American International University- Bangladesh (AIUB) Faculty of Engineering (FE)

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | DATA COMMUNICATION | **Course Code:** | COE3103 |
| **Semester:** | Summer 2024-25 | **Section:** | D |
| **Faculty:** | MOHAMMAD ASADUZZAMAN KHAN | **Group:** | 05 |

|  |  |
| --- | --- |
| **Experiment No:** | 09 |
| **Experiment Name:** | Message Passing and Receiving Using Modulator. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Group Members** | | **Name** | **ID** |
|  | 1. | Basudeb Kundu | 23-50856-1 |
| 2. | Prohlad Chandra Das | 23-50922-1 |
| 3. | Debashis Kumar Das | 23-50953-1 |
| 4. | Indronill Dutta Nill | 23-50974-1 |
| 5. | Nafiur Rahman Nirob | 23-50991-1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Performance Date:** | 07-9-25 | **Due Date:** | 14-9-25 |

**Marking Rubrics (to be filled by Lab Instructor)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Category | Proficient [6] | Good [4] | Acceptable [2] | Unacceptable [1] | Secured Marks |
| **Theoretical Background, Methods & procedures sections** | All information, measures and variables are provided and  explained. | All Information provided is sufficient, but more explanation is  needed. | Most information is correct, but some information may be  missing or inaccurate. | Much information is missing and/or inaccurate. |  |
| **Results** | All of the criteria are met; results are described clearly and accurately; | Most criteria are met, but there may be some lack of clarity and/or incorrect information. | Experimental results don’t match exactly with the theoretical values and/or analysis  is unclear. | Experimental results are missing or incorrect; |  |
| **Discussion** | Demonstrates thorough and sophisticated understanding.  Conclusions drawn are appropriate for  analyses; | Hypotheses are clearly stated, but some concluding statements not supported by data or data not well  integrated. | Some hypotheses missing or misstated; conclusions not supported by data. | Conclusions don’t match hypotheses, not supported by data; no integration of data from different sources. |  |
| **General formatting** | Title page, placement of figures and figure captions, and other  formatting issues all correct. | Minor errors in formatting. | Major errors and/or missing information. | Not proper style in text. |  |
| **Writing & organization** | Writing is strong and easy to understand; ideas are fully elaborated and connected; effective transitions between sentences; no  typographic, spelling, or grammatical errors. | Writing is clear and easy to understand; ideas are connected; effective transitions between sentences; minor typographic, spelling, or grammatical errors. | Most of the required criteria are met, but some lack of clarity, typographic, spelling, or grammatical errors are present. | Very unclear, many errors. |  |
| Comments: |  | | | Total Marks (Out of **30**): |  |

**Title**

Message Passing and Receiving Using Modulator.

**Abstract:**

This experiment is designed

1. To understand the concept of message encoding and decoding.

2. To understand the concept of serial transmission and reception of message.

3. To develop understanding of data transmission and reception process.

**Introduction:**

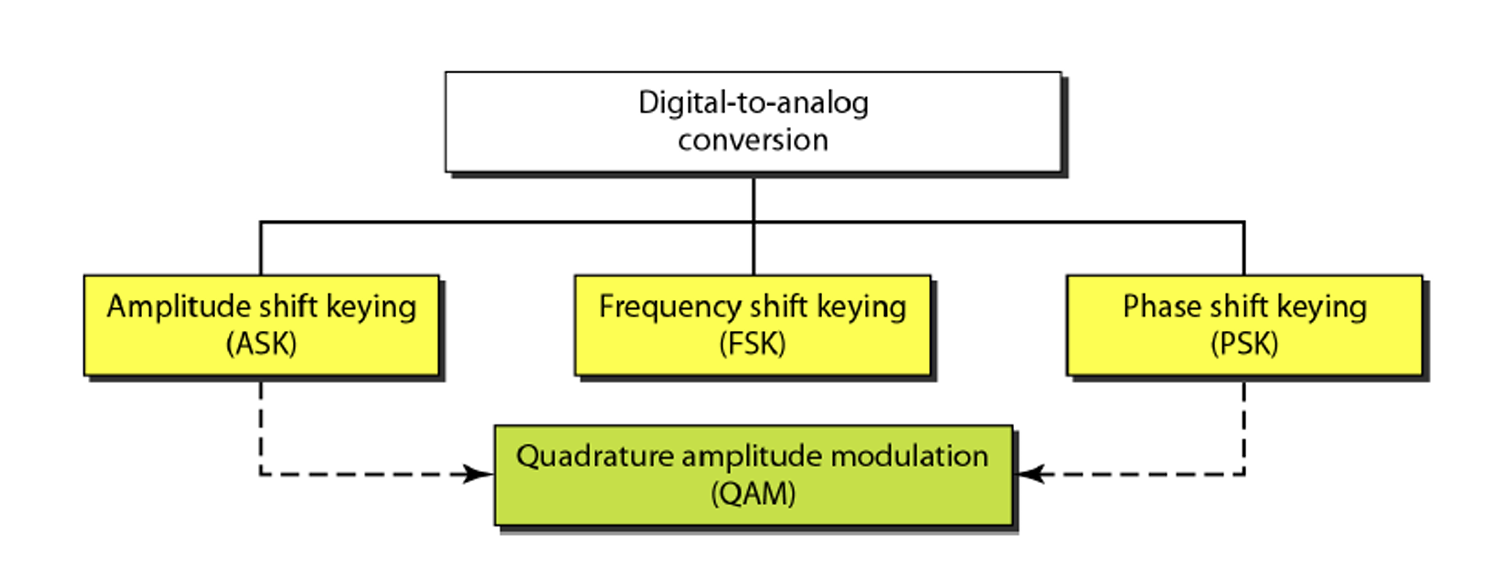
In digital communication systems, transmitting and receiving text messages over physical media requires a series of signal processing steps to ensure that the information is accurately conveyed. This experiment focuses on understanding the fundamental principles of message encoding, serial data transmission, modulation, demodulation, and message decoding.

