EE301 Assignment Report

Communication System

Team Members:

Harsh Gupta - 12140750

Hemanth Gaddey - 12140660

Dhruv Gupta - 12140580

Rounak R. Kamble - 12141410

Akshat Arora - 12140170

8th November, 2023

Introduction

We transmitted the binary string ‘111010110111’ from one arduino to another and varied the time period of pulses as:

T = 20 ms , T = 200 microseconds and T = 20 microseconds.

We aimed at transmitting and studying the losses.

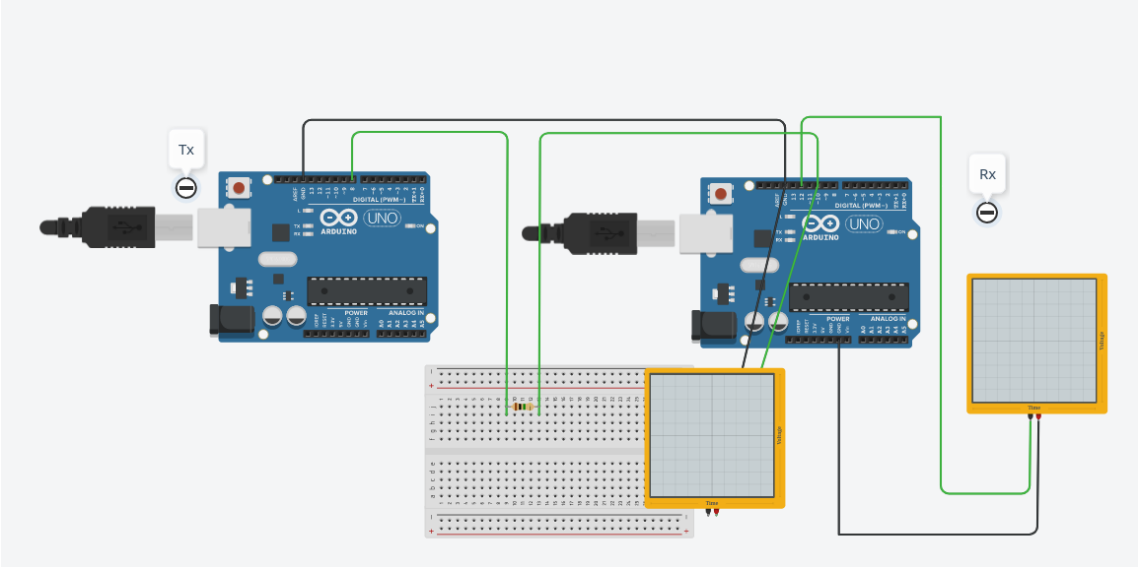
Transmitter Code:

```
#define T 10000

String t =  "111010110111";
int n;

void setup() {
  pinMode(8, OUTPUT);
  Serial.begin(9600);
  n = t.length();
}

void loop() {
  for(int i = 0; i<n; i++){
      if(t[i] == '1'){
          digitalWrite(8, HIGH);
      }
      else{
          digitalWrite(8, LOW);
      }
      delayMicroseconds(T);
  }
}
```

Receiver Code:

