Interfaz CAN

Transmisión de Datos

Ing. Luis Di Pinto (ldipinto@herrera.unt.edu.ar)

http://www.microprocesadores.unt.edu.ar/transmision/

Conceptos clave clase anterior

- **Interfaz I2C**: *Inter-Integrated Circuit*.
  - Estándar de facto para comunicaciones de corta distancia.
  - Comunicación sincrónica, half dúplex, multipunto, de tipo Master-Slave, que admite múltiples Masters.
  - Solamente dos líneas: SDA (datos) y SCL (clock).
  - Simple y de bajo costo.
  - Bus compartido único, con direccionamiento de 7 bits.
  - Cinco velocidades de operación.
- Protocolo simple, con control de flujo
  - Genera overhead.
- Permite arbitraje del bus.

Repaso Esquema General (2da parte)

- En esta segunda parte de la materia, vamos a enfocarnos en un conjunto de interfaces seriales.
  - Algunas de las más usadas y populares.



Temas que veremos

- Características generales.
- Esquemas de conexión.
- Tramas.
- Sincronización.
- Arbitraje.
- Control de flujo.
- Variantes.

Introducción

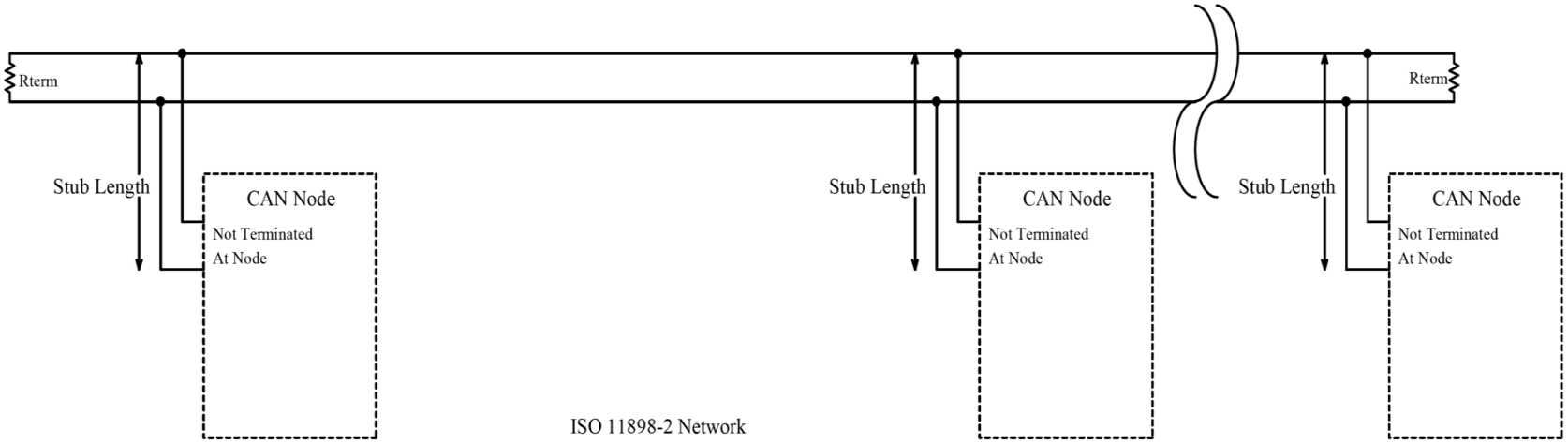
- *Controller Area Network*
- Estándar ISO 11898, del 2003.
  - Originado en Bosch a mediados de 1980, inicialmente pensado para automóviles.
  - Permite comunicar múltiples subsistemas, diferentes entre sí, a través de un único bus.
- Usado principalmente en ambientes ruidosos o industriales:
  - Vehículos, ascensores, equipos médicos, robótica.
- Permite múltiples dispositivos Master, con un esquema de prioridades.
- Asincrónica, *half dúplex*, esquema multicast.
- Implementa un protocolo basado en mensajes.
  - Define la capa de enlace y la capa física, para conexiones de alta y de baja velocidad.

Características de conexión

- Es un bus de dos cables.
  - Al igual que I2C.
  - Usualmente, un par trenzado enmallado.
  - Con una impedancia característica de 120 Ω.
- Usa señales **diferenciales**.
  - Como RS-485.
  - Colector abierto, con resistencias de *pull up*.
- Las señales se denominan simplemente CANH y CANL.
  - En estado inactivo (recesivo), o al enviar un ‘1’, CANH ≤ CANL, las líneas permanecen con tensión.
  - En estado activo (dominante), al enviar un ‘0’, CANH > CANL.
- La conexión es diferente según la velocidad pretendida.

