MANUAL DE ESTUDIO · MECATRÓNICA AUTOMOTRIZ

Redes automotrices: topologías, modelo OSI y arquitectura multibus

Capítulo 03 · En el video: **minuto 30:18** · Duración 14:38

Objetivo. Elegir topología y bus para cada dominio y calcular la carga de una red CAN.

Ideas clave

Del cable dedicado al bus

Cablear cada señal punto a punto multiplicaba conectores y kilos de cobre. Un bus serie multiplexado transmite muchas señales por un par trenzado y permite compartir sensores entre ECU.

Topologías

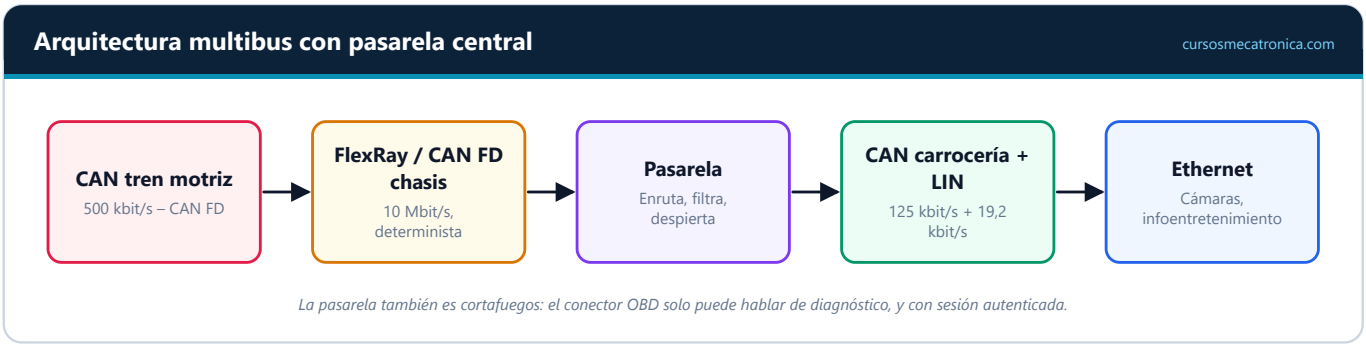
Bus lineal (CAN, LIN): simple, pero un corto lo afecta entero. Estrella (Ethernet, FlexRay activo): aísla fallas en un acoplador. Anillo (MOST): cada nodo repite. Árbol: combinación jerárquica.

Acceso al medio

Maestro-esclavo (LIN): el maestro pregunta y cada esclavo responde. CSMA con arbitraje no destructivo (CAN): habla quien quiere y gana la prioridad más alta. TDMA (FlexRay, TSN): cada nodo tiene su ranura de tiempo, lo que hace el sistema determinista y componible.

Modelo OSI

Los buses automotrices usan sobre todo las capas física, de enlace y de aplicación; el transporte (ISO-TP) aparece cuando un mensaje no cabe en una trama. SAE clasifica las redes por velocidad: clase A (< 10 kbit/s, LIN), B (10–125 kbit/s), C (125 kbit/s–1 Mbit/s, CAN de alta velocidad) y D (> 1 Mbit/s, FlexRay, Ethernet).



Partición de redes por dominio.

Pasarelas

La pasarela enruta mensajes entre redes, traduce formatos, filtra como cortafuegos y gestiona el reposo y el despertar de cada red para no descargar la batería con el auto estacionado.

Ethernet con TSN

Las cámaras y los computadores centrales necesitan gigabits. Time-Sensitive Networking agrega a Ethernet sincronización de relojes y reservas de tiempo para tráfico de control.

Fórmulas y relaciones

Relación	Expresión	Cómo leerla
Bits de una trama CAN (peor caso)	$b = 47 + 8 \cdot DLC + \lfloor (34 + 8 \cdot DLC - 1)/4 \rfloor$	135 bits con 8 bytes.
Carga del bus	$U = \Sigma(b_i/T_i)$ / bitrate	Conviene < 50–60 %.
Ranura TDMA	$t_{ranura} = ciclo$ / $n_{ranuras}$	Latencia acotada.

Ejemplo resuelto: carga del CAN de tren motriz

1. 14 mensajes con periodos de 10 a 1000 ms (la mayoría de 8 bytes).
2. Cada trama de 8 bytes ocupa hasta 135 bits.
3. Suma de bits por segundo dividida por 500 kbit/s: $U = 14{,}0$ %.
4. A 250 kbit/s la carga sube a 27,9 %.