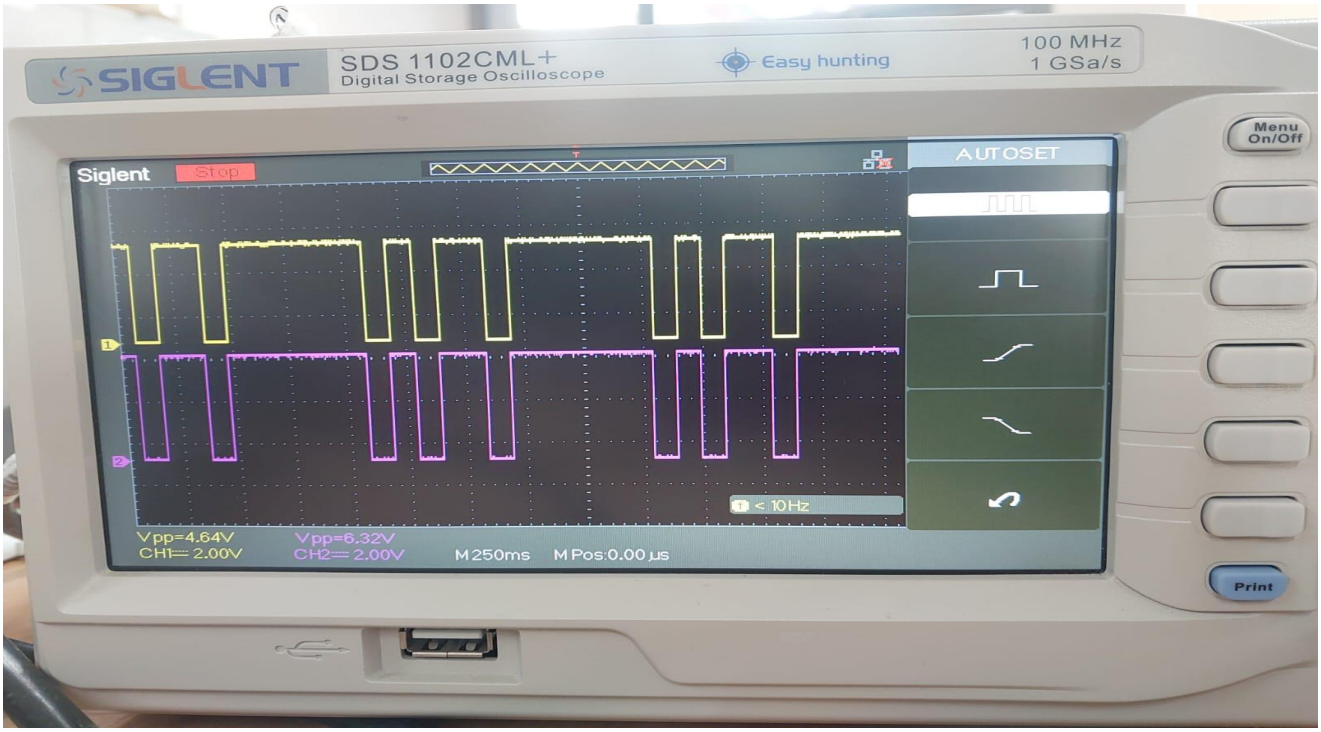
```
const int ipin = 10;
const int opin = 12;
const int TimePeriod = 10000; //Unit: microseconds

void setup() {
  pinMode(ipin, INPUT);
  pinMode(opin, OUTPUT);
  Serial.begin(9600);
}
int signal;
int n = 0;
void loop() {
      delayMicroseconds(TimePeriod);
      signal = digitalRead(ipin);
      digitalWrite(opin,signal);
      Serial.println(signal);
}
```

Circuit:



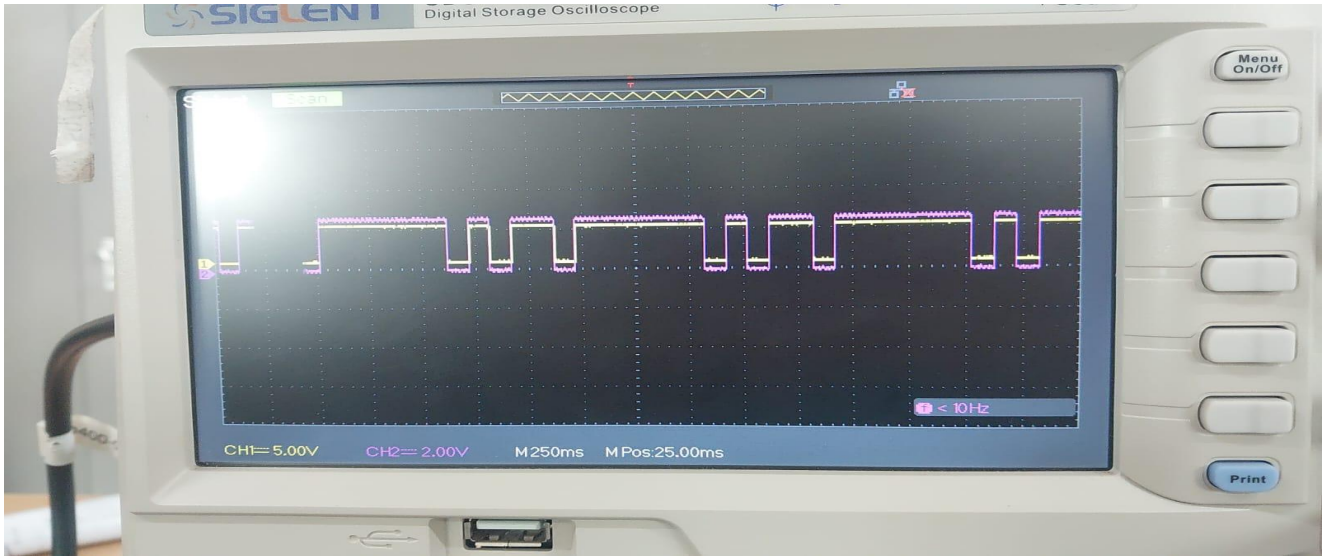
T = 100ms (10,000µs):

(i) Very Low Resistance in transmission wire:



Yellow: Signal the Receiver Arduino is receiving
Purple: Signal the Transmitter Arduino is Transmitting
This is transmitted using 3 m single core wire, which offers very minimal resistance, hence no visible attenuation.Both appear to be in sync with no loss in waveform.

