MANUAL DE ESTUDIO · MECATRÓNICA AUTOMOTRIZ

Sensores automotrices: principios físicos y protocolos SENT y PSI5

Capítulo 05 · En el video: **minuto 1:00:13** · Duración 15:29

Objetivo. Relacionar cada principio físico con su sensor automotriz y decodificar una trama SENT.

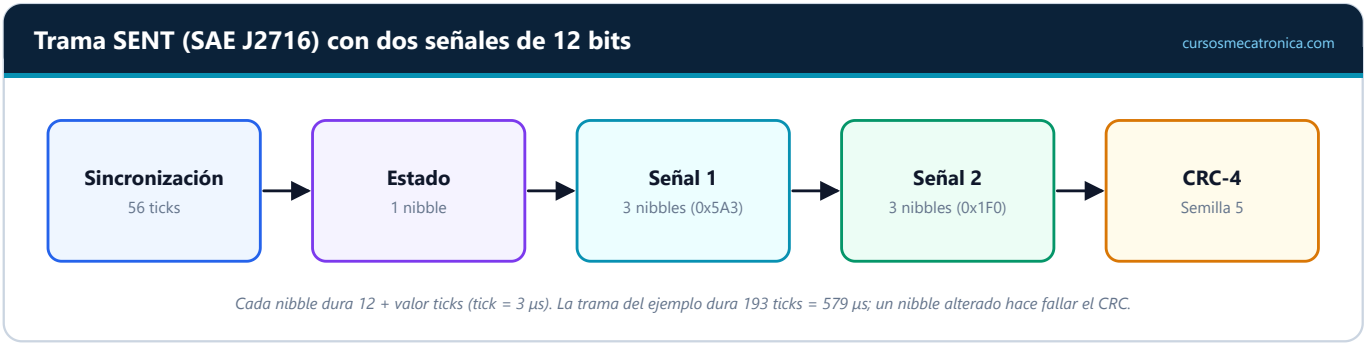
Ideas clave

Mapa de sensores

Tren motriz (presión, temperatura, régimen, lambda), chasis (velocidad de rueda, aceleración, guiñada, ángulo de dirección), carrocería (lluvia, luz, humedad) y asistencia (radar, cámara, ultrasonido).

Niveles de integración

Del elemento sensor solo, al sensor con acondicionamiento, al sensor inteligente con ADC, autodiagnóstico y salida digital. SENT envía datos unidireccionales con pulsos de duración variable; PSI5 es un lazo de corriente de dos hilos para airbags.



Salida de sent_y_tolerancias.py.

Errores y fiabilidad

Offset, ganancia, deriva térmica, histéresis y cuantización se reparten en una banda de tolerancia. El peor caso suma todos; el presupuesto estadístico (RSS) los combina en cuadratura.

Ambiente y micromecánica

Temperaturas extremas, vibración, combustibles y sal. La micromecánica en silicio (MEMS) fabrica estructuras móviles junto a su electrónica.

Principios inductivos y magnéticos

Bobinas, corrientes de Foucault y resolvers miden posición sin contacto. El efecto Hall entrega una tensión proporcional al campo; la magnetorresistencia gigante (GMR) y la anisótropa (AMR) cambian su resistencia con la dirección del campo: la AMR, con estructuras "barber pole", se linealiza para medir ángulo.

Capacitivos, térmicos y piezoeléctricos

Los capacitivos miden desplazamientos MEMS y humedad; NTC y PTC miden temperatura; el piezoeléctrico entrega carga con la fuerza dinámica (detonación). Las cámaras CMOS de alto rango dinámico ven a la vez el sol y la sombra.

Fórmulas y relaciones

Relación	Expresión	Cómo leerla
Efecto Hall	$V_H = I \cdot B \,/\, (n \cdot q \cdot t)$	t: espesor de la placa.
Nibble SENT	$t = (12 + \text{valor}) \cdot \text{tick}$	Tick típico 3 µs.
Peor caso	$e = \Sigma \lvert e_i \rvert$	Conservador.
Estadístico (RSS)	$e = \sqrt{(\Sigma e_i^{\,2})}$	Errores independientes.