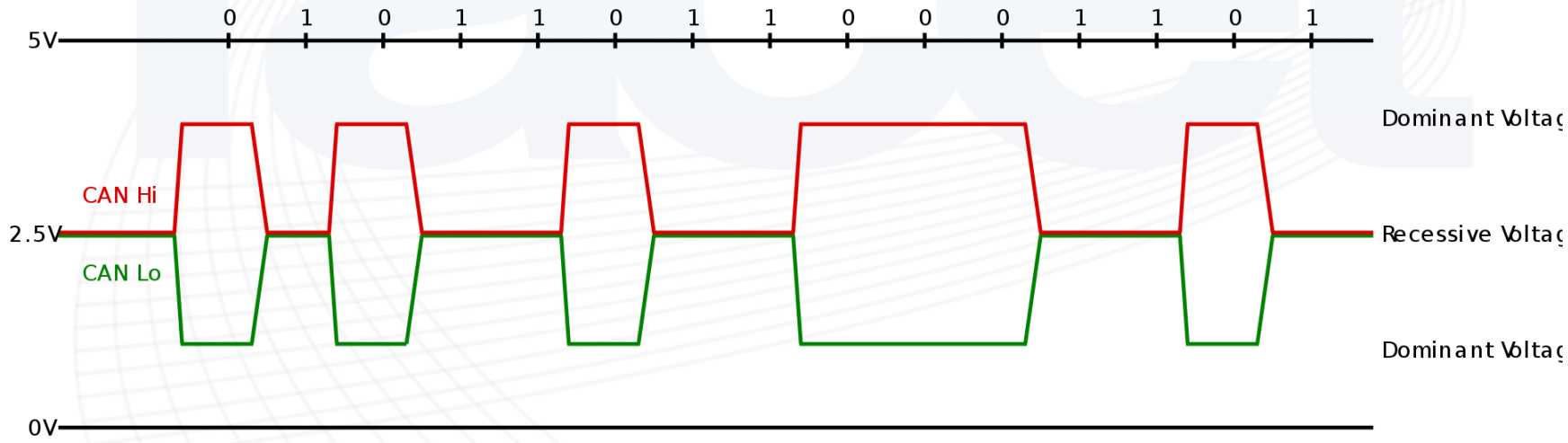
Conexión de alta velocidad

- Hasta 1Mbps, estándar ISO 11898-2.
  - Conexión lineal, con resistencias de 120 Ω **en cada extremo**.
  - Longitud máxima de 40 m para 1 Mbps.
    - Menores velocidades permiten mayores longitudes.



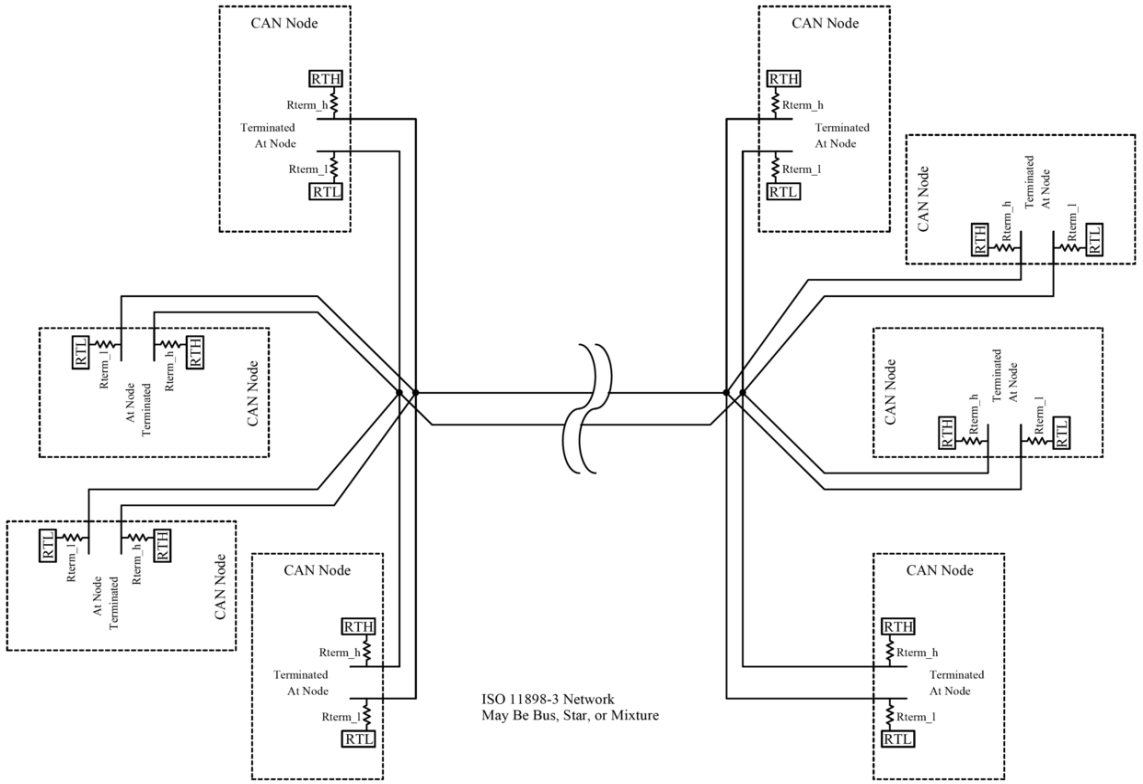
Conexión de alta velocidad

- Para transmitir un ‘1’ se define una tensión de modo común de 2.5 V en cada línea. Estado de reposo (recesivo).
- Para transmitir un ‘0’, CANH = 3.5 V y CANL = 1.5 V. Estado activo (dominante)
- Esto produce que:
  - La tensión diferencial en ‘0’ es de 2 V.
  - Cualquier tensión diferencial < 0.5 V será considerada un ‘1’.