(ii)1 MΩ Resistance in transmission wire:



Yellow: Signal the Receiver Arduino is receiving
Purple: Signal the Transmitter Arduino is Transmitting

Here we added 1MΩ resistance so attenuation can be seen, but as T = 10 ms, the frequency content has major band width around 1/T that is 10 Hz and if we model our system(channel of transmission) as low pass filter with R = 1MΩ and parasitic capacitance of wire calculated using the link:
https://www.emisoftware.com/calculator/wire-pair-capacitance/

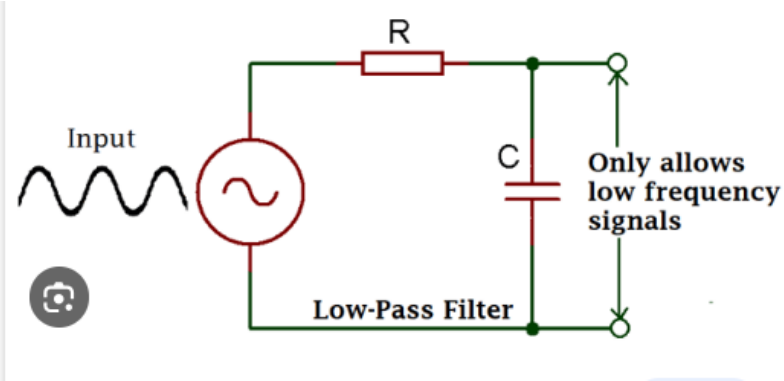
Inputs

Wire 1 radius	r_{w1}	0.5	mm
Wire 2 radius	r_{w2}	0.5	mm
Separation	s	5	cm
Length	l	3	m
Relative Permittivity	ϵ_r	1	

Outputs

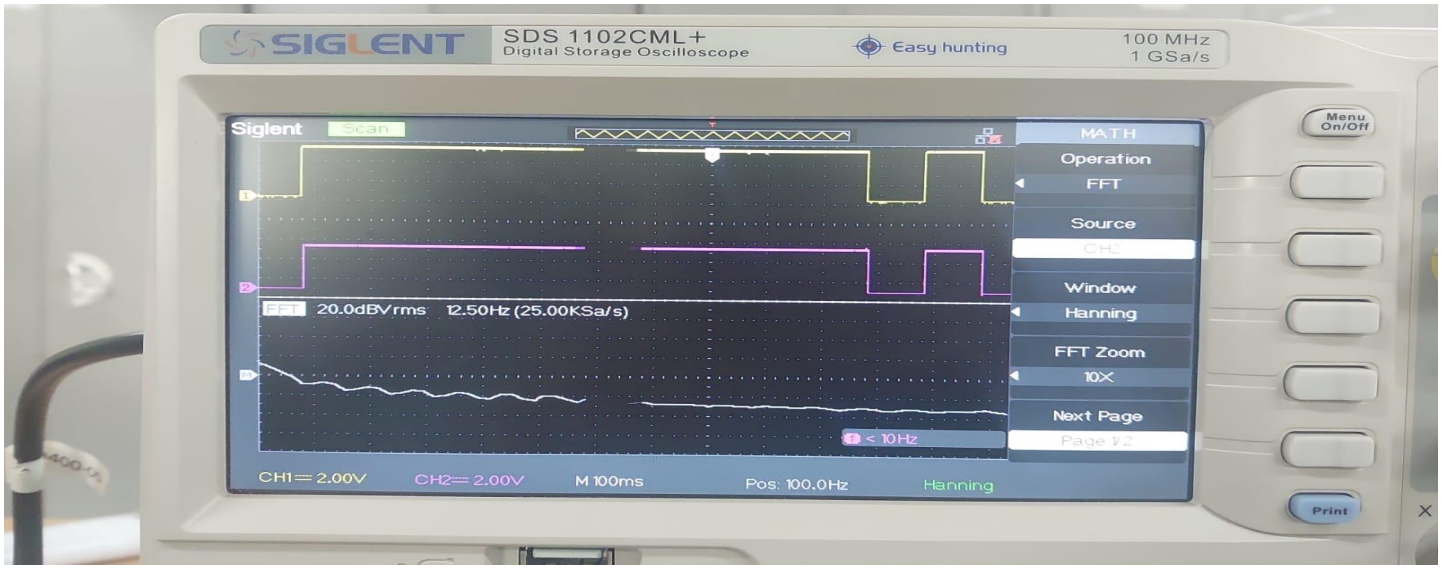
Capacitance: C = **1.810e-11 F**

the cut-off frequency is 1/(2*pi*R*C) = 8.793 KHz, which is way larger than 10Hz. So the 10 Hz content is passed and the signal is well reconstructed.