Ejemplo resuelto: una trama SENT de presión y temperatura

1. Señal 1 = 0x5A3, señal 2 = 0x1F0, estado 0.
2. Nibbles de datos: 5, A, 3, 1, F, 0; CRC-4 con semilla 5 = 7.
3. Ticks: 56, 12, 17, 22, 15, 13, 27, 12, 19 → 193 ticks = 579 µs.
4. Si un nibble cambia en el camino, el CRC no coincide y la trama se descarta.

El receptor recupera 0x5A3 y 0x1F0. Para el error del sensor, cuatro términos de 0,5 + 0,8 + 1,0 + 0,1 % suman 2,4 % en el peor caso y 1,38 % en RSS (sent_y_tolerancias.py).

Principios físicos y aplicaciones

Principio	Mide	Aplicación
Inductivo / Foucault	Posición, régimen	Cigüeñal, posición del pedal
Hall	Campo magnético	Árbol de levas, velocidad de rueda activa
AMR / GMR	Dirección del campo	Ángulo de dirección, posición
Capacitivo	Desplazamiento, humedad	Aceleración MEMS, humedad
Piezorresistivo	Deformación	Presión
Piezoeléctrico	Fuerza dinámica	Detonación

Código del capítulo

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

sent_y_tolerancias.py — codificación y decodificación SENT con CRC-4 y presupuesto de error peor caso y RSS. *(extracto: primeras 60 de 61 líneas)*

```
"""
sent_y_tolerancias.py — Capítulo 05: codificación y decodificación de una trama SENT
(SAE J2716) con su CRC-4 y presupuesto de error de un sensor.
"""
import math

TICK_US = 3.0


def tabla_crc4():
    """t[i] = i·x⁴ mod (x⁴ + x³ + x² + 1)."""
    t = []
```

```
    for i in range(16):
        v = i << 4
        for bit in range(7, 3, -1):
            if v & (1 << bit):
                v ^= 0b11101 << (bit - 4)
        t.append(v)
    return t


T = tabla_crc4()


def crc4(nibbles):
    crc = 5                      # semilla 0101
    for n in nibbles:
        crc = n ^ T[crc]
    return T[crc]                # aumentado con un nibble 0


def codificar(estado, senal1, senal2):
    datos = [(senal1 >> 8) & 0xF, (senal1 >> 4) & 0xF, senal1 & 0xF,
             (senal2 >> 8) & 0xF, (senal2 >> 4) & 0xF, senal2 & 0xF]
    nibbles = [estado] + datos + [crc4(datos)]
    return [56] + [12 + n for n in nibbles]          # duraciones en ticks


def decodificar(ticks):
    if ticks[0] != 56:
        raise ValueError('falta el pulso de sincronización')
    n = [t - 12 for t in ticks[1:]]
    datos, crc = n[1:7], n[7]
    if crc4(datos) != crc:
        return None
    return (datos[0] << 8 | datos[1] << 4 | datos[2], datos[3] << 8 | datos[4] << 4 | datos[5])


if __name__ == '__main__':
    print('Tabla CRC-4:', T)
    ticks = codificar(0, 0x5A3, 0x1F0)
    print('Ticks:', ticks)
    print(f'Duración de la trama: {sum(ticks)} ticks = {sum(ticks) * TICK_US:.0f} µs')
    print('Decodificada:', [hex(x) for x in decodificar(ticks)])
    ticks[3] += 1                                    # un nibble alterado
    print('Con un nibble alterado:', decodificar(ticks))

    errores = {'offset': 0.5, 'ganancia': 0.8, 'deriva térmica': 1.0, 'cuantización ADC': 0.1}
    peor = sum(errores.values())
    rss = math.sqrt(sum(e * e for e in errores.values()))
```

Errores frecuentes

- Sumar los errores en el peor caso y sobredimensionar el sensor (o hacer RSS con errores correlacionados).
- Olvidar el nibble cero de aumento en el CRC SENT.
- Pensar que un sensor Hall mide velocidad cero igual que un inductivo: el inductivo no genera señal parado.

Autoevaluación

1. ¿Qué protocolo usan los sensores de airbag?
2. ¿Cuánto dura el pulso de sincronización SENT?
3. ¿Qué sensor necesita la estructura barber pole?

Respuestas. 1. PSI5. **2.** 56 ticks. **3.** El AMR.