CURSO COMPLETO: SISTEMAS DE COMUNICACIÓN Y REDES INALÁMBRICAS EN MECATRÓNICA

Video 01 — Parte 1A: Comunicaciones Digitales, Control Centralizado, Jerárquico, Distribuido y Redes — Bit Rate y Baud Rate, Teorema de Shannon-Hartley, Sincronismo, Pirámide CIM, DCS, Topologías y Acceso al Medio

Parte del Curso: Parte 1A — Comunicaciones Digitales, Control Centralizado, Jerárquico, Distribuido y Redes
Sistemas / Protocolos Estudiados: Capacidad de Shannon-Hartley, Límite de Nyquist, Codificación Manchester, Pirámide CIM / DCS y Acceso Determinista TDMA / Maestro-Esclavo

```
Rb = Rs·log2(M) | C_Nyquist = 2·B·log2(M) | C_Shannon = B·log2(1 + 10^(SNR_dB/10))
```

1. Síntesis de Conceptos Clave y Reglas de Ingeniería

- Distinción entre tasa de bits (Rb en bps) y tasa de símbolos o baudios (Rs en baudios) mediante Rb = Rs·log2(M), e inmunidad regenerativa de señales discretas frente a ruido acumulado.
- Límite de Nyquist para canales sin ruido y teorema de capacidad de canal de Shannon-Hartley C = B·log2(1 + S/N) en presencia de ruido blanco gaussiano aditivo.
- Modos de transmisión Simplex, Half-Duplex y Full-Duplex; codificación en banda base NRZ frente a Manchester diferencial (con transición de reloj garantizada y componente DC nula) y sincronización asíncrona (8N1) frente a síncrona.
- Comparación entre arquitectura de Control Centralizado (vulnerabilidad de punto único de falla SPOF), Control Jerárquico (Pirámide CIM ISA-95 de 5 niveles: Campo, Control, SCADA, MES y ERP) y Control Distribuido (DCS/CPS), junto con topologías (Estrella, Bus Multidrop con terminadores, Anillo Redundante y Malla) y acceso al medio determinista (Maestro-Esclavo, Token Passing y TDMA).

2. Parámetros Nominales de Diseño de Red y Enlace RF (Video 01)

Parámetro de Comunicación / Radiofrecuencia	Símbolo	Valor Nominal	Función en la Red Industrial / Enlace Inalámbrico
Velocidad de Transmisión en Serie	`Baud Rate / Rb`	500000 bps	Caudal de transferencia en bus físico UART / RS-485 / CAN
Ancho de Banda de Canal	`B [kHz]`	250 kHz	Determina la capacidad de Shannon-Hartley y tasa de símbolos
Relación Señal a Ruido	`SNR [dB]`	24 dB	Margen sobre el piso de ruido térmico e interferencia industrial
Frecuencia Portadora ISM y Potencia TX	`f [MHz] / Pt [dBm]`	868 MHz / +14 dBm	Cálculo de longitud de onda λ, antena λ/4 y ecuación de Friis
Factor de Ensanchamiento LoRa y Alcance	`SF / d [m]`	SF8 / 300 m	Compromiso entre sensibilidad de recepción y Tiempo en el Aire
Watchdog de Comunicación Determinista	`T_timeout [ms]`	20 ms	Ventana máxima antes de conmutar actuadores a estado Fail-Safe

3. Desarrollo Académico Diapositiva por Diapositiva

Diapositiva 01: Fundamentos de Comunicaciones Digitales en Mecatrónica

Inmunidad Regenerativa de Señales Discretas, Tasa de Bits (Bit Rate) frente a Tasa de Símbolos (Baud Rate)

- **Turno 1 (Masculina):** En un entorno industrial severo lleno de variadores de frecuencia, motores y contactores, las señales analógicas sufren atenuación y acumulan ruido electromagnético irreversible a lo largo del cable.
- **Turno 2 (Femenina):** Las comunicaciones digitales resuelven este problema transmitiendo estados discretos que el receptor regenera limpiamente mediante umbrales lógicos, siempre que el ruido no supere el margen de decisión.
- **Turno 3 (Masculina):** En ingeniería debemos distinguir con precisión entre la velocidad de transferencia de bits, o Bit Rate erre sub be medida en bits por segundo, y la velocidad de modulación, o Baud Rate erre sub ese medida en baudios o símbolos por segundo.
- **Turno 4 (Femenina):** Ambas magnitudes se relacionan mediante la ecuación erre sub be igual a erre sub ese por el logaritmo en base dos de eme: si cada símbolo codifica cuatro niveles de tensión o fase, cada baudio transporta exactamente dos bits.

