CECS 460 – Lab 6: System Verification and Packet-Based Communication

Fall 2025 | ESP-32 DevKit v1 Platform

Objective

Design, implement, and verify a packet-based BPSK communication link using three ESP-32 DevKit v1 modules acting as transmitter (TX), receiver (RX), and logic analyzer (LA). Students will perform system-level verification by measuring timing, validating checksums, and analyzing performance/power trade-offs.

Background

In previous labs, you implemented UART communication and LUT-based BPSK modulation/demodulation. This lab introduces packet framing, checksum verification, and timing analysis using GPIO instrumentation. It connects those implementations to SoC-level system verification concepts.

System Setup

Role	Hardware	Connection
TX	ESP-32 DevKit v1	GPIO 23 → Signal Line
RX	ESP-32 DevKit v1	GPIO 22 ← Signal Line
LA	ESP-32 DevKit v1	GPIO 21 → Monitor Line
Common GND	Shared across all boards	—

TX sends a single-wire BPSK signal to RX. LA taps into the same line to monitor transitions.

Tasks

1. Packet Framing

[SYNC][LENGTH][DATA0…DATA7][CHECKSUM]
SYNC byte: 0xAA (10101010)
LENGTH: number of data bytes (≤ 8)
DATA: ASCII or binary payload
CHECKSUM: XOR of all bytes (SYNC excluded)

2. Transmission (TX)

- Use LUT-based BPSK modulator from Lab 5.
- Frame data and send continuously at a fixed baud-like rate.
- Toggle GPIO before/after transmission for timing capture.

3. Reception (RX)

• Detect SYNC pattern to align to frame start.
• Store DATA bytes, compute checksum, and compare.
• If valid → toggle green LED / UART print 'OK'.
• If invalid → toggle red LED / UART print 'ERR'.

4. Logic Analyzer (LA)

• Sample TX line and timestamp edges.
• Record TX and RX response signals.
• Export waveform to CSV or capture screenshot.

5. System-Level Verification

Measure and report:
• Frame latency (TX → RX)
• Error-free frames / total frames
• CPU load or ISR execution time using esp_timer_get_time().

Deliverables

1. Demonstration video (1–2 minutes) showing transmission, reception, and logic analyzer timing.
2. Lab Report (~2 pages + figures) including block diagram, waveform screenshots, results, and trade-off discussion.
3. Code submission (clean, commented ESP-IDF or Arduino source) with README.md.

Evaluation Rubric (100 pts total)

Category	Description	Points
Functionality	TX/RX/LA operate as specified; checksum validation works.	40
Verification Evidence	Timing plots, error rate data, and checksum validation shown.	25
Documentation	Report clarity, block diagram, results table, and trade-off discussion.	20
Code Quality	Readable, commented, and organized source files.	10
Professionalism	Demo clarity and submission organization.	5

Reflection Questions

- How did you verify system timing and data integrity without Vivado tools?
- Which parameter most impacted reliability — baud rate, ISR length, or signal integrity?
- How do your observations map onto the Performance–Power–Cost trade-off triangle?