CURSO COMPLETO: SISTEMAS DE COMUNICACIÓN Y REDES INALÁMBRICAS EN MECATRÓNICA

Video 03 — Parte 2A: Estado del Arte Inalámbrico, Propagación RF, Wi-Fi y Bluetooth / BLE — Taxonomía WPAN/WLAN/LPWAN, Bandas ISM, Ecuación de Friis, Link Budget, Wi-Fi 6/HaLow, MQTT y Bluetooth Low Energy (GATT y Mesh)

Parte del Curso: Parte 2A — Estado del Arte Inalámbrico, Propagación RF, Wi-Fi y Bluetooth / BLE
Sistemas / Protocolos Estudiados: Ecuación de Friis, Presupuesto de Enlace (Link Budget), Wi-Fi 6 (OFDMA/TWT), Wi-Fi HaLow (802.11ah), MQTT y Bluetooth Low Energy (GATT / Mesh)

```
FSPL[dB] = 20·log10(d_km) + 20·log10(f_MHz) + 32.44 | Fade_Margin = Pr[dBm] - Sensibilidad_min[dBm] >= 15 dB
```

1. Síntesis de Conceptos Clave y Reglas de Ingeniería

- Taxonomía inalámbrica mecatrónica (WPAN, WLAN, LPWAN, WWAN) y el trilema físico fundamental entre alcance geográfico, velocidad de transferencia de datos y consumo energético.
- Propagación en bandas libres ISM (Sub-GHz 433/868/915 MHz frente a 2.4 GHz y 5/6 GHz), cálculo de antena λ/4, ecuación de transmisión de Friis y presupuesto de enlace completo en dBm/dBi con margen de seguridad frente a desvanecimiento multitrayecto en naves metálicas.
- Evolución hacia Wi-Fi 6 (IEEE 802.11ax) con acceso ortogonal OFDMA por Unidades de Recurso (RU), MU-MIMO bidireccional y ahorro energético Target Wake Time (TWT); alcance de 1 km con Wi-Fi HaLow (IEEE 802.11ah Sub-1 GHz) y telemetría industrial con MQTT (QoS 0, 1 y 2), WebSockets y CoAP.
- Diferencias entre Bluetooth Clásico (BR/EDR para flujo continuo SPP) y Bluetooth Low Energy (BLE 5.x con modos 2M PHY y LE Coded PHY de largo alcance); los 40 canales BLE (canales 37, 38 y 39 de Advertising libres de Wi-Fi y salto adaptativo AFH), roles GAP, jerarquía de Servicios y Características GATT (Read, Write, Notify, Indicate) y redes Bluetooth Mesh (Relay, Friend y Low Power Nodes con doble cifrado AES-CCM).

2. Parámetros Nominales de Diseño de Red y Enlace RF (Video 03)

Parámetro de Comunicación / Radiofrecuencia	Símbolo	Valor Nominal	Función en la Red Industrial / Enlace Inalámbrico
Velocidad de Transmisión en Serie	`Baud Rate / Rb`	1000000 bps	Caudal de transferencia en bus físico UART / RS-485 / CAN
Ancho de Banda de Canal	`B [kHz]`	2000 kHz	Determina la capacidad de Shannon-Hartley y tasa de símbolos
Relación Señal a Ruido	`SNR [dB]`	20 dB	Margen sobre el piso de ruido térmico e interferencia industrial
Frecuencia Portadora ISM y Potencia TX	`f [MHz] / Pt [dBm]`	2440 MHz / +20 dBm	Cálculo de longitud de onda λ, antena λ/4 y ecuación de Friis
Factor de Ensanchamiento LoRa y Alcance	`SF / d [m]`	SF7 / 150 m	Compromiso entre sensibilidad de recepción y Tiempo en el Aire
Watchdog de Comunicación Determinista	`T_timeout [ms]`	25 ms	Ventana máxima antes de conmutar actuadores a estado Fail-Safe

3. Desarrollo Académico Diapositiva por Diapositiva

Diapositiva 01: Introducción a los Protocolos Inalámbricos y Estado del Arte

Taxonomía Espacial (WPAN, WLAN, LPWAN, WWAN) y el Trilema Alcance vs. Velocidad vs. Consumo Energético