Diapositiva 02: Capacidad de Canal de Shannon-Hartley, Límite de Nyquist y SNR

C_Nyquist = 2·B·log2(M), Teorema C = B·log2(1 + S/N) y Relación Señal a Ruido en Decibelios

- **Turno 1 (Masculina):** ¿Cuál es la velocidad máxima teórica a la que podemos enviar datos por un cable o enlace de radio sin que los pulsos se solapen? Harry Nyquist demostró que en un canal ideal de ancho de banda be hercios la tasa máxima de muestreo es dos be símbolos por segundo.
- **Turno 2 (Femenina):** Sin embargo, todo canal físico real presenta ruido térmico e interferencia, cuya intensidad se cuantifica mediante la relación señal a ruido ese ene erre, expresada en decibelios como diez por el logaritmo en base diez de la potencia de señal sobre la potencia de ruido.
- **Turno 3 (Masculina):** Claude Shannon completó la teoría con la célebre ecuación de Shannon-Hartley: la capacidad máxima libre de errores ce es igual al ancho de banda be multiplicado por el logaritmo en base dos de uno más la relación lineal señal a ruido.
- **Turno 4 (Femenina):** Esta ley fundamental revela al ingeniero mecatrónico que si el ruido en la planta aumenta y degrada la relación señal a ruido, estamos obligados a reducir la tasa de bits o ampliar el ancho de banda para mantener una comunicación confiable.

Diapositiva 03: Modos de Transmisión y Codificación en Banda Base y Banda Ancha

Simplex, Half-Duplex, Full-Duplex, Códigos de Línea NRZ vs. Manchester Diferencial y Modulación

- **Turno 1 (Masculina):** Según la direccionalidad del flujo de información, clasificamos todo enlace mecatrónico en tres modos operativos: Simplex cuando es unidireccional, Half-Duplex cuando es bidireccional alternado por turnos y Full-Duplex cuando transmite y recibe simultáneamente.
- **Turno 2 (Femenina):** Un sensor óptico básico que solo envía pulsos opera en modo Simplex, un bus industrial erre ese cuatro ochenta y cinco sobre un único par trenzado opera en Half-Duplex, y una interfaz Ethernet o ese pe i con líneas separadas opera en Full-Duplex.
- **Turno 3 (Masculina):** Cuando enviamos pulsos digitales directamente sobre el cable hablamos de transmisión en Banda Base: el código más simple es ene erre zeta, o no retorno a cero, pero una larga cadena de unos o ceros idénticos provoca que el receptor pierda la sincronización del reloj.
- **Turno 4 (Femenina):** Por ello muchos buses industriales emplean codificación Manchester o Manchester diferencial, que fuerza una transición de voltaje justo en la mitad de cada periodo de bit, eliminando la componente de corriente continua y permitiendo extraer el reloj directamente de los datos.

Diapositiva 04: Sincronización de Datos: Transmisión Asíncrona frente a Síncrona

Tramas de Carácter con Start/Stop Bits vs. Tramas de Bloque con Reloj Compartido o Recuperado

- **Turno 1 (Masculina):** Para que el receptor muestree cada bit en el instante exacto de máxima apertura del diagrama de ojo, existen dos paradigmas de sincronización temporal: la transmisión asíncrona y la transmisión síncrona.
- **Turno 2 (Femenina):** En la transmisión asíncrona no existe línea de reloj entre los equipos; cada byte se encapsula de forma independiente añadiendo un bit de inicio en nivel bajo, cinco a nueve bits de datos, un bit opcional de paridad y uno o dos bits de parada en nivel alto.
- **Turno 3 (Masculina):** Al detectar el flanco de bajada del bit de inicio, el oscilador local del receptor reinicia su cuenta y muestrea el centro de cada bit, lo cual exige que los relojes de ambos extremos no difieran en más de un dos punto cinco por ciento, con una sobrecarga del veinte por ciento.
- **Turno 4 (Femenina):** En cambio, la transmisión síncrona envía bloques continuos de cientos de bytes acompañados de una señal de reloj dedicada o codificada en la portadora, superando el noventa y cinco por ciento de eficiencia útil a velocidades de megabits o gigabits por segundo.

Diapositiva 05: Arquitectura de Control Centralizado en Sistemas Mecatrónicos

Concentración de Cómputo, Cableado Radial Punto a Punto y Vulnerabilidad de Punto Único de Falla

- **Turno 1 (Masculina):** Analicemos ahora cómo se organizan arquitectónicamente los controladores dentro de una máquina o planta industrial, comenzando por el esquema histórico más directo: la arquitectura de control centralizado.
- **Turno 2 (Femenina):** En el control centralizado, una única unidad central de procesamiento o P L C maestro de gran capacidad lee directamente todos los sensores del sistema, ejecuta la totalidad de los lazos de control y comanda cada actuador mediante cables dedicados.
- **Turno 3 (Masculina):** Su ventaja principal radica en que el algoritmo global tiene acceso instantáneo a todas las variables de estado en una misma memoria, facilitando el diseño de máquinas pequeñas o bancos de prueba compactos con pocos ejes.
- **Turno 4 (Femenina):** Sin embargo, al crecer la planta aparecen tres desventajas críticas: kilómetros de mazos de cables analógicos costosos y pesados, difícil mantenimiento y un peligroso punto único de falla, pues si el procesador central se detiene, toda la fábrica colapsa.

