Aim ↬ To perform PCM [Pulse Code Modulation] using Trainer Kit & verify the same using MATLAB.

Apparatus Required ↬ MATLAB, ST2802.

Theory ↬

Pulse Code Modulation (PCM) converts an analog signal into its digital form by sampling the continuous signal at regular intervals, quantizing each sample, and encoding it into binary format. This digital representation is suitable for storage and transmission in digital systems and is widely used in applications like digital telephony and audio processing.

The process starts with sampling, where the analog signal is measured at discrete intervals. According to the Nyquist theorem, the sampling rate must be at least twice the maximum frequency of the signal to prevent information loss. Each sample is then quantized, meaning it is rounded to the nearest value within predefined levels, with the number of levels determining the digital representation's resolution.

After quantization, the values are encoded into binary form, with the number of bits used depending on the desired accuracy. For instance, an 8-bit system can represent 256 levels, while a 16-bit system can represent 65,536 levels, enhancing fidelity.

Differential PCM (DPCM) improves upon standard PCM by encoding the difference between successive samples rather than their absolute values. This reduces redundancy and improves efficiency, especially in signals with small variations. Continuously variable slope delta modulation (CVSD) further enhances this process by using variable step sizes for quantization, adapting dynamically to changes in the signal amplitude. This results in better performance, particularly in low-bandwidth scenarios.

Mathematically, the PCM process can be represented as follows: the analog signal x(t) is sampled at intervals, yielding $x[n] = x(nT_S)$, where T_S is the sampling interval. The sampled values are then quantized into discrete levels $Q(x[n])$ and encoded into binary.

PCM provides robust, noise-resistant transmission compared to analog systems. However, quantization introduces distortion known as quantization noise, which can be minimized by increasing the number of quantization levels.

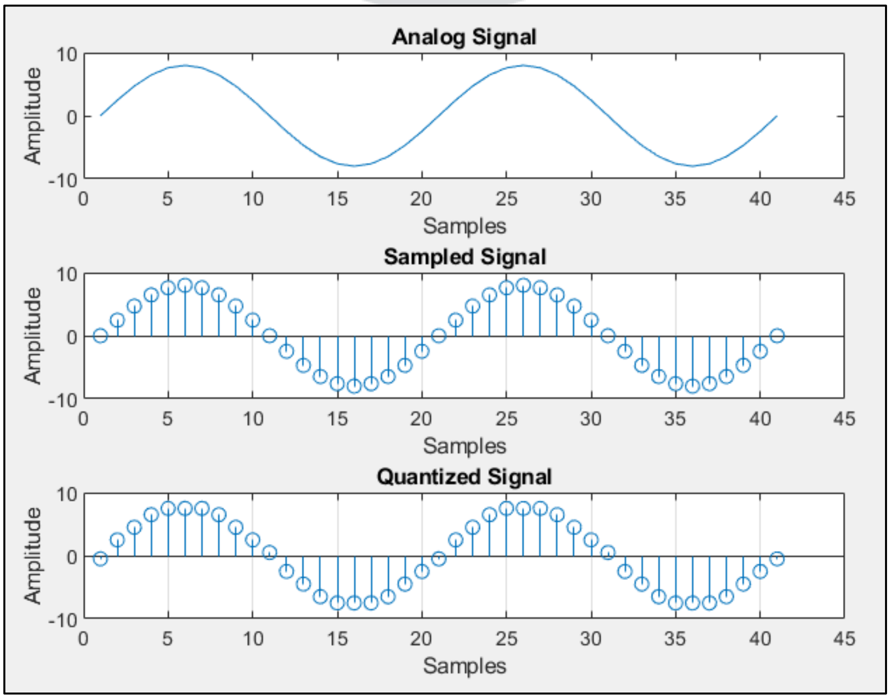
In practice, the PCM signal can be simulated using MATLAB to observe the digital representation of an analog signal. The process can also be demonstrated using hardware like the ST2802 Trainer Kit, allowing hands-on verification of digital encoding of analog signals.

