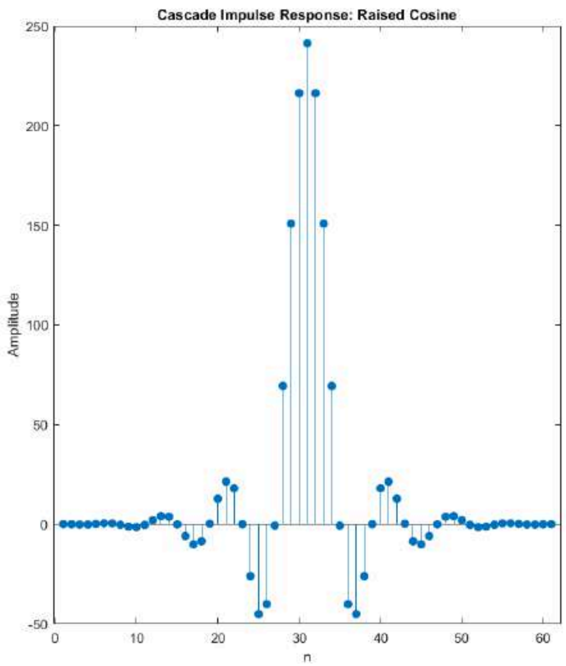
## 🔹 What the Output Shows

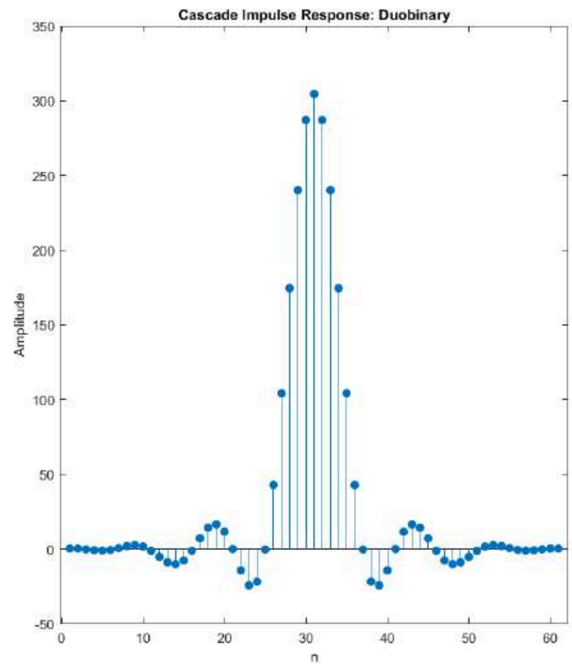
- **Figure (3 Subplots):**
  - **Impulse Response of g_T(n):**
    - Symmetric about the center.

        - Indicates **linear-phase FIR filter**.
        - Smooth decay on both sides of the main lobe.
    - **Normalized Magnitude Response of g_T(f):**
        - Passband is flat (good transmission within bandwidth).
        - Cosine roll-off shape defines the transition band.
        - Stopband attenuates unwanted frequencies → reduces interference.
    - **Cascade Impulse Response (g_T * g_R):**
        - Sharply peaked at the center.
        - Small sidelobes appear due to finite filter length.
        - Acts like a **Nyquist filter** → ensures minimal Inter-Symbol Interference (ISI).

- Photos of merge both problems :




Cascade Impulse Response: Raised Cosine
Amplitude
n


Cascade Impulse Response: Duobinary
Amplitude
n