Diapositiva 06: Arquitectura de Control Jerárquico y la Pirámide CIM / ISA-95

Los 5 Niveles de Automatización: Campo (Nivel 0), Control (1), Supervisión SCADA (2), MES (3) y ERP (4)

- **Turno 1 (Masculina):** Para ordenar la complejidad de una instalación moderna, la arquitectura de control jerárquico divide el sistema en estratos especializados según la pirámide de manufactura integrada por computador ce i eme y la norma internacional ISA noventa y cinco.
- **Turno 2 (Femenina):** En la base se ubica el Nivel cero de Campo con sensores y actuadores físicos que operan en microsegundos o milisegundos, gobernados directamente por el Nivel uno de Control donde residen los P L C, microcontroladores y servodrives de tiempo real duro.
- **Turno 3 (Masculina):** Por encima se sitúa el Nivel dos de Supervisión y Adquisición de Datos SCADA e interfaces humano-máquina hache eme i, que consolidan alarmas, recetas y tendencias históricas con tiempos de refresco de décimas de segundo a segundos.
- **Turno 4 (Femenina):** En la cúspide operan el Nivel tres de sistemas de ejecución de manufactura eme e ese, que gestiona la trazabilidad, calidad y mantenimiento en minutos, y el Nivel cuatro de planificación de recursos empresariales e erre pe, donde se programan órdenes e inventarios en horas o días.

Diapositiva 07: Arquitectura de Control Distribuido (DCS) y Sistemas Ciberfísicos

Descentralización Inteligente, Lazos Locales Autónomos, Tolerancia a Fallos y Modularidad Plug-and-Produce

- **Turno 1 (Masculina):** La evolución moderna de la mecatrónica reemplaza el mazo de cables radiales por la arquitectura de Control Distribuido, conocida como de ce ese, donde la inteligencia se reparte físicamente junto a cada articulación, motor o estación de proceso.
- **Turno 2 (Femenina):** En un robot industrial o vehículo autónomo distribuido, cada eje posee su propio microcontrolador local que cierra el lazo de corriente, velocidad y posición a diez kilohercios, comunicándose con el coordinador únicamente mediante consignas digitales por un bus compartido.
- **Turno 3 (Masculina):** Esta descentralización reduce hasta en un ochenta por ciento el peso y costo del cableado, elimina la caída de tensión analógica y otorga tolerancia a fallos: si un nodo secundario entra en alerta, los demás controladores locales ejecutan una parada controlada y segura.
- **Turno 4 (Femenina):** Al dotar a cada módulo mecatrónico de capacidad de autodiagnóstico y comunicación en red, damos paso a los Sistemas Ciberfísicos de la Industria cuatro punto cero, donde las estaciones se acoplan bajo el paradigma conectar y producir.

Diapositiva 08: Redes Industriales y Topologías Físicas y Lógicas

Comparativa de Ingeniería entre Topología en Estrella, Bus Lineal Multidrop, Anillo Redundante y Malla

- **Turno 1 (Masculina):** La disposición geométrica y eléctrica con la que interconectamos los nodos de una red mecatrónica define su topología, la cual impacta directamente en el costo de instalación, la latencia y la supervivencia ante roturas de cable.
- **Turno 2 (Femenina):** En la topología en Estrella todos los nodos se conectan a un conmutador o nodo maestro central: si un cable individual se corta solo se pierde ese equipo, pero el sistema depende por completo de la salud del conmutador central.
- **Turno 3 (Masculina):** En la topología de Bus Lineal o Multidrop, típica de erre ese cuatro ochenta y cinco y bus CAN, todos los dispositivos comparten un único par troncal con derivaciones cortas y resistencias terminadoras en ambos extremos que absorben las reflexiones de onda.
- **Turno 4 (Femenina):** Para procesos críticos empleamos la topología en Anillo Redundante, donde el último esclavo se reconecta al maestro permitiendo que los paquetes viajen en sentido opuesto en menos de un milisegundo si se rompe el cable, o la topología en Malla en redes inalámbricas.

Diapositiva 09: Métodos Deterministas y Probabilísticos de Acceso al Medio (MAC)

Sondeo Maestro-Esclavo (Polling), Paso de Testigo (Token Passing), TDMA Sincronizado y CSMA/CD con Arbitraje

- **Turno 1 (Masculina):** Cuando múltiples controladores comparten un mismo cable o canal de radio, ¿quién tiene derecho a transmitir en cada milisegundo sin provocar una colisión destructiva? Esa es la misión del protocolo de control de acceso al medio, o subcapa MAC.
- **Turno 2 (Femenina):** Las redes de oficina clásicas utilizaban acceso probabilístico ce ese eme a con detección de colisiones: cada nodo escucha antes de hablar y, si dos transmiten a la vez, ambos abortan y esperan un tiempo aleatorio, lo cual destruye el determinismo temporal.
- **Turno 3 (Masculina):** Para garantizar un tiempo máximo de respuesta estricto, los buses industriales emplean métodos deterministas: en el esquema Maestro-Esclavo, un único nodo maestro interroga secuencialmente a cada esclavo mediante sondeo y ningún esclavo habla sin permiso.
- **Turno 4 (Femenina):** Otros dos métodos deterministas clave son el Paso de Testigo o Token Passing, donde una trama especial circula otorgando el turno de transmisión por un tiempo acotado, y el Acceso Múltiple por División de Tiempo te de eme a, que asigna ranuras exactas de microsegundos a cada nodo.