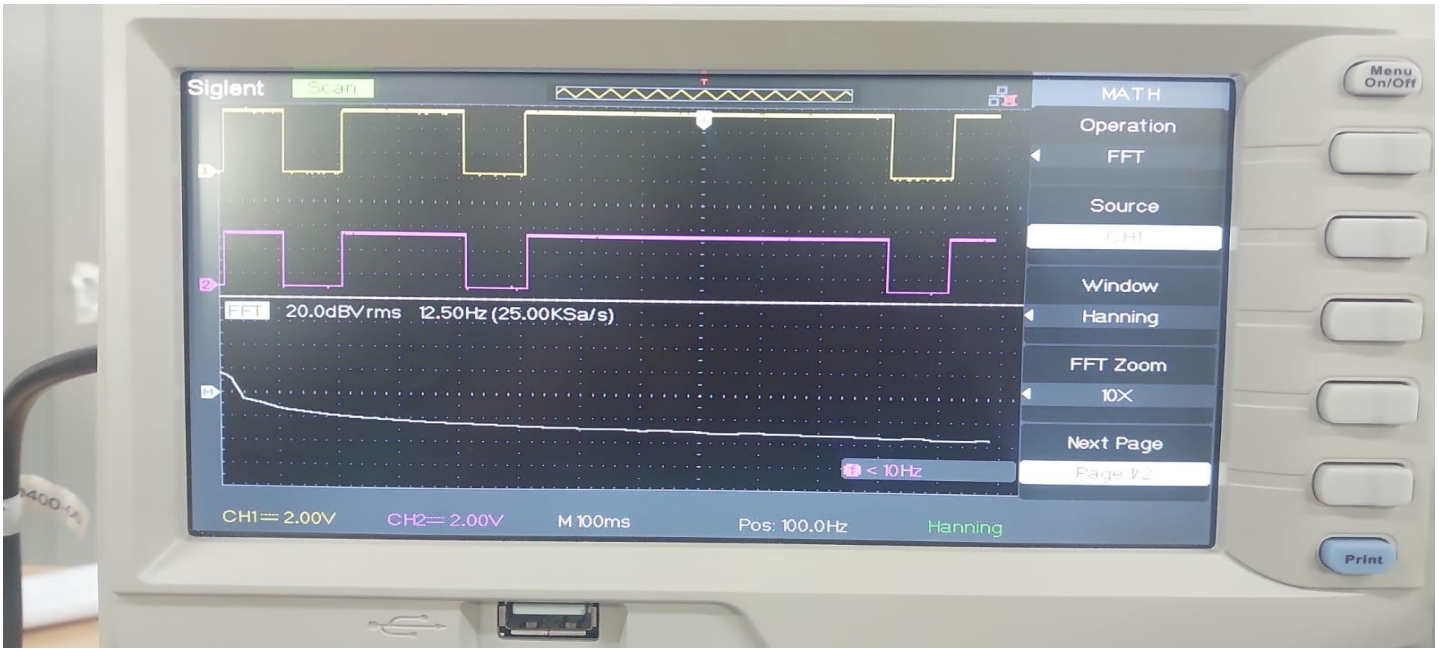


This can be seen as from the frequency analysis of the signal in oscilloscope:

Frequency Analysis for the above case:



Frequency spectrum for **Purple** signal (Transmitter signal)



Frequency spectrum for **Yellow** signal (Received signal)

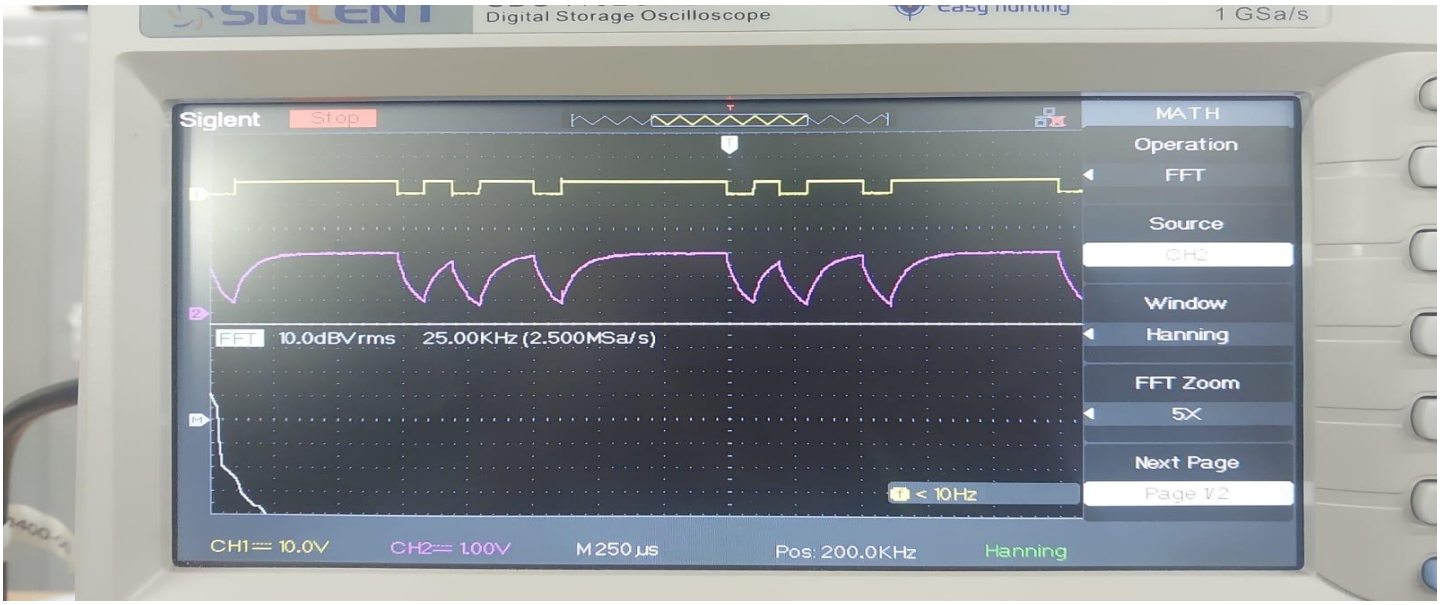
Both the channels have more or less the same frequency spectrum.