Code ↬

```matlab
% Pulse Code Modulation
clc;
close all;
clear all;

n = 4;
n1 = 10;

L = 2^n;
Vmax = 8;
x = 0:pi/n1:4*pi;
ActualSignal = Vmax * sin(x);

figure;
subplot(3,1,1);
plot(ActualSignal);
title('Analog Signal');
xlabel('Samples');
ylabel('Amplitude');

subplot(3,1,2);
stem(ActualSignal);
grid on;
title('Sampled Signal');
xlabel('Samples');
ylabel('Amplitude');

Vmin = -Vmax;
StepSize = (Vmax - Vmin) / L;
QuantizationLevels = Vmin:StepSize:Vmax;
codebook = Vmin - (StepSize/2):StepSize:Vmax + (StepSize/2);

[ind, q] = quantiz(ActualSignal, QuantizationLevels, codebook);
NonZeroInd = find(ind ~= 0);
ind(NonZeroInd) = ind(NonZeroInd) - 1;
BelowVminInd = find(q == Vmin - (StepSize/2));
q(BelowVminInd) = Vmin + (StepSize/2);

subplot(3,1,3);
stem(q);
grid on;
```

```
title('Quantized Signal');
xlabel('Samples');
ylabel('Amplitude');

figure;
TransmittedSig = de2bi(ind, 'left-msb');
SerialCode = reshape(TransmittedSig', [1 size(TransmittedSig, 1) *
size(TransmittedSig, 2)]);

subplot(2,1,1);
grid on;
stairs(SerialCode);
axis([0 100 -2 3]);
title('Transmitted Signal');

RecievedCode = reshape(SerialCode, n, length(SerialCode) / n);
index = bi2de(RecievedCode', 'left-msb');
q = (StepSize * index);
q = q + (Vmin + (StepSize/2));

subplot(2,1,2);
grid on;
plot(q);
title('Demodulated Signal');
xlabel('Samples');
ylabel('Amplitude');
```