Diapositiva 10: Síntesis de la Parte 1A y Transición al Siguiente Módulo

De la Capacidad de Shannon y las Arquitecturas Distribuidas al Modelo OSI y los Buses Serie/Paralelo

- **Turno 1 (Masculina):** En esta primera parte hemos establecido los cimientos matemáticos y arquitectónicos de las comunicaciones mecatrónicas: desde la relación entre tasa de bits y tasa de baudios hasta la capacidad de canal de Shannon-Hartley frente al ruido.
- **Turno 2 (Femenina):** Comparamos los modos Simplex, Half-Duplex y Full-Duplex, la codificación Manchester y la sincronización asíncrona y síncrona que gobiernan la integridad física de cada bit transmitido en planta.
- **Turno 3 (Masculina):** Asimismo, contrastamos las arquitecturas de control centralizado, jerárquico en cinco niveles y distribuido, junto con las topologías en bus, estrella, anillo redundante y malla bajo esquemas deterministas de acceso al medio.
- **Turno 4 (Femenina):** Con esta base firme, acompáñennos en el siguiente módulo, correspondiente a la Parte uno B, donde desglosaremos el modelo OSI de siete capas, los protocolos industriales y las interfases de comunicación en serie y en paralelo.

4. Firmware C++/MCU: Trama Industrial, CRC-16 y FSM Fail-Safe (comunicacion_embebida_video_01.ino)

```
/*
 * ============================================================================
 * CURSO COMPLETO: SISTEMAS DE COMUNICACIÓN Y REDES INALÁMBRICAS EN MECATRÓNICA
 * VIDEO 01: Video 01 — Parte 1A: Comunicaciones Digitales, Control Centralizado, Jerárquico, Distribuido y Redes
 * Archivo: comunicacion_embebida_video_01.ino
 * ----------------------------------------------------------------------------
 * Descripción Técnica:
 *   Firmware determinista en C/C++ Embebido para Microcontrolador (ESP32-S3 /
 *   STM32 / Arduino Industrial) que implementa:
 *     1. Empaquetado y análisis sintáctico de tramas binarias industriales
 *        compatibles con RS-485 (Modbus RTU) y pasarelas inalámbricas (BLE/LoRa).
 *     2. Cálculo polinómico de Verificación de Redundancia Cíclica CRC-16
 *        (polinomio reflectado 0xA001) para inmunidad ante ruido electromagnético.
 *     3. Control de dirección Half-Duplex (pin DE/RE de transceptor MAX485) y
 *        lectura de telemetría de radiofrecuencia (RSSI / SNR).
 *     4. Máquina de Estados Finitos (FSM) no bloqueante con supervisión de
 *        Timeout (20 ms) y estado seguro (Fail-Safe) ante pérdida de enlace.
 *
 * Enfoque del Módulo:
 *   Capacidad de Shannon-Hartley, Límite de Nyquist, Codificación Manchester, Pirámide CIM / DCS y Acceso
Determinista TDMA / Maestro-Esclavo
 * ============================================================================
 */

#include <Arduino.h>
#include <stdint.h>

// --- Definición de Pines de Interfase de Comunicación y Estado Seguro ---
const uint8_t PIN_RS485_DE_RE   = 4;  // Control de dirección Half-Duplex (HIGH = TX, LOW = RX)
const uint8_t PIN_CS_SPI_RADIO  = 10; // Chip Select (CS#) para transceptor SPI (LoRa SX1262 / CAN)
const uint8_t PIN_LED_ENLACE_OK = 2;  // Indicador de trama válida recibida
const uint8_t PIN_DO_FAILSAFE   = 8;  // Salida de actuador con corte seguro por Timeout

// --- Parámetros Nominales de Comunicación (Video 01) ---
const uint32_t BAUD_RATE_SERIE   = 500000UL;
const uint8_t  NODO_DIRECCION_ID = 0x01;       // Dirección de esclavo / nodo mecatrónico
const uint32_t TIMEOUT_ENLACE_MS = 20UL;      // Tiempo máximo sin trama antes de Fail-Safe
const float    FREQ_PORTADORA_MHZ = 868.0f;
const int8_t   TX_POWER_DBM       = 14;
const uint8_t  LORA_SPREADING_SF  = 8;

// --- Estructura de Trama Industrial Unificada (12 Bytes con CRC-16) ---
// [0]: SOF (0xAA) | [1]: Nodo ID | [2]: Función | [3..4]: Setpoint (uint16)
// [5..6]: Variable Proceso PV (uint16) | [7]: RSSI_dBm (int8) | [8]: EstadoFlags
// [9]: Secuencia | [10..11]: CRC-16-Modbus (Little-Endian)
struct TramaMecatronica {
  uint8_t  sof;
  uint8_t  nodoId;
  uint8_t  funcion;
  uint16_t consignaSp;
  uint16_t variablePv;
  int8_t   rssiDbm;
  uint8_t  estadoFlags;
  uint8_t  secuencia;
  uint16_t crc16;
};

enum EstadoEnlaceFSM : uint8_t {
  ENLACE_ESPERANDO_SOF = 0,
  ENLACE_RECIBIENDO_BYTES,
  ENLACE_VALIDANDO_CRC,
  ENLACE_EJECUTANDO_RESPUESTA,