- **Turno 1 (Masculina):** En la mecatrónica contemporánea, eliminar el cable físico resulta indispensable para dotar de libertad de movimiento a flotas de robots móviles autónomos, drones de inspección, ejes rotativos continuos y redes masivas de sensores industriales.
- **Turno 2 (Femenina):** El estado del arte agrupa los protocolos inalámbricos según su cobertura en cuatro categorías: redes de área personal como Bluetooth y ZigBee, redes locales como Wi-Fi, redes de área amplia y bajo consumo como LoRaWAN y redes celulares como cinco G.
- **Turno 3 (Masculina):** Ningún protocolo inalámbrico único puede resolver todos los problemas, pues las leyes de la física imponen un trilema estricto entre tres variables: alcance geográfico, tasa de transferencia de datos y consumo de energía de la batería.
- **Turno 4 (Femenina):** Si requerimos transmitir video de alta definición usamos Wi-Fi seis o cinco G asumiendo mayor consumo; si buscamos diez años de autonomía a kilómetros de distancia usamos LoRaWAN enviando pocos bytes; y para sensores en máquina elegimos Bluetooth BLE o ZigBee.

Diapositiva 02: Propagación de Radiofrecuencia, Bandas ISM y Ecuación de Friis

Relación λ = c / f, Bandas Sub-GHz (433/868/915 MHz) vs. 2.4/5 GHz y Pérdida en el Espacio Libre (FSPL)

- **Turno 1 (Masculina):** Toda onda electromagnética se propaga a la velocidad de la luz ce, relacionando su frecuencia efe con su longitud de onda lambda mediante lambda igual a ce sobre efe, parámetro que determina el tamaño físico de las antenas de cuarto de onda.
- **Turno 2 (Femenina):** La mayoría de los sistemas mecatrónicos operan en las bandas libres ISM: las bandas sub-gigahercio de cuatrocientos treinta y tres, ochocientos sesenta y ocho y novecientos quince megahercios, y las bandas globales de dos punto cuatro y cinco gigahercios.
- **Turno 3 (Masculina):** La ecuación de transmisión de Friis demuestra que la potencia recibida disminuye con el cuadrado de la distancia de y con el cuadrado de la frecuencia: en decibelios, la pérdida en el espacio libre aumenta seis decibelios cada vez que duplicamos la distancia o la frecuencia.
- **Turno 4 (Femenina):** Por esta razón física, una señal de novecientos quince megahercios sufre casi nueve decibelios menos de atenuación y atraviesa muros industriales con mucha mayor facilidad que una señal de dos punto cuatro gigahercios, aunque dispone de menor ancho de banda.

Diapositiva 03: Presupuesto de Enlace (Link Budget), RSSI y Desvanecimiento Multitrayecto

Cálculo en dBm y dBi, Sensibilidad del Receptor, Margen de Desvanecimiento (Fade Margin) y Reflexiones en Planta

- **Turno 1 (Masculina):** Para diseñar un enlace inalámbrico que jamás se caiga en plena operación industrial, el ingeniero calcula el Presupuesto de Enlace o Link Budget, sumando potencias y ganancias y restando pérdidas utilizando unidades logarítmicas en decibelios.
- **Turno 2 (Femenina):** Expresamos la potencia absoluta en de be eme, referida a un milivatio: cero de be eme equivale exactamente a un milivatio, veinte de be eme equivalen a cien milivatios, y una sensibilidad de menos ciento veinte de be eme representa un picovatio.
- **Turno 3 (Masculina):** La potencia recibida es igual a la potencia transmitida en de be eme, más la ganancia de las antenas en de be i, menos las pérdidas en cables, la pérdida en el espacio libre y la atenuación por obstáculos metálicos.
- **Turno 4 (Femenina):** Para garantizar confiabilidad frente al desvanecimiento multitrayecto causado por rebotes de onda en estructuras de acero, la potencia recibida medida por el indicador RSSI debe superar la sensibilidad mínima del receptor con un margen de seguridad de al menos quince a veinte decibelios.

Diapositiva 04: Estándares Wi-Fi (IEEE 802.11n/ac/ax Wi-Fi 6) en Automatización

Acceso Ortogonal OFDMA por Unidades de Recurso, MU-MIMO Bidireccional y Ahorro Energético TWT