A diagram of a computer code

AI-generated content may be incorrect.

***Fig1: Asynchronous* transmission**

The process begins at the transmitter end, where the input message is first encoded into binary form using ASCII codes. These binary sequences are then serialized and converted into digital signals. To effectively transmit the digital data over analog waveform channels such as twisted pair cables or wireless RF links, modulation techniques such as Amplitude Shift Keying (ASK) or Frequency Shift Keying (FSK) are applied.



**Fig2: Digital to Analog Conversion**

At the receiver side, the analog signal undergoes demodulation to recover the original binary sequence. The binary data is then decoded back into human-readable text. Due to the presence of channel noise, especially modeled here as Additive White Gaussian Noise (AWGN), the receiver must handle and correct for possible signal impairments.

This experiment provides practical experience in simulating the entire communication chain—from message encoding to signal modulation, channel noise modeling, and message recovery—thereby reinforcing theoretical concepts with hands-on implementation using MATLAB.

**Results and Discussion**

|  |
| --- |
| **Function bin2asc.m Code:** |
| function txt = bin2asc(dn)  %bin2asc Serial binary to ASCII to text conversion  %8 bits per char , LSB first  % >> txt= bin2asc(dn) <<  % where dn is binary input sequence  % txt is output text string  L=length(dn); %Length of input string  L8=8\*floor(L/8); %Multiple of 8 Length  B=reshape(dn(1:L8),8,L8/8); %Cols of B are bits of chars  p2=2.^(0:7); %power of 2  dec=p2\*B; %Binary to decimal conversion  txt=char(dec); %ASCII (decimal) to txt  end |

|  |
| --- |
| **Function asc2bn.m Code:** |
| function dn = asc2bn(txt)    dec=double(txt) %Text to ASCII (decimal)  p2=2.^(0:-1:-7); % 2^0,2^-1,.......,2^-7  B=mod(floor(p2'\*dec),2); %Decimal to binary conversion  %Columns of B are bits of chars  dn=reshape(B,1,numel(B));%Bytes to serial conbversion  end |