  ENLACE_ERROR_FAILSAFE
};

EstadoEnlaceFSM g_estadoEnlace = ENLACE_ESPERANDO_SOF;
uint8_t         g_bufferRx[12];
```

```
uint8_t         g_indiceRx = 0;
uint32_t        g_ultimoPaqueteValidoMs = 0;
uint8_t         g_contadorSecuenciaTx = 0;
uint16_t        g_consignaActiva = 0;

// --- Cálculo Polinómico CRC-16-Modbus (Polinomio 0xA001) ---
uint16_t calcularCrc16Modbus(const uint8_t *datos, uint8_t longitud) {
  uint16_t crc = 0xFFFFU;
  for (uint8_t i = 0; i < longitud; i++) {
    crc ^= static_cast<uint16_t>(datos[i]);
    for (uint8_t bit = 0; bit < 8; bit++) {
      if ((crc & 0x0001U) != 0U) {
        crc = (crc >> 1U) ^ 0xA001U;
      } else {
        crc >>= 1U;
      }
    }
  }
  return crc;
}

// --- Transmisión Half-Duplex sobre Par Diferencial RS-485 / Enlace Serie ---
void transmitirTramaRespuesta(uint8_t funcion, uint16_t pvActual, int8_t rssiEstimado) {
  uint8_t tramaTx[12];
  tramaTx[0] = 0xAAU;
  tramaTx[1] = NODO_DIRECCION_ID;
  tramaTx[2] = funcion;
  tramaTx[3] = static_cast<uint8_t>(g_consignaActiva & 0xFFU);
  tramaTx[4] = static_cast<uint8_t>((g_consignaActiva >> 8U) & 0xFFU);
  tramaTx[5] = static_cast<uint8_t>(pvActual & 0xFFU);
  tramaTx[6] = static_cast<uint8_t>((pvActual >> 8U) & 0xFFU);
  tramaTx[7] = static_cast<uint8_t>(rssiEstimado);
  tramaTx[8] = (g_estadoEnlace == ENLACE_ERROR_FAILSAFE) ? 0x80U : 0x01U;
  tramaTx[9] = g_contadorSecuenciaTx++;

  uint16_t crc = calcularCrc16Modbus(tramaTx, 10);
  tramaTx[10] = static_cast<uint8_t>(crc & 0xFFU);
  tramaTx[11] = static_cast<uint8_t>((crc >> 8U) & 0xFFU);

  // Conmutar transceptor diferencial a modo Transmisión (DE/RE = HIGH)
  digitalWrite(PIN_RS485_DE_RE, HIGH);
  Serial.write(tramaTx, 12);
  Serial.flush(); // Esperar vaciado completo del registro de desplazamiento UART
  digitalWrite(PIN_RS485_DE_RE, LOW);
}

void setup() {
  pinMode(PIN_RS485_DE_RE, OUTPUT);
  pinMode(PIN_CS_SPI_RADIO, OUTPUT);
  pinMode(PIN_LED_ENLACE_OK, OUTPUT);
  pinMode(PIN_DO_FAILSAFE, OUTPUT);

  digitalWrite(PIN_RS485_DE_RE, LOW);   // Modo escucha por defecto en bus Half-Duplex
  digitalWrite(PIN_CS_SPI_RADIO, HIGH); // Línea CS# inactiva en nivel alto
  digitalWrite(PIN_DO_FAILSAFE, LOW);

  Serial.begin(BAUD_RATE_SERIE);
  g_ultimoPaqueteValidoMs = millis();
}

void loop() {
  uint32_t ahoraMs = millis();

  // 1. Procesamiento No Bloqueante de Bytes Entrantes en la FSM de Protocolo
  while (Serial.available() > 0) {
    uint8_t byteLeido = static_cast<uint8_t>(Serial.read());

    if (g_estadoEnlace == ENLACE_ESPERANDO_SOF || g_estadoEnlace == ENLACE_ERROR_FAILSAFE) {
```

```
      if (byteLeido == 0xAAU) {
        g_bufferRx[0] = byteLeido;
        g_indiceRx = 1;
        g_estadoEnlace = ENLACE_RECIBIENDO_BYTES;
      }
    } else if (g_estadoEnlace == ENLACE_RECIBIENDO_BYTES) {
      g_bufferRx[g_indiceRx++] = byteLeido;
      if (g_indiceRx >= 12U) {
        g_estadoEnlace = ENLACE_VALIDANDO_CRC;
      }
    }

    if (g_estadoEnlace == ENLACE_VALIDANDO_CRC) {
      uint16_t crcRecibido = static_cast<uint16_t>(g_bufferRx[10]) |
                             (static_cast<uint16_t>(g_bufferRx[11]) << 8U);
      uint16_t crcCalculado = calcularCrc16Modbus(g_bufferRx, 10);

      if (crcRecibido == crcCalculado && g_bufferRx[1] == NODO_DIRECCION_ID) {
        g_consignaActiva = static_cast<uint16_t>(g_bufferRx[3]) |
                           (static_cast<uint16_t>(g_bufferRx[4]) << 8U);
        g_ultimoPaqueteValidoMs = ahoraMs;
        digitalWrite(PIN_LED_ENLACE_OK, HIGH);
        digitalWrite(PIN_DO_FAILSAFE, HIGH);

        uint16_t pvSensorLocal = static_cast<uint16_t>(analogRead(A0));
        int8_t rssiMedido = -68; // Ejemplo de nivel RSSI en dBm (-68 dBm)
        transmitirTramaRespuesta(g_bufferRx[2], pvSensorLocal, rssiMedido);
      }
      g_indiceRx = 0;
      g_estadoEnlace = ENLACE_ESPERANDO_SOF;
    }
  }

  // 2. Supervisión Determinista de Watchdog de Comunicación (Timeout Fail-Safe)
  if ((ahoraMs - g_ultimoPaqueteValidoMs) > TIMEOUT_ENLACE_MS) {
    g_estadoEnlace = ENLACE_ERROR_FAILSAFE;
    g_consignaActiva = 0U;
    digitalWrite(PIN_LED_ENLACE_OK, LOW);
    digitalWrite(PIN_DO_FAILSAFE, LOW); // Desenergiza actuador por pérdida de red
  }
}
```