T = 100µs:

To study the losses well we scaled down the T to 100 microseconds although arduino's analog read(ADC) has a sampling rate of around 100 microseconds (on the receiver side) (source: www.arduino.cc/reference/en/language/functions/analog-io/analogread/), yet was done to study the losses at the end of the channel of 3 m.

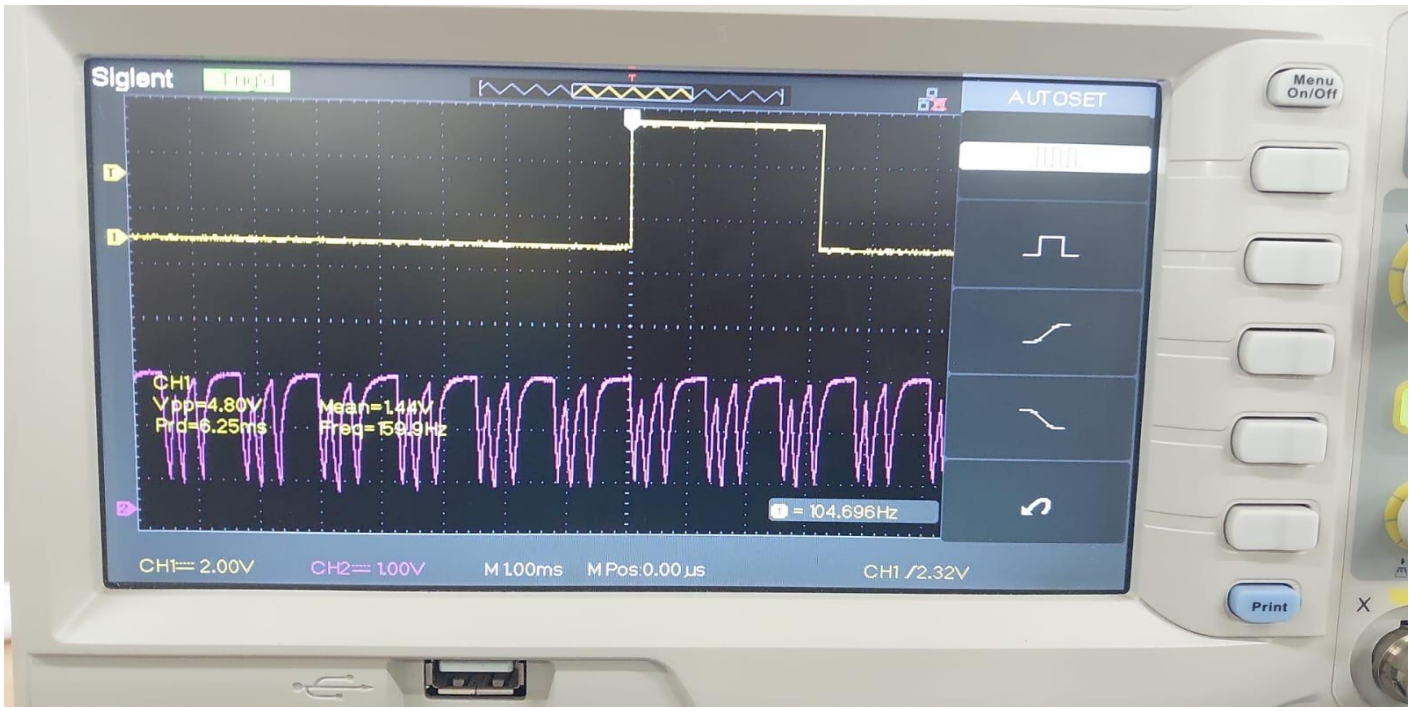
For T = 100 microseconds, the frequency bandwidth is about 10 kHz and as modeled previously the channel acts like a low pass filter with cutt-off frequency of 8.79 KHz. So some frequencies will be attenuated resulting in not so well constructed signal this time. This can also be seen as charging and discharging of RC circuits in time domain.