Output ↬



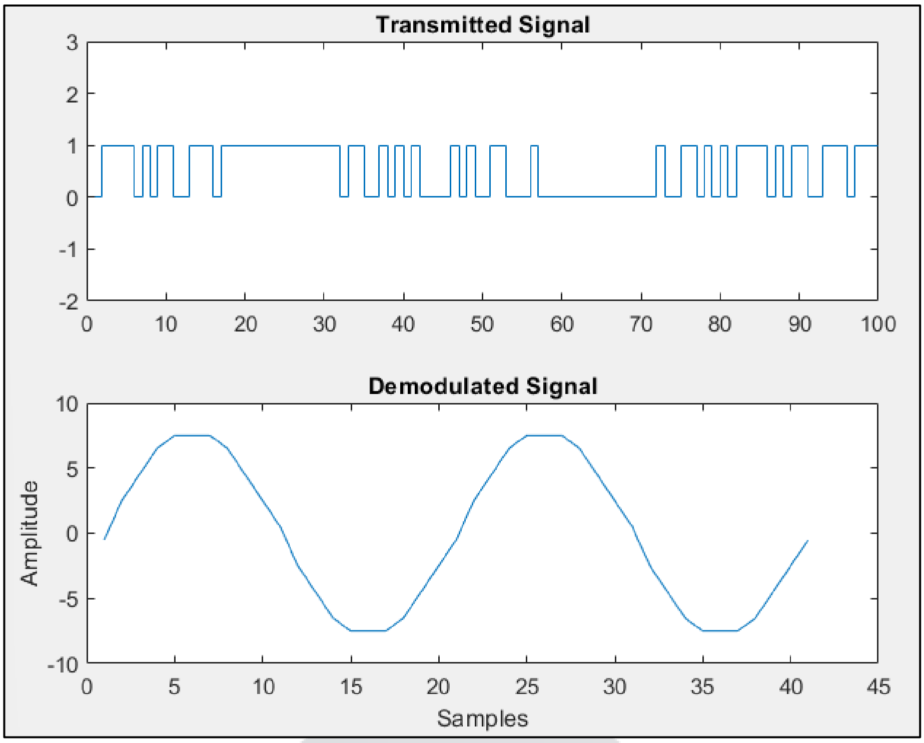


Fig. i) MATLAB Output

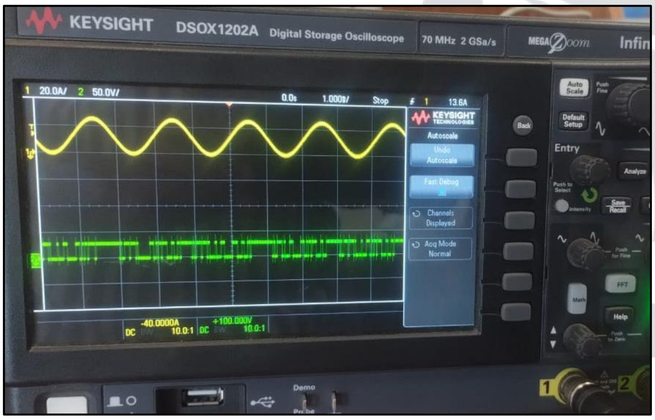


Fig. ii) Pulse Code Modulation

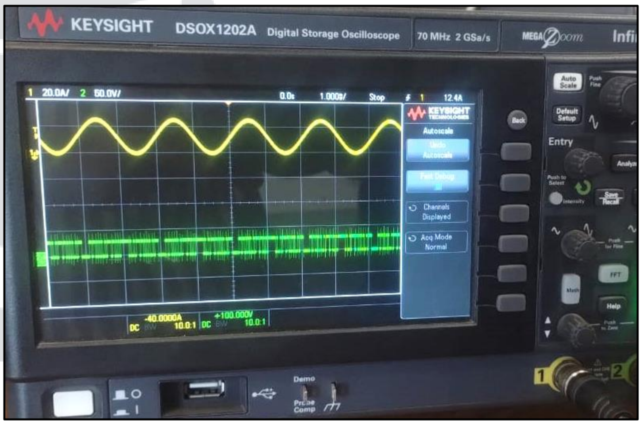


Fig. iii) Differential Pulse Code Modulation

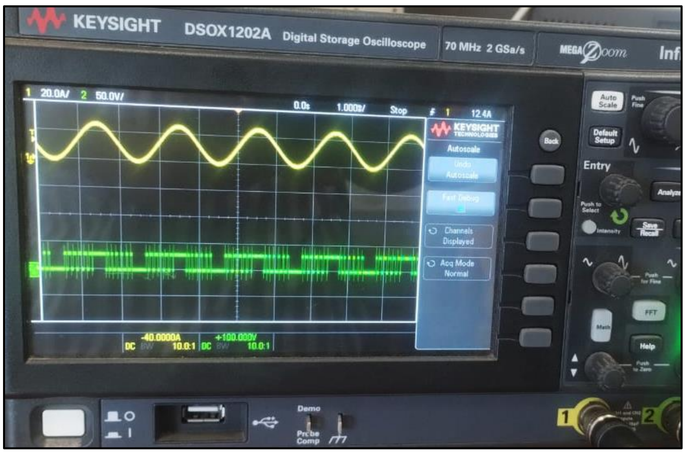


Fig. iv) Continuously Variable Slope Delta Modulation

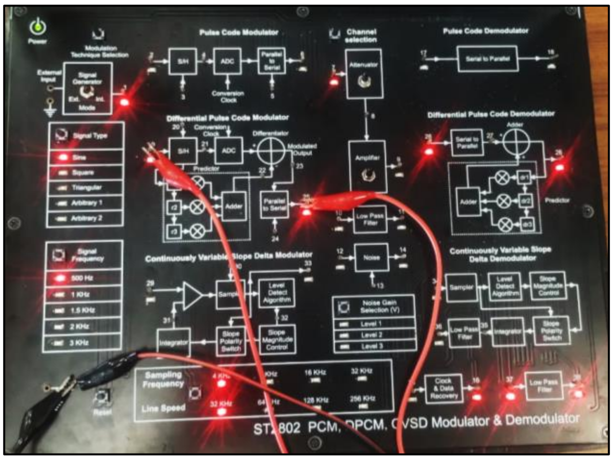


Fig. v) ST2802 Trainer Kit

Result ↬

The Pulse Code Modulation (PCM) process was successfully demonstrated using the ST2802 Trainer Kit and MATLAB. The analog input signal was sampled, quantized, and encoded into a digital format.

Conclusion ↬

PCM effectively converts an analog signal into its digital form for transmission and storage. Sampling, quantizing, and encoding the signal allow the analog information to be accurately represented in a binary format.

Precautions ↬

- Use proper sampling frequency (Nyquist criterion).
- Minimize quantization noise by choosing adequate levels.
- Check connections and settings on the Trainer Kit.
- Ensure MATLAB code accuracy for verification.