5. Script Python SBC: Shannon-Hartley, Presupuesto de Enlace Friis y LoRa CSS (analisis_redes_y_enlace_rf_video_01.py)

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
================================================================================
CURSO COMPLETO: SISTEMAS DE COMUNICACIÓN Y REDES INALÁMBRICAS EN MECATRÓNICA
VIDEO 01: Video 01 — Parte 1A: Comunicaciones Digitales, Control Centralizado, Jerárquico, Distribuido y Redes
Archivo: analisis_redes_y_enlace_rf_video_01.py
--------------------------------------------------------------------------------
Descripción Técnica:
  Herramienta de ingeniería en Python para Computadora de Placa Única (SBC /
  Raspberry Pi / Gateway IIoT) que ejecuta:
    1. Cálculo de Capacidad de Canal de Shannon-Hartley y Límite de Nyquist.
    2. Cálculo del Presupuesto de Enlace de Radiofrecuencia (Link Budget),
       Pérdida en el Espacio Libre (Ecuación de Friis / FSPL) y Margen de
       Desvanecimiento (Fade Margin) en bandas ISM (868 MHz).
    3. Cálculo de Tiempo en el Aire (Time-on-Air) y tasa de bits de la
       modulación Chirp Spread Spectrum (CSS) de LoRa (SF8).
    4. Empaquetado y verificación de tramas Modbus RTU con CRC-16 y payload
       JSON para telemetría MQTT / OPC UA.

Enfoque del Módulo:
  Capacidad de Shannon-Hartley, Límite de Nyquist, Codificación Manchester, Pirámide CIM / DCS y Acceso
Determinista TDMA / Maestro-Esclavo
================================================================================
"""

import math
import struct
import json
from typing import Dict, Any

# --- Parámetros Nominales del Enlace Mecatrónico (Video 01) ---
BW_CANAL_KHZ      = 250.0
SNR_DB_NOMINAL    = 24.0
FREQ_ISM_MHZ      = 868.0
TX_POWER_DBM      = 14.0
DISTANCIA_METROS  = 300.0
LORA_SF_NOMINAL   = 8


def calcular_capacidad_shannon_nyquist(bw_khz: float, snr_db: float, niveles_m: int = 4) -> Dict[str, float]:
    """Calcula la capacidad máxima de Shannon-Hartley y la tasa de Nyquist."""
    bw_hz = bw_khz * 1000.0
    snr_lineal = 10.0 ** (snr_db / 10.0)
    capacidad_shannon_bps = bw_hz * math.log2(1.0 + snr_lineal)
    capacidad_nyquist_bps = 2.0 * bw_hz * math.log2(niveles_m)
    return {
        "Ancho_Banda_kHz": round(bw_khz, 2),
        "SNR_dB": round(snr_db, 2),
        "SNR_Lineal": round(snr_lineal, 4),
        "Capacidad_Nyquist_kbps": round(capacidad_nyquist_bps / 1000.0, 3),
        "Capacidad_Shannon_kbps": round(capacidad_shannon_bps / 1000.0, 3),
    }


def calcular_presupuesto_enlace_friis(
    pt_dbm: float,
    freq_mhz: float,
    distancia_m: float,
    gt_dbi: float = 2.15,
    gr_dbi: float = 2.15,
    perdidas_obstaculos_db: float = 12.0,
    sensibilidad_rx_dbm: float = -118.0,
) -> Dict[str, Any]:
    """Evalúa la ecuación de Friis, FSPL y el Margen de Desvanecimiento (Fade Margin)."""
    distancia_km = max(distancia_m / 1000.0, 0.001)
    longitud_onda_m = 299.792458 / freq_mhz
```

