UART Communication Stress Test with NodeMCU ESP8266

Lab-03

Name: Md. Samir Hossain
ID: 2022-3-60-161
Course: CSE406 (Internet of Things)
Section: 01
Semester: Summer-2025

Dr. Raihan Ul Islam
Associate Professor
Department of Computer Science & Engineering

July 22, 2025

1 Introduction

UART (Universal Asynchronous Receiver-Transmitter) is a serial communication protocol that facilitates asynchronous data exchange between devices via TX (transmit) and RX (receive) pins, with the baud rate dictating the transmission speed. The objective of this lab was to assess UART communication performance by measuring throughput (bytes per second), transfer speed (messages per second), and error rate (percentage of failed or corrupted messages). This was achieved through a stress test involving two NodeMCU ESP8266 boards connected using expansion boards and a breadboard, with test parameters including baud rates of 9600, 38400, and 115200; message sizes of 10, 50, and 100 bytes; and transmission intervals of 0, 10, and 100 ms.

2 Methodology

2.1 Hardware Setup

Two NodeMCU ESP8266 boards were interconnected using expansion boards and a breadboard. The wiring configuration linked NodeMCU 1's D5 (TX) to NodeMCU 2's D6 (RX), NodeMCU 2's D5 (TX) to NodeMCU 1's D6 (RX), and established a shared ground (GND) via the breadboard. Proper wiring was verified to prevent communication issues such as loose connections or pin mismatches.

2.2 Software Setup

The Arduino IDE was utilized to program NodeMCU 1 as the Master and NodeMCU 2 as the Slave. The Master transmitted messages with sequence numbers across varying baud rates, message sizes, and intervals, while the Slave echoed these messages back. Output was logged using CoolTerm into files named `nodemcu1_output.txt` and `nodemcu2_output.txt`. Baud rate synchronization was managed by the Master sending commands like `SETBAUD:38400`, with the Slave responding `OK:38400` before switching rates.

3 Results

The stress test outcomes are presented in the table below, detailing throughput, message rate, and error rate for each combination of baud rate, message size, and interval.

Table 1: Results for Baud Rate 9600

Message Size (bytes)	Interval (ms)	Throughput (bytes/s)	Message Rate (msg/s)	Error Rate (%)
10	0	412.00	42.30	0.24
10	10	320.00	33.10	0.00
10	100	75.50	8.50	0.00
50	0	484.10	9.90	0.00
50	10	440.00	9.00	0.00
50	100	244.00	5.00	0.00
100	0	484.10	4.90	0.00
100	10	464.30	4.70	0.00
100	100	325.70	3.30	0.00

Table 2: Results for Baud Rate 38400

Message Size (bytes)	Interval (ms)	Throughput (bytes/s)	Message Rate (msg/s)	Error Rate (%)
10	0	1607.20	156.20	0.06
10	10	643.00	65.40	0.00
10	100	85.40	9.60	0.00
50	0	1909.00	38.40	0.00
50	10	1379.00	27.80	0.00
50	100	391.00	8.00	0.00
100	0	1919.00	19.30	0.00
100	10	1609.00	16.20	0.00
100	100	652.40	6.60	0.00

Table 3: Results for Baud Rate 115200

Message Size (bytes)	Interval (ms)	Throughput (bytes/s)	Message Rate (msg/s)	Error Rate (%)
10	0	107.00	11.80	44.92
10	10	156.00	16.70	42.51
10	100	61.10	6.90	52.17
50	0	317.50	6.50	80.00
50	10	263.60	5.40	85.19
50	100	253.80	5.20	88.46
100	0	98.00	1.00	100.00
100	10	98.00	1.00	100.00
100	100	98.00	1.00	100.00