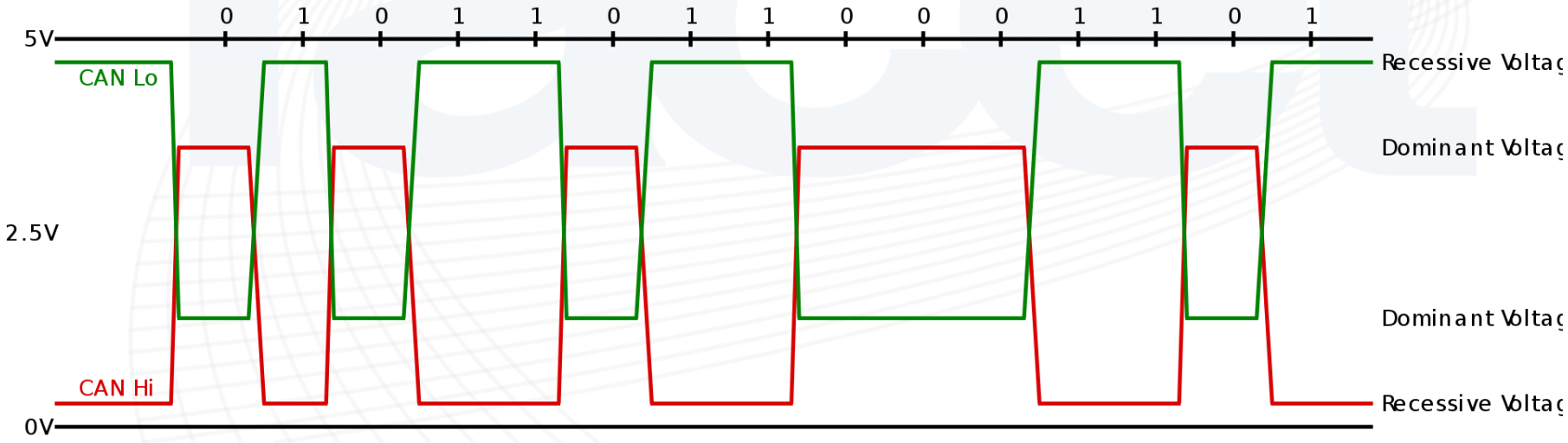
Conexión de baja velocidad

- Hasta 125 kbps, estándar ISO 11898-3.
  - También denominada conexión tolerante a fallos.
  - Longitud de hasta 500 m para 125 kbps, y 1000 m a 50 kbps.
- Puede ser una conexión lineal, o estrella, o múltiples estrellas conectadas por un bus.
- Tiene resistencias de terminación **en cada nodo**.
  - La resistencia de terminación total debe ser cercana (no menor) a 100 Ω.



Conexión de baja velocidad

- Para transmitir un ‘1’ (estado de reposo o recesivo), CANH = 0 V y CANL = 5 V.
- Para transmitir un ‘0’, CANH = 5 V y CANL = 0 V.
  - Esto produce que un receptor deba simplemente mirar la diferencia de tensión entre CANH y CANL.
- Vcc puede ser 3.3 V en vez de 5 V.



Conectores

- El estándar no define aspectos mecánicos.
- Por lo que surgieron varias formas de conectores.
  - El más común es el conector DB 9 (de RS-232).
  - Asignación de pines: CANL (2), CANH (7), GND (3) y VCC (9).

Nodos CAN

- Se denomina nodo en una red CAN a cada dispositivo que se conecta al bus, y está compuesto por:
  - Un sensor o actuador.
  - Un procesador encargado de manejar los mensajes.
  - Un controlador CAN, que prepara el mensaje para ser transmitido/recibido.
    - Normalmente, parte del mismo microcontrolador.
  - Un *transciever* CAN, que prepara las señales para adaptarse al bus.
    - Usualmente no es parte del microcontrolador, sino que es un circuito externo.

Capa de enlace

- Todo lo anterior constituye la capa física del estándar.
- Esta parte del estándar (la ISO 11898-1) se encarga de:
  - Bit timing y sincronización.
  - Armar las tramas de los mensajes.
  - Arbitraje.
  - Control de flujo.
  - Detección de errores.

Formato de las tramas

- Las redes CAN soportan dos formatos de tramas:
  - Estándar