```
    antena_cuarto_onda_cm = (longitud_onda_m / 4.0) * 100.0

    # FSPL(dB) = 20*log10(d_km) + 20*log10(f_MHz) + 32.44
    fspl_db = 20.0 * math.log10(distancia_km) + 20.0 * math.log10(freq_mhz) + 32.44
    pr_dbm = pt_dbm + gt_dbi + gr_dbi - fspl_db - perdidas_obstaculos_db
    fade_margin_db = pr_dbm - sensibilidad_rx_dbm

    return {
        "Frecuencia_MHz": freq_mhz,
        "Longitud_Antena_Cuarto_Onda_cm": round(antena_cuarto_onda_cm, 2),
        "FSPL_Espacio_Libre_dB": round(fspl_db, 2),
        "Potencia_Recibida_RSSI_dBm": round(pr_dbm, 2),
        "Margen_Desvanecimiento_dB": round(fade_margin_db, 2),
        "Enlace_Industrial_Viable": fade_margin_db >= 15.0,
    }


def calcular_parametros_fisicos_lora(sf: int, bw_khz: float = 125.0, cr_denom: int = 5, payload_bytes: int = 16) -
> Dict[str, float]:
    """Calcula velocidad de símbolo, velocidad de bits y Time-on-Air para LoRa CSS."""
    bw_hz = bw_khz * 1000.0
    chips_por_simbolo = 2 ** sf
    tasa_simbolos_rs = bw_hz / chips_por_simbolo
    t_simbolo_ms = (1.0 / tasa_simbolos_rs) * 1000.0
    tasa_bits_bps = sf * tasa_simbolos_rs * (4.0 / cr_denom)

    # Estimación estándar de símbolos de trama y Time-on-Air
    de_low_rate = 1 if (sf >= 11 and bw_khz <= 125.0) else 0
    num = 8 * payload_bytes - 4 * sf + 28 + 16
    den = 4 * (sf - 2 * de_low_rate)
    n_payload = 8 + max(math.ceil(num / den) * cr_denom, 0)
    tiempo_aire_ms = (12.25 + n_payload) * t_simbolo_ms

    return {
        "Spreading_Factor": sf,
        "Chips_por_Simbolo": chips_por_simbolo,
        "Tasa_Bits_Efectiva_bps": round(tasa_bits_bps, 2),
        "Tiempo_en_el_Aire_ms": round(tiempo_aire_ms, 2),
    }


def calcular_crc16_modbus(data: bytes) -> int:
    """Calcula el CRC-16-Modbus (polinomio reflectado 0xA001)."""
    crc = 0xFFFF
    for b in data:
        crc ^= b
        for _ in range(8):
            if crc & 0x0001:
                crc = (crc >> 1) ^ 0xA001
            else:
                crc >>= 1
    return crc & 0xFFFF


def construir_trama_modbus_rtu(slave_id: int, funcion: int, reg_inicio: int, valor_reg: int) -> bytes:
    """Construye una trama Modbus RTU completa con verificación CRC-16."""
    cabecera = struct.pack(">BBHH", slave_id, funcion, reg_inicio, valor_reg)
    crc = calcular_crc16_modbus(cabecera)
    return cabecera + struct.pack("<H", crc)


if __name__ == "__main__":
    print("=" * 78)
    print(" CURSO: SISTEMAS DE COMUNICACIÓN Y REDES INALÁMBRICAS — VIDEO 01")
    print(" Enfoque:", "Capacidad de Shannon-Hartley, Límite de Nyquist, Codificación Manchester, Pirámide CIM /
DCS y Acceso Determinista TDMA / Maestro-Esclavo")
    print("=" * 78)
```

```
    cap = calcular_capacidad_shannon_nyquist(BW_CANAL_KHZ, SNR_DB_NOMINAL)
    print("\n[1] Capacidad de Canal (Nyquist y Shannon-Hartley):", json.dumps(cap, ensure_ascii=False))

    rf = calcular_presupuesto_enlace_friis(TX_POWER_DBM, FREQ_ISM_MHZ, DISTANCIA_METROS)
    print("\n[2] Presupuesto de Enlace RF (Ecuación de Friis):", json.dumps(rf, ensure_ascii=False))

    lora_cfg = calcular_parametros_fisicos_lora(LORA_SF_NOMINAL, BW_CANAL_KHZ)
    print("\n[3] Parámetros Físicos LoRa CSS:", json.dumps(lora_cfg, ensure_ascii=False))

    trama_rtu = construir_trama_modbus_rtu(slave_id=1, funcion=0x06, reg_inicio=0x0100, valor_reg=1500)
    print("\n[4] Trama Modbus RTU con CRC-16:", trama_rtu.hex(" ").upper())
```

6. Programa PLC IEC 61131-3 (ST + IL): Pasarela de Red, Heartbeat y Watchdog (pasarela_industrial_plc_video_01.st)

```
(* ============================================================================
 * CURSO COMPLETO: SISTEMAS DE COMUNICACIÓN Y REDES INALÁMBRICAS EN MECATRÓNICA
 * VIDEO 01: Video 01 — Parte 1A: Comunicaciones Digitales, Control Centralizado, Jerárquico, Distribuido y Redes
 * Archivo: pasarela_industrial_plc_video_01.st
 * ----------------------------------------------------------------------------
 * Estándar: IEC 61131-3 — Texto Estructurado (ST) + Equivalencia en Lista de
 *           Instrucciones (IL) para Pasarela de Comunicaciones Industriales
 * Descripción Técnica:
 *   Bloque de Función Industrial (FB_GestorRedMecatronica_V01) que
 *   supervisa el intercambio cíclico de datos entre el PLC maestro, buses de
 *   campo serie/Ethernet (Modbus RTU/TCP, CANopen, PROFINET) y nodos sensores
 *   inalámbricos (Wi-Fi MQTT, BLE, ZigBee, LoRaWAN, WirelessHART), validando:
 *     1. Contador de latido (Heartbeat) y secuencia de trama para detectar
 *        congelamiento o pérdida del enlace.
 *     2. Umbral de potencia recibida RSSI y relación señal a ruido SNR.
 *     3. Temporizador Watchdog de red (TON) de 20 ms con activación
 *        automática de estado seguro (Fail-Safe) en los actuadores de planta.
 * ============================================================================ *)