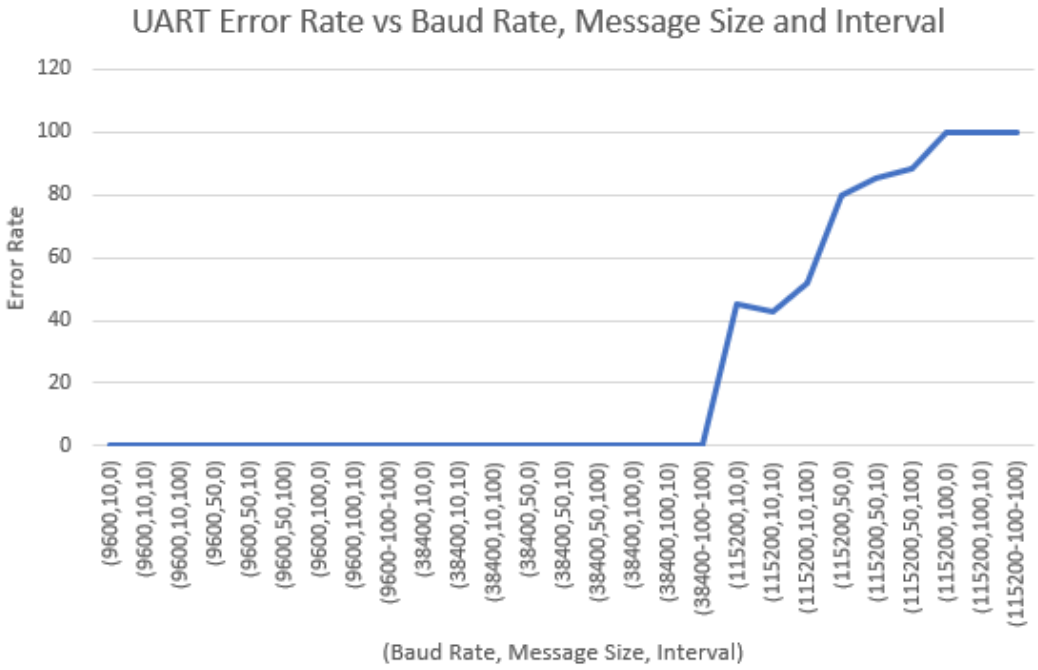

Figure 1: UART Error Rate vs Baud Rate, Message Size, and Interval

Observations
Throughput generally increases with higher baud rates and larger message sizes, peaking at 1919.00 bytes/s for 38400 baud with 100-byte messages and 0 ms interval, but drops significantly at 115200 baud due to elevated error rates. - Message rate decreases as message size and interval increase, with a maximum of 156.20 msg/s at 38400 baud, 10 bytes, and 0 ms interval. - Error rates are minimal (0–0.24%) at 9600 and 38400 baud but surge at 115200 baud, reaching 100% for 100-byte messages, indicating setup unreliability at high speeds.

4 Analysis

4.1 Throughput

Throughput escalates with baud rate and message size, achieving up to 1919.00 bytes/s at 38400 baud with 100-byte messages and 0 ms interval. At 115200 baud, however, throughput diminishes due to high error rates, constrained by SoftwareSerial's buffer limitations and potential breadboard-induced noise.

4.2 Transfer Speed

Message rate peaks with smaller messages and shorter intervals, such as 156.20 msg/s at 38400 baud with 10-byte messages and 0 ms interval. Larger messages reduce the rate due to increased transmission time, illustrating a trade-off between speed and data volume per message.

4.3 Error Rate

Error rates remain low at 9600 and 38400 baud (0–0.24%) but climb sharply at 115200 baud, up to 100% for 100-byte messages. This suggests SoftwareSerial struggles with high-speed data, possibly due to buffer overflows or electrical noise. Mismatch errors (e.g., receiving '6;DXXX' instead of '6:DXXX') were mitigated by trimming extra characters like \r\n.

4.4 Best Configuration

The optimal setup is 38400 baud with 50-byte messages and 10 ms intervals, yielding 1379.00 bytes/s throughput, 27.80 msg/s, and 0% error rate, balancing performance and reliability effectively.

4.5 Challenges

Key challenges included synchronizing baud rates between Master and Slave and managing high error rates at 115200 baud. Mismatch errors from extra characters required data trimming, highlighting SoftwareSerial and hardware setup limitations.

5 Conclusion

The stress test confirmed UART's efficacy at moderate baud rates, with 38400 baud, 50-byte messages, and 10 ms intervals achieving 1379.00 bytes/s, 27.80 msg/s, and 0% error rate. At 115200 baud, excessive errors underscored SoftwareSerial's inadequacy for high-speed communication in this configuration. Future enhancements could involve hardware UART, improved wiring to reduce noise, or error correction techniques.

6 References

- NodeMCU Documentation: `https://nodemcu.readthedocs.io/en/release/`