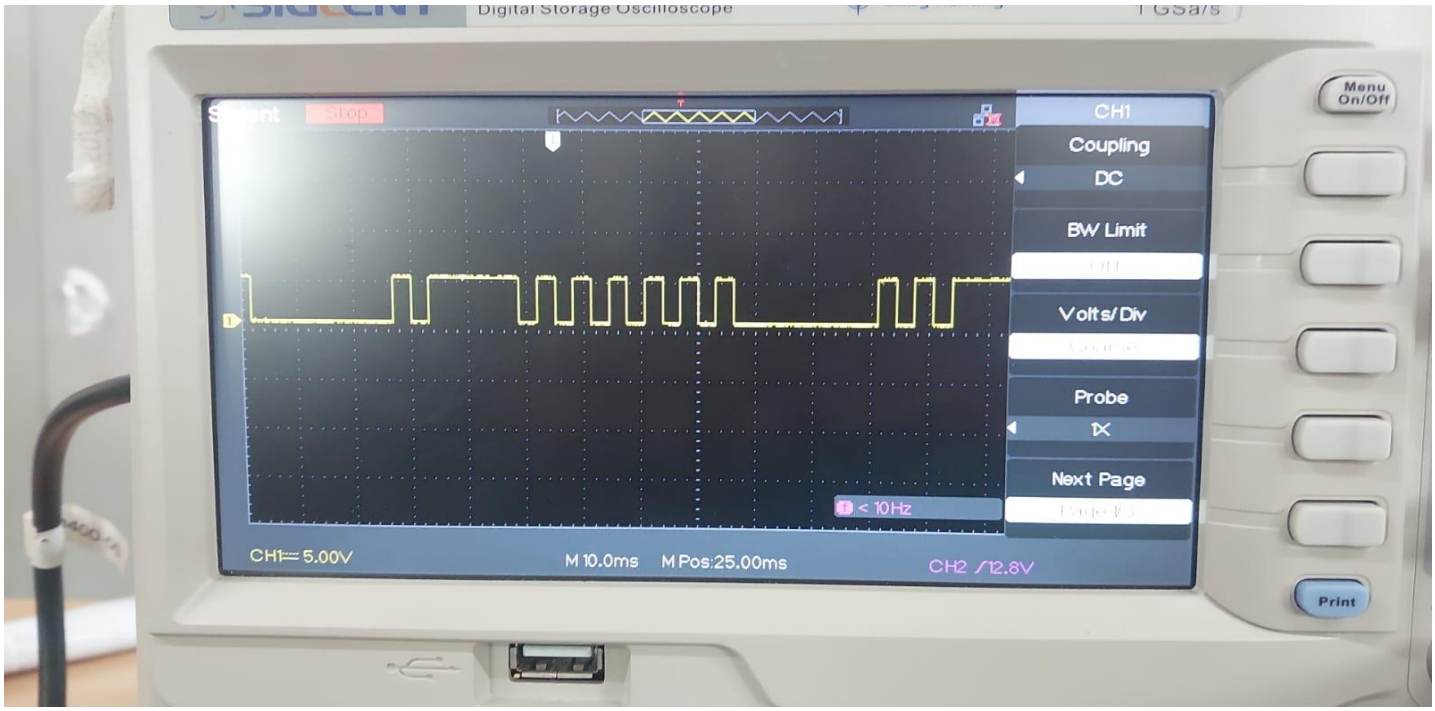
Frequency spectrum for **Yellow** signal (Signal at starting of the channel, Transmitted Signal)



Frequency spectrum for **Purple** signal (Signal at the end of the channel, Rx Arduino is receiving)

When this wave form (purple) is given to the arduino we can see the incorrect sequence due to sampling and distorted waveform received.





In both the above pictures,**Yellow** signal is the signal Rx Arduino is outputting while the **Purple** signal is the signal the receiver Arduino receives from the transmitter arduino.

To make comparison easy:

Yellow: Signal the Receiver Arduino is Outputting

Purple: Signal the Receiver Arduino is getting from the channel

Frequency spectrum for **Purple** signal (signal from Receiver Arduino)