FUNCTION_BLOCK FB_GestorRedMecatronica_V01
VAR_INPUT
    (* Señales de Diagnóstico de Capa de Enlace y Transporte *)
    xTramaValidaCrcOk     : BOOL; (* 1 = Verificación polinómica CRC-16 correcta *)
    xParoEmergenciaRed_NC : BOOL; (* 1 = Canal de seguridad PROFIsafe / E-Stop OK *)
    byContadorHeartbeatRx : BYTE; (* Contador incremental enviado por el nodo remoto *)
    iRssiMedidoDbm        : INT;  (* Nivel de potencia recibida en dBm, ej. -72 *)
    iSnrMedidoDb          : INT;  (* Relación Señal a Ruido en dB, ej. 24 *)

    (* Datos de Proceso Recibidos en el Payload de la Trama *)
    iVariableProcesoRemota : INT; (* Valor crudo 0..27648 desde nodo remoto *)
    rConsignaLocalMaestro  : REAL := 75.0; (* Consigna a transmitir hacia actuador remoto *)
END_VAR

VAR_OUTPUT
    (* Banderas de Salud del Enlace y Conmutación Fail-Safe *)
    xEnlaceOperativoOk    : BOOL; (* 1 = Comunicación determinista activa *)
    xAlertaCalidadRfBaja  : BOOL; (* 1 = Margen de desvanecimiento reducido *)
    xFalloTimeoutFailSafe : BOOL; (* 1 = Pérdida de comunicación -> Parada segura *)

    (* Variables de Proceso Validadas hacia el Lazo de Control / SCADA *)
    rVariableProcesoSegura : REAL; (* Valor escalado o valor seguro por defecto *)
    iPalabraMandoTx        : INT;  (* Registro empaquetado para PDO / Modbus 0x06 *)
    byHeartbeatTx          : BYTE; (* Latido transmitido por el PLC maestro *)
END_VAR

VAR
    tonWatchdogRed         : TON;
    byHeartbeatPrevio      : BYTE := 16#00;
    xLatidoActualizado     : BOOL := FALSE;
    iUmbralMinimoRssiDbm   : INT  := -98;
END_VAR

(* ============================================================================
 * 1. DETECCIÓN DE ACTUALIZACIÓN DE LATIDO (HEARTBEAT) Y INTEGRIDAD CRC-16
 *    Equivalencia en Lista de Instrucciones (IL):
 *      LD     byContadorHeartbeatRx
 *      NE     byHeartbeatPrevio
 *      AND    xTramaValidaCrcOk
 *      ST     xLatidoActualizado
 * ============================================================================ *)
xLatidoActualizado := (byContadorHeartbeatRx <> byHeartbeatPrevio) AND xTramaValidaCrcOk;

IF xLatidoActualizado THEN
    byHeartbeatPrevio := byContadorHeartbeatRx;
END_IF;

(* ============================================================================
```

```
 * 2. TEMPORIZADOR WATCHDOG DE COMUNICACIÓN (TON DE 20 MS)
 *    Si no llegan tramas válidas dentro de la ventana determinista, activa Fail-Safe
 * ========================================================================== *)
tonWatchdogRed(
    IN := NOT xLatidoActualizado,
    PT := T#20MS
);

xFalloTimeoutFailSafe := tonWatchdogRed.Q OR (NOT xParoEmergenciaRed_NC);
xEnlaceOperativoOk    := (NOT xFalloTimeoutFailSafe) AND xTramaValidaCrcOk;

(* ============================================================================
 * 3. SUPERVISIÓN DEL PRESUPUESTO DE ENLACE INALÁMBRICO (RSSI Y SNR)
 * ========================================================================== *)
IF (iRssiMedidoDbm < iUmbralMinimoRssiDbm) OR (iSnrMedidoDb < -15) THEN
    xAlertaCalidadRfBaja := TRUE;
ELSE
    xAlertaCalidadRfBaja := FALSE;
END_IF;

(* ============================================================================
 * 4. GESTIÓN DETERMINISTA DE DATOS Y CONMUTACIÓN A ESTADO SEGURO (FAIL-SAFE)
 * ========================================================================== *)
IF xEnlaceOperativoOk THEN
    (* Escala la variable remota (0..27648 -> 0.0..100.0 %) y empaqueta consigna *)
    rVariableProcesoSegura := (INT_TO_REAL(iVariableProcesoRemota) / 27648.0) * 100.0;
    iPalabraMandoTx        := REAL_TO_INT((rConsignaLocalMaestro / 100.0) * 27648.0);
    byHeartbeatTx          := byHeartbeatTx + 16#01;
ELSE
    (* Estado Seguro (Fail-Safe): Fuerza consigna de potencia a cero *)
    rVariableProcesoSegura := 0.0;
    iPalabraMandoTx        := 0;
END_IF;

END_FUNCTION_BLOCK
```