- **Turno 1 (Masculina):** La familia de estándares IEEE ochocientos dos punto once, conocida universalmente como Wi-Fi, ha evolucionado desde redes de oficina hasta convertirse en la columna vertebral de alto caudal para robots móviles y cámaras de visión artificial.
- **Turno 2 (Femenina):** Mientras que Wi-Fi cuatro y Wi-Fi cinco utilizaban modulación OFDM donde un solo dispositivo ocupaba todo el canal en cada instante, el estándar ocho cientos dos punto once a equis, o Wi-Fi seis, introduce el acceso múltiple OFDMA.
- **Turno 3 (Masculina):** Con OFDMA, el punto de acceso divide el canal de radio en pequeñas Unidades de Recurso de frecuencia, permitiendo transmitir y recibir paquetes de decenas de controladores y sensores industriales exactamente en el mismo milisegundo y sin colisiones.
- **Turno 4 (Femenina):** Además, Wi-Fi seis incorpora múltiples antenas multiusuario MU-MIMO en enlace ascendente y descendente, cifrado robusto WPA tres y la función Target Wake Time, que agenda el microsegundo exacto en que cada sensor despierta para transmitir.

Diapositiva 05: Wi-Fi HaLow (IEEE 802.11ah) y Telemetría con MQTT, WebSockets y CoAP

Conectividad IP Sub-1 GHz de Largo Alcance y Arquitectura Publicador/Suscriptor con Calidad de Servicio QoS

- **Turno 1 (Masculina):** Cuando un proyecto mecatrónico necesita las ventajas del protocolo IP nativo de Wi-Fi pero a distancias de hasta un kilómetro atravesando naves industriales, la respuesta del estado del arte es Wi-Fi HaLow, definido por la norma IEEE ochocientos dos punto once a hache.
- **Turno 2 (Femenina):** Al operar en la banda sub-gigahercio alrededor de novecientos megahercios con canales estrechos de uno a ocho megahercios, Wi-Fi HaLow multiplica por diez el radio de cobertura del Wi-Fi tradicional manteniendo compatibilidad con la pila te ce pe i pe.
- **Turno 3 (Masculina):** Sobre esta capa de red, los microcontroladores como el e ese pe treinta y dos y las computadoras Raspberry Pi intercambian telemetría utilizando el protocolo ligero eme cu te te bajo una arquitectura desacoplada de Publicador y Suscriptor mediada por un Broker.
- **Turno 4 (Femenina):** El protocolo eme cu te te ofrece tres niveles de Calidad de Servicio: cero para envío rápido sin acuse, uno para entrega garantizada al menos una vez, y dos para entrega exactamente una vez en comandos críticos, complementado por WebSockets y CoAP.

Diapositiva 06: Arquitectura Bluetooth Clásico (BR/EDR) frente a Bluetooth Low Energy (BLE 5.x)

Flujo Continuo frente a Ráfagas de Ultra-Bajo Consumo y Modos 2M PHY / LE Coded PHY (Largo Alcance)

- **Turno 1 (Masculina):** Al integrar conectividad Bluetooth en un sistema mecatrónico es fundamental comprender que bajo el mismo nombre comercial coexisten dos tecnologías de radio distintas: Bluetooth Clásico y Bluetooth de Baja Energía, abreviado be ele e.
- **Turno 2 (Femenina):** Bluetooth Clásico utiliza setenta y nueve canales de un megahercio en la banda de dos punto cuatro gigahercios y está diseñado para enlaces continuos punto a punto, como el perfil de puerto serie virtual ese pe pe en módulos hache ce cero cinco o transmisión de audio.
- **Turno 3 (Masculina):** Por el contrario, Bluetooth Low Energy fue diseñado desde cero para sensores e instrumentación que permanecen dormidos consumiendo microamperios, despiertan en un par de milisegundos, transmiten su trama de datos y vuelven a dormir inmediatamente.
- **Turno 4 (Femenina):** A partir de Bluetooth cinco, la capa física be ele e añade dos modos revolucionarios: el modo de dos megabits por segundo que reduce a la mitad el tiempo de radio encendido, y el modo codificado LE Coded PHY que cuadruplica el alcance superando los doscientos metros.

Diapositiva 07: Pila de Protocolos BLE: Capa Física AFH, L2CAP, GAP y Conexiones

Los 40 Canales de 2 MHz (3 de Advertising Anti-Wi-Fi + 37 de Datos con Salto de Frecuencia) y Roles GAP