Hay margen para más mensajes y para los reintentos tras errores. carga_bus_y_pasarela.py simula también una pasarela que reenvía, descarta o bloquea tramas según su origen.

Buses automotrices

Bus	Velocidad	Acceso	Uso
LIN	≤ 20 kbit/s	Maestro-esclavo	Espejos, asientos, sensores de lluvia
CAN / CAN FD	0,5–1 / 2–8 Mbit/s	CSMA con arbitraje	Tren motriz, chasis, carrocería
FlexRay	10 Mbit/s	TDMA + dinámico	Chasis activo, by-wire
MOST	25–150 Mbit/s	Anillo síncrono	Audio (en retirada)
Ethernet automotriz	100 Mbit/s–10 Gbit/s	Conmutado, TSN	Troncal, cámaras, ADAS

Código del capítulo

Los archivos completos vienen en el paquete ZIP del capítulo (carpeta `codigo/`).

carga_bus_y_pasarela.py — carga de un bus CAN con 14 mensajes y pasarela con rutas y cortafuegos.

```
"""
carga_bus_y_pasarela.py — Capítulo 03: carga de un bus CAN y pasarela con tabla de
enrutamiento y cortafuegos entre dominios.
"""


def bits_trama(dlc):
    """Trama CAN estándar (11 bits) en el peor caso de relleno de bits."""
    return 47 + 8 * dlc + (34 + 8 * dlc - 1) // 4


def carga_bus(mensajes, bitrate):
    bits_s = sum(bits_trama(dlc) * 1000 / periodo_ms for _, dlc, periodo_ms in mensajes)
    return bits_s / bitrate


TREN_MOTRIZ = [  # (ID, DLC, periodo ms)
```

```
    (0x0A0, 8, 10), (0x0A8, 8, 10), (0x0C0, 8, 10), (0x110, 8, 20), (0x120, 8, 20),
    (0x1A0, 6, 20), (0x1F0, 8, 50), (0x200, 8, 50), (0x280, 4, 100), (0x300, 8, 100),
    (0x3C0, 8, 100), (0x400, 8, 200), (0x480, 2, 500), (0x500, 8, 1000)]

RUTAS = {0x0A0: 'carroceria', 0x1A0: 'chasis', 0x300: 'carroceria'}   # ID -> red destino
DIAGNOSTICO = range(0x7E0, 0x7F0)


def pasarela(id_trama, origen, sesion_diagnostico=False):
    if id_trama in DIAGNOSTICO and not sesion_diagnostico:
        return 'bloqueada (diagnóstico sin sesión autenticada)'
    if origen == 'obd':
        if id_trama in DIAGNOSTICO:
            return 'a tren motriz'
        return 'bloqueada (desde el conector solo se admite diagnóstico)'
    destino = RUTAS.get(id_trama)
    return f'reenviada a {destino}' if destino else 'descartada (sin ruta)'


if __name__ == '__main__':
    print(f'Bits por trama de 8 bytes (peor caso): {bits_trama(8)}')
    for rate in (500_000, 250_000):
        print(f'Carga del CAN de tren motriz a {rate // 1000} kbit/s:'
              f' {100 * carga_bus(TREN_MOTRIZ, rate):.1f} %')
    print('\nPasarela:')
    pruebas = [(0x0A0, 'tren', False), (0x1A0, 'tren', False), (0x123, 'tren', False),
               (0x0A0, 'obd', False), (0x7E0, 'obd', False), (0x7E0, 'obd', True)]
    for t, o, s in pruebas:
        print(f'  0x{t:03X} desde {o:<5s}: {pasarela(t, o, s)}')
```

Errores frecuentes

- Calcular la carga sin los bits de relleno.
- Dejar que el conector OBD tenga acceso directo a las redes internas.
- Usar CAN para video: el ancho de banda no alcanza.

Autoevaluación

1. ¿Qué método de acceso usa LIN?
2. ¿Qué hace determinista a FlexRay?
3. ¿Qué carga tiene el bus del ejemplo a 500 kbit/s?

Respuestas. 1. Maestro-esclavo. **2.** Las ranuras de tiempo fijas (TDMA). **3.** 14,0 %.