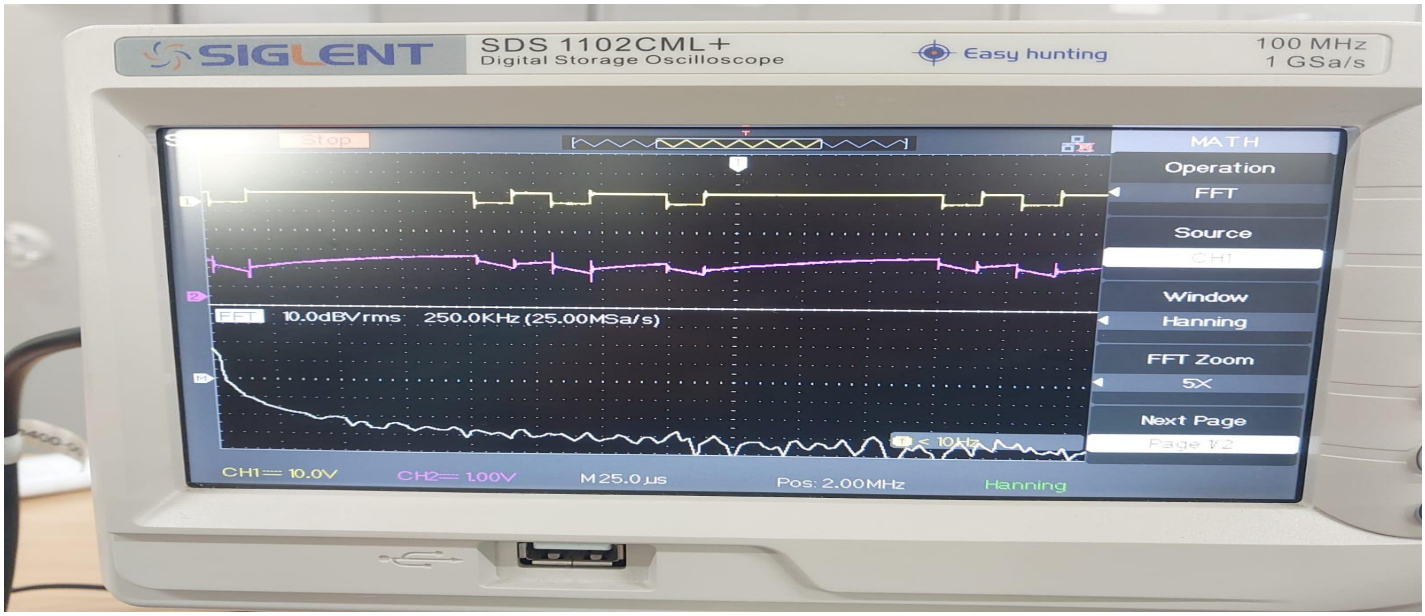
Frequency spectrum for **Yellow** signal (Transmitted signal)

T = 10µs:

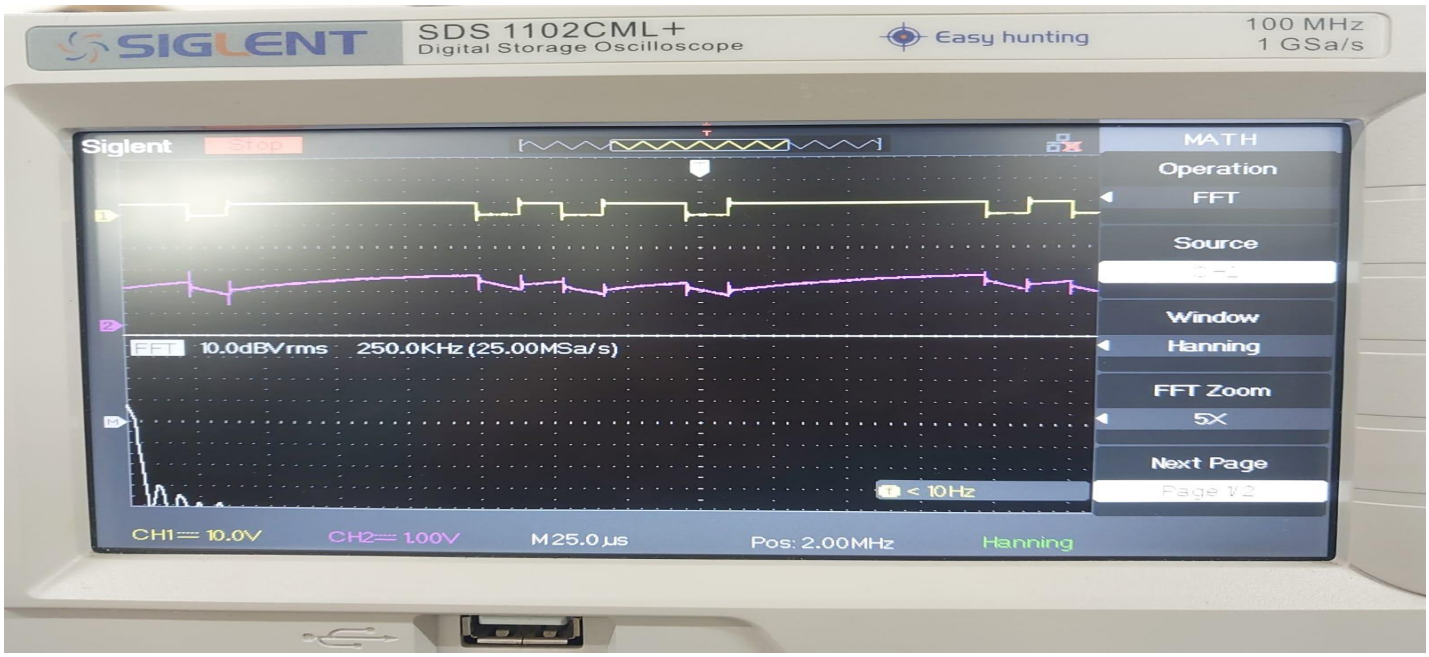
Yellow: Signal the Transmitter Arduino is Transmitting

Purple: Signal at the end of the channel

Pulse with T = 10 microseconds bandwidth is about 100 kHz so lot of frequency is cut off so the wave is more distorted than than T = 100 microseconds.