- **Turno 1 (Masculina):** ¿Cómo logra Bluetooth de Baja Energía operar con total fiabilidad en la banda de dos punto cuatro gigahercios incluso cuando hay decenas de redes Wi-Fi transmitiendo alrededor? Gracias al diseño inteligente de sus cuarenta canales de dos megahercios.
- **Turno 2 (Femenina):** Tres de esos canales —el treinta y siete, el treinta y ocho y el treinta y nueve— se dedican exclusivamente a los paquetes de anuncio o Advertising y están situados estratégicamente en los huecos libres entre los canales principales uno, seis y once de Wi-Fi.
- **Turno 3 (Masculina):** Una vez establecida la conexión, los nodos pasan a los treinta y siete canales de datos utilizando Salto Adaptativo de Frecuencia: el enlace cambia de canal hasta mil seiscientas veces por segundo evitando dinámicamente las frecuencias ocupadas por interferencias.
- **Turno 4 (Femenina):** Por encima, el Perfil de Acceso Genérico GAP define los cuatro roles posibles de un dispositivo mecatrónico: Periférico para el sensor o actuador, Central para el controlador maestro o tableta, Emisor sin conexión y Observador.

Diapositiva 08: Servicios, Características y UUIDs en el Perfil GATT de BLE

Jerarquía Perfil -> Servicio -> Característica -> Descriptor (CCCD) y Operaciones Read, Write, Notify e Indicate

- **Turno 1 (Masculina):** En vez de enviar un chorro desestructurado de bytes, Bluetooth Low Energy organiza todas las variables de un nodo mecatrónico dentro de una base de datos orientada a objetos gobernada por el Perfil de Atributos Genéricos, o GATT.
- **Turno 2 (Femenina):** La jerarquía GATT es sumamente clara: un Dispositivo expone uno o varios Servicios, por ejemplo un Servicio de Telemetría de Motor, y cada Servicio contiene múltiples Características individuales, como velocidad en erre pe eme, corriente en amperios o consigna de posición.
- **Turno 3 (Masculina):** Tanto los Servicios como las Características se identifican mediante un número único universal o UUID, que puede ser de dieciséis bits para perfiles estandarizados o de ciento veintiocho bits para desarrollos mecatrónicos propietarios.
- **Turno 4 (Femenina):** El controlador Central puede leer o escribir una Característica, pero para telemetría en tiempo real activamos su descriptor de configuración habilitando las Notificaciones, con las cuales el sensor empuja automáticamente cada nueva muestra sin esperar sondeo.

Diapositiva 09: Redes Bluetooth Mesh en Automatización de Planta

Inundación Gestionada (Managed Flooding), Nodos Relay, Friend y Low Power, y Doble Cifrado AES-CCM

- **Turno 1 (Masculina):** Para cubrir naves industriales completas con cientos de luminarias, sensores ambientales y balizas sin depender de un único nodo central, el estándar Bluetooth Mesh transforma los enlaces be ele e en una red en malla muchos a muchos.
- **Turno 2 (Femenina):** Bluetooth Mesh opera mediante inundación gestionada bajo un modelo de publicación y suscripción: cuando un nodo publica un mensaje hacia una dirección de grupo, los nodos repetidores o Relay lo retransmiten controlando su tiempo de vida te te ele y evitando bucles mediante caché.
- **Turno 3 (Masculina):** Para que los sensores alimentados por batería puedan participar en la malla sin agotar su energía escuchando el tráfico continuo, se establece una relación cooperativa entre un Nodo de Bajo Consumo y un Nodo Amigo conectado a la red eléctrica.
- **Turno 4 (Femenina):** El Nodo Amigo almacena temporalmente todos los mensajes dirigidos al sensor dormido y se los entrega cuando este despierta, todo ello protegido de forma obligatoria con doble cifrado AES de ciento veintiocho bits separando las claves de red y las claves de aplicación.

Diapositiva 10: Síntesis de la Parte 2A y Transición al Siguiente Módulo

Del Presupuesto de Enlace RF, Wi-Fi 6 y BLE hacia ZigBee, LoRaWAN, WirelessHART, UWB y 5G Industrial

- **Turno 1 (Masculina):** En esta primera mitad de la Parte dos hemos dominado los principios físicos de la radiofrecuencia en mecatrónica: la ecuación de transmisión de Friis, el comportamiento de las bandas ISM y el cálculo riguroso del presupuesto de enlace en decibelios.
- **Turno 2 (Femenina):** Estudiamos cómo Wi-Fi seis con acceso ortogonal OFDMA y Wi-Fi HaLow sub-gigahercio habilitan enlaces IP de alta capacidad y largo alcance integrados con protocolos de telemetría como eme cu te te y WebSockets.
- **Turno 3 (Masculina):** Asimismo, desglosamos la arquitectura interna de Bluetooth Low Energy, su salto adaptativo de frecuencia, el modelado de sensores y actuadores mediante servicios y características GATT y la escalabilidad de Bluetooth Mesh.
- **Turno 4 (Femenina):** Acompáñennos en el siguiente módulo, correspondiente a la Parte dos B, donde completaremos el estado del arte inalámbrico analizando ZigBee, la modulación Chirp de LoRa y LoRaWAN, WirelessHART, banda ultraancha y cinco G industrial.

