CECS 460 – Lab 5: LUT-Based BPSK Communication System with Logic Analysis

Learning Objectives

- Implement hardware-assisted modulation using LUTs on an ESP-32.
- Design a receiver that demodulates BPSK signals in firmware using LUT(s).
- Integrate a real-time digital logic analyzer peripheral for debugging and verification.
- Demonstrate end-to-end embedded communication and synchronization between three SoC nodes.

Team Roles and Responsibilities

Student	ESP-32 Role	Primary Tasks	Deliverable
A	Transmitter (BPSK TX)	Generate BPSK waveform using LUT; modulate UART data from PC; transmit over GPIO pin at 100–500 kHz.	TX firmware + UART-to-BPSK report
B	Receiver (BPSK RX)	Sample incoming GPIO waveform; recover bitstream using phase detection or zero-cross; forward data to PC.	RX firmware + demodulation documentation
C	Logic Analyzer / Debug Node	Sample both TX and RX GPIO lines; display timing on PC (GUI).	Analyzer firmware + visualization tool

Hardware Setup

• 3 × ESP-32 DevKit v1 boards
• Breadboard + jumper wires
• Common ground between all ESP-32s
• TX pin (GPIO 25) → RX pin (GPIO 26) via short wire
• Logic analyzer ESP taps both lines
• 3 USB cables to 3 PCs running serial monitors or apps

PC1 ↔ UART ↔ ESP32-TX →(GPIO wire)→ ESP32-RX ↔ UART ↔ PC2

↓

ESP32-LA ↔ PC3

Part 1 – BPSK Transmitter (Student A)

Use a LUT to generate the carrier frequency, your choice, and then determine best method to 'flip' carrier to indicate bit change. Determine message start, sync, data format.

UART bytes entered in program from the PC drive a bitstream. Each bit selects either +phase or 180° phase-shifted version of the LUT. Output waveform, potential methods = dacWrite(), PWM at the appropriate bit rate or ???.

Example Code (caution, sample only):

```
const uint8_t sineLUT[64] = {128,140,152,164,...};
for(each bit){
  phase = (bit) ? 0 : 32; // 180° offset
  for(i=0;i<64;i++){
    dacWrite(25, sineLUT[(i+phase)%64]);
    delayMicroseconds(2);
  }
}
```

Part 2 – BPSK Receiver (Student B)

Sample the input signal and detect phase reversals by sign changes or correlation with a reference LUT. Reconstruct the bitstream and send decoded bytes to PC via UART.

Example Code (caution, sample only):

```
if (prevSample * currSample < 0) phaseFlip = true;
bitValue = phaseFlip ? 0 : 1;
```

Part 3 – Mini Logic Analyzer (Student C)

Sample two GPIO inputs (TX, RX) into a circular buffer. Stream via UART to PC for visualization in Python or Processing. Trigger on TX start bit for alignment.

Assessment Rubric (50 pts)

Category	Points	Description
Functionality	20	End-to-end data transfer verified
Design Documentation	6	Diagrams, timing, LUT description
Code Quality	4	Modular, readable C/Python

Verification	10	Analyzer capture + analysis
Presentation	10	Team demo + explanation

Optional Extensions

- Add Manchester or QPSK encoding.
- Implement spectrum display via FFT on PC.
- Synchronize ESP-32 boards over Wi-Fi and compare wireless vs. wired latency.