|  |
| --- |
| **Code:** |
| clc;  clear all;  close all;  %% --- Input Message ---  Transmitted\_Message = 'Deva';  disp('\*\*\*\*\*\*\*\*\*\*');  disp(['Original message at Transmitter: ', Transmitted\_Message]);  %% --- Convert Text to Binary ---  x = reshape(dec2bin(Transmitted\_Message,8).'-'0',1,[]);  %% --- Add Start(1) and Stop(0) Bits Asynchronously ---  async\_bits = [];  for k = 1:8:length(x)  byte = x(k:k+7);  framed = [1 byte 0]; % Start=1, Stop=0  async\_bits = [async\_bits framed];  end  disp('Asynchronous Binary at Transmitter (start=1, stop=0):');  disp(async\_bits);  %% --- Generate NRZ Signal ---  bp = 0.000001; % Bit period  br = 1/bp; % Bit rate  bit = [];  for n = 1:length(async\_bits)  if async\_bits(n) == 1  se = 5\*ones(1,100);  else  se = zeros(1,100);  end  bit = [bit se];  end  t1 = bp/100:bp/100:100\*length(async\_bits)\*(bp/100);  figure;  subplot(4,1,1);  plot(t1,bit,'LineWidth',2.5); grid on;  axis([0 bp\*length(async\_bits) -0.5 6]);  ylabel('Amplitude (volt)'); xlabel('Time (sec)');  title('Asynchronous Digital Signal (NRZ)');  %% --- ASK Modulation ---  A1 = 5;  A2 = 0;  f = br\*10;  t2 = bp/99:bp/99:bp;  Ns = length(t2);  m = [];  for i = 1:length(async\_bits)  if async\_bits(i) == 1  y = A1\*cos(2\*pi\*f\*t2);  else  y = A2\*cos(2\*pi\*f\*t2);  end  m = [m y];  end  t3 = bp/99:bp/99:bp\*length(async\_bits);  subplot(4,1,2);  plot(t3,m,'LineWidth',1.5); grid on;  axis([0 bp\*length(async\_bits) -6 6]);  xlabel('Time (sec)'); ylabel('Amplitude (volt)');  title('ASK Modulated Signal (Transmitter Output)');  %% --- Add Noise (AWGN Channel) ---  SNR\_dB = 10;  signal\_power = mean(m.^2);  SNR\_linear = 10^(SNR\_dB/10);  noise\_power = signal\_power / SNR\_linear;  noise = sqrt(noise\_power) \* randn(size(m));  Rec = m + noise;  t4 = bp/99:bp/99:bp\*length(async\_bits);  subplot(4,1,3);  plot(t4,Rec); grid on;  axis([0 bp\*length(async\_bits) -6 6]);  xlabel('Time (sec)'); ylabel('Amplitude (volt)');  title('Received Signal at Receiver');  %% --- Demodulation ---  mn = [];  for n = 1:Ns:length(Rec)  carrier = cos(2\*pi\*f\*t2);  segment = Rec(n:n+Ns-1);  mm = carrier .\* segment;  z = trapz(t2, mm);  zz = round((2\*z/bp));  if (zz > (A1+A2)/2)  a = 1;  else  a = 0;  end  mn = [mn a];  end  disp('Raw demodulated bits at Receiver:');  disp(mn);  %% --- Remove Start and Stop Bits ---  mn\_clean = [];  for k = 1:10:length(mn) % 8 data + 2 framing bits  byte = mn(k:k+9); % full framed byte  data\_bits = byte(2:9); % remove start & stop  mn\_clean = [mn\_clean data\_bits];  end  disp('Cleaned data bits at Receiver (start/stop removed):');  disp(mn\_clean);  %% --- Convert Bits Back to Text ---  received\_text = char(bin2dec(reshape(char(mn\_clean+'0'),8,[]).')).';  disp('\*\*\*\*\*\*\*\*\*\*');  disp(['Recovered message at Receiver: ', received\_text]);  disp('\*\*\*\*\*\*\*\*\*\*');  %% --- Plot Final Demodulated Bits ---  bit = [];  for n = 1:length(mn\_clean)  if mn\_clean(n) == 1  se = 5\*ones(1,100);  else  se = zeros(1,100);  end  bit = [bit se];  end  t5 = bp/100:bp/100:100\*length(mn\_clean)\*(bp/100);  subplot(4,1,4);  plot(t5,bit,'LineWidth',2.5); grid on;  axis([0 bp\*length(mn\_clean) -0.5 6]);  ylabel('Amplitude (volt)'); xlabel('Time (sec)');  title('Demodulated Signal at Receiver (Data Bits Only)'); |

|  |
| --- |
| **Result and Discussion:** |
| **Fig3: Binary Information at Transmitter** |
| **Fig4: Binary Information at Receiver** |
| **Fig5: Original & Received Message** |
| **Fig6: Asynchronous Communication** |
| The experiment successfully demonstrated the transmission and reception of asynchronous data. The message “Deva” was encoded into ASCII, framed with start and stop bits, and transmitted as an asynchronous digital signal. The signal was modulated using ASK and passed through an AWGN channel to simulate real-world noise conditions. At the receiver, the noisy signal was demodulated, and the framing bits were removed to recover the original binary data. The recovered message matched the transmitted text, verifying the correctness of the asynchronous communication process. Although channel noise introduced slight distortions, the threshold detection ensured accurate symbol recovery. This validates the complete flow of asynchronous encoding → framing → modulation → transmission → demodulation → decoding. |

**Conclusion**

The experiment successfully showcased the full cycle of asynchronous data communication. A text message was converted into binary, framed with start and stop bits, and transmitted using Amplitude Shift Keying (ASK) modulation through a noisy channel. At the receiver, the signal was accurately reconstructed, and the original message was recovered. Despite channel noise, threshold detection ensured correct decoding, confirming the robustness of the method. Overall, the experiment validates the reliability of asynchronous transmission for character-based data exchange and emphasizes its practical significance in digital communication systems.