4. Firmware C++/MCU: Trama Industrial, CRC-16 y FSM Fail-Safe (comunicacion_embebida_video_03.ino)

```
/*
 * ============================================================================
 * CURSO COMPLETO: SISTEMAS DE COMUNICACIÓN Y REDES INALÁMBRICAS EN MECATRÓNICA
 * VIDEO 03: Video 03 — Parte 2A: Estado del Arte Inalámbrico, Propagación RF, Wi-Fi y Bluetooth / BLE
 * Archivo: comunicacion_embebida_video_03.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 (25 ms) y estado seguro (Fail-Safe) ante pérdida de enlace.
 *
 * Enfoque del Módulo:
 *   Ecuación de Friis, Presupuesto de Enlace (Link Budget), Wi-Fi 6 (OFDMA/TWT), Wi-Fi HaLow (802.11ah), MQTT y
Bluetooth Low Energy (GATT / Mesh)
 * ============================================================================
 */

#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 03) ---
const uint32_t BAUD_RATE_SERIE   = 1000000UL;
const uint8_t  NODO_DIRECCION_ID = 0x01;       // Dirección de esclavo / nodo mecatrónico
const uint32_t TIMEOUT_ENLACE_MS = 25UL;      // Tiempo máximo sin trama antes de Fail-Safe
const float    FREQ_PORTADORA_MHZ = 2440.0f;
const int8_t   TX_POWER_DBM       = 20;
const uint8_t  LORA_SPREADING_SF  = 7;

// --- 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_03.py)

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
================================================================================
CURSO COMPLETO: SISTEMAS DE COMUNICACIÓN Y REDES INALÁMBRICAS EN MECATRÓNICA
VIDEO 03: Video 03 — Parte 2A: Estado del Arte Inalámbrico, Propagación RF, Wi-Fi y Bluetooth / BLE
Archivo: analisis_redes_y_enlace_rf_video_03.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 (2440 MHz).
    3. Cálculo de Tiempo en el Aire (Time-on-Air) y tasa de bits de la
       modulación Chirp Spread Spectrum (CSS) de LoRa (SF7).
    4. Empaquetado y verificación de tramas Modbus RTU con CRC-16 y payload
       JSON para telemetría MQTT / OPC UA.

Enfoque del Módulo:
  Ecuación de Friis, Presupuesto de Enlace (Link Budget), Wi-Fi 6 (OFDMA/TWT), Wi-Fi HaLow (802.11ah), MQTT y
Bluetooth Low Energy (GATT / Mesh)
================================================================================
"""

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

# --- Parámetros Nominales del Enlace Mecatrónico (Video 03) ---
BW_CANAL_KHZ      = 2000.0
SNR_DB_NOMINAL    = 20.0
FREQ_ISM_MHZ      = 2440.0
TX_POWER_DBM      = 20.0
DISTANCIA_METROS  = 150.0
LORA_SF_NOMINAL   = 7


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 03")
    print(" Enfoque:", "Ecuación de Friis, Presupuesto de Enlace (Link Budget), Wi-Fi 6 (OFDMA/TWT), Wi-Fi HaLow
(802.11ah), MQTT y Bluetooth Low Energy (GATT / Mesh)")
    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_03.st)

```
(* ================================================================================
 * CURSO COMPLETO: SISTEMAS DE COMUNICACIÓN Y REDES INALÁMBRICAS EN MECATRÓNICA
 * VIDEO 03: Video 03 — Parte 2A: Estado del Arte Inalámbrico, Propagación RF, Wi-Fi y Bluetooth / BLE
 * Archivo: pasarela_industrial_plc_video_03.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_V03) 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 25 ms con activación
 *        automática de estado seguro (Fail-Safe) en los actuadores de planta.
 * ================================================================================ *)

FUNCTION_BLOCK FB_GestorRedMecatronica_V03
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. 20 *)

    (* 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 25 MS)
 *    Si no llegan tramas válidas dentro de la ventana determinista, activa Fail-Safe
 * ========================================================================== *)
tonWatchdogRed(
    IN := NOT xLatidoActualizado,
    PT := T#25MS
);

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
```