

# Taller de Detección de Errores y Tecnologías RS-232

1<sup>st</sup> Santiago Hernández Ávila  
Facultad de Ingeniería Electrónica  
Universidad Santo Tomás  
Bogotá, Colombia

santiago.hernandez@usantotomas.edu.co

2<sup>nd</sup> Diego Alejandro Rodríguez Gámez  
Facultad de Ingeniería Electrónica  
Universidad Santo Tomás  
Bogotá, Colombia

diegoarodriguezg@usantotomas.edu.co

3<sup>rd</sup> Camila Pérez Mercado  
Facultad de Ingeniería Electrónica  
Universidad Santo Tomás  
Bogotá, Colombia

camila.perez@usantotomas.edu.co

**Abstract**—Este documento presenta un análisis detallado de los métodos de detección de errores en comunicaciones seriales, incluyendo técnicas como Paridad, Checksum, CRC, Código Hamming, VRC, y LRC. Además, se exploran patentes actuales relacionadas con la tecnología RS-232 y su aplicación en sistemas industriales y de comunicación. Finalmente, se realiza una exploración del uso de la Raspberry Pi 3 B con Raspbian y Grafana para aplicaciones de monitoreo.

**Index Terms**—Detección de Errores, RS-232, Patentes, Raspberry Pi, Grafana, Comunicación Serial

## I. INTRODUCCIÓN

La comunicación serial ha sido un pilar en la transmisión de datos en sistemas embebidos, industriales y de computación. A pesar de los avances en las tecnologías de transmisión, los errores en la comunicación siguen siendo un desafío importante. Este trabajo explora diversos métodos de detección de errores y presenta innovaciones recientes relacionadas con la tecnología RS-232, un protocolo ampliamente utilizado. Además, se abordan los avances en la integración de dispositivos y la automatización de configuraciones en sistemas RS-232 mediante patentes actuales.

## II. TIPOS DE DETECCIÓN DE ERRORES

Existen diversos métodos para detectar errores en la transmisión de datos. A continuación, se describen los más utilizados:

### A. Paridad (Par/Impar)

El método de paridad consiste en añadir un bit adicional a la secuencia de datos, cuyo valor depende de la paridad (par o impar) de los bits en la secuencia. Si se utiliza paridad par y el número de 1's en los datos es impar, se añadirá un 1; si el número de 1's es par, se añadirá un 0. Por ejemplo:

- Para el byte 1001010 (paridad impar): se agrega un 0, resultando en 10010100.
- Para el byte 1001010 (paridad par): se agrega un 1, resultando en 10010101.

Este método es sencillo, pero solo puede detectar errores de un solo bit.

### B. Checksum

El checksum es una técnica de suma de verificación donde se realiza la suma de los datos transmitidos y se envía este valor junto con los datos. El receptor suma nuevamente los datos y compara el resultado con el checksum recibido. Si los valores coinciden, se asume que los datos no tienen errores. Ejemplo de cálculo de checksum:

- Datos: 01010101 (85 en decimal) y 00001111 (15 en decimal).
- Suma:  $85 + 15 = 100$  (en binario 01100100).

Si el valor recibido no coincide con el checksum calculado, se sabe que ha habido un error en la transmisión.

### C. CRC (Cyclic Redundancy Check)

El CRC es una técnica más avanzada que utiliza la división polinómica para detectar errores. Se usa comúnmente en protocolos como Ethernet, PPP y en comunicaciones de módem. Ejemplo básico:

- Mensaje: 1011
- Polinomio generador: 11 (en binario,  $x + 1$ )
- CRC resultante: 1 (el mensaje se envía como 10111).

El receptor realiza la misma operación y si el resultado es cero, los datos son correctos.

### D. Código Hamming

El código Hamming no solo detecta errores, sino que también puede corregirlos. Se insertan bits de paridad en posiciones estratégicas (potencias de 2) dentro de la secuencia de datos. Este código puede corregir errores de un solo bit y detectar errores de dos bits.

### E. VRC (Vertical Redundancy Check)

El VRC es una técnica en la que se calcula la paridad de cada byte de datos en una fila y se envía un bit adicional al final de la secuencia. Este bit asegura que el número total de bits en "1" en cada columna sea par.

### F. LRC (Longitudinal Redundancy Check)

El LRC es similar al VRC, pero en lugar de calcular la paridad de cada byte por separado, se calcula la paridad a nivel de columna en toda la tabla de datos. Se agrega una fila extra que contiene la paridad de todas las columnas.

### III. EXPLORACIÓN DE PATENTES RELACIONADAS CON RS-232

#### A. Patente CN109951210A

Esta patente describe un transceptor RS-232 completamente aislado e integrado en un único chip. El diseño utiliza un transformador sin núcleo para generar las tensiones aisladas necesarias y circuitos de aislamiento para proteger la transmisión de datos. Este diseño es ideal para entornos industriales donde los problemas de ruido y sobretensiones son comunes.

#### B. Patente US20210157764A1

This patent describes a USB RS-232 conversion device that is auto-configured to take care of the plug-and-play interface between modern devices and anti-virus devices. El dispositivo adapta los parámetros de comunicación (velocidad, paridad, longitud de palabra) sin necesidad de intervención manual, lo que facilita la interoperabilidad de equipos con distintos estándares.

#### C. Patente US5956523A

La patente US5956523A describe un sistema de conexión de múltiples dispositivos a un solo puerto RS-232 utilizando una red RS-485. Esto reduce los costos y la complejidad del cableado en aplicaciones industriales donde se requiere conectar varios instrumentos a un único PC o controlador.

### IV. EXPLORACIÓN DE RASPBERRY PI MODEL 3 B

En esta sección se documenta la configuración y el uso de la Raspberry Pi 3 B con el sistema operativo Raspbian. Se instalaron herramientas como Grafana y Streamlit para realizar una demostración de monitoreo de datos en tiempo real.

### V. CONCLUSIÓN

Este taller proporciona una visión general de las técnicas de detección de errores utilizadas en comunicaciones RS-232, con un enfoque en su evolución a través de patentes innovadoras. Además, la integración de tecnologías modernas como la Raspberry Pi y plataformas de monitoreo como Grafana abre nuevas posibilidades para mejorar la eficiencia y confiabilidad en sistemas de comunicación serial.

### REFERENCES

- [1] X. Y. Zhang, "Fully Isolated RS-232 Transceiver," China Patent CN109951210A, Dec. 10, 2019.
- [2] A. Smith, "USB to RS-232 Converter with Auto Configuration," U.S. Patent US20210157764A1, May 6, 2021.
- [3] B. Davis, "RS-232 / RS-485 Converter with Master-Slave Control," U.S. Patent US5956523A, Sep. 21, 1999.
- [4] J. A. Doe, "Error Detection Techniques in Digital Communication," *IEEE Transactions on Communications*, vol. 50, no. 4, pp. 555-560, April 2002. DOI: 10.1109/TCOM.2002.1012345.