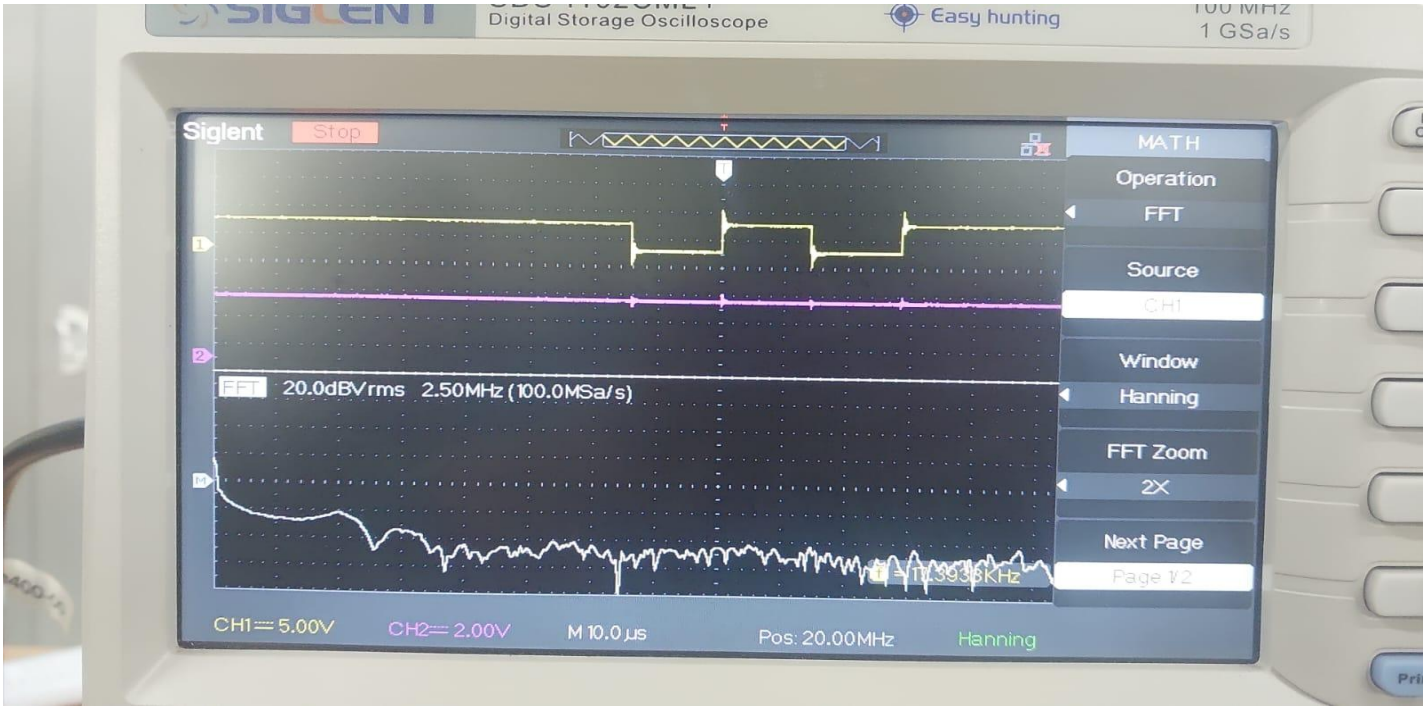


Frequency spectrum for **Yellow** signal (Transmitted signal)



Frequency spectrum for **Purple** signal (signal at the end of the channel)

Now Observing the output of Receiver Arduino:



Frequency spectrum for **Yellow** signal (Transmitted signal)
As the analog read of arduino takes around 100 microseconds to read the signal, hence it completely fails to read the signal when T = 10 microseconds.

Frequency spectrum for **Purple** signal (Signal output from Receiver Arduino)

Gibbs Ringing Phenomenon:

It occurs when a sharp transition or edge in the signal is approximated by a series of discrete samples since such a sharp discontinuity in the time domain requires infinite frequency content, and in practice, it is not possible to sample infinite frequency content . The phenomenon manifests as oscillations or ripples near the transition edge, which are not present in the original continuous signal. These oscillations result from the system's attempt to represent the abrupt change with limited resolution.

The phenomena became more and more apparent as we kept on decreasing the Time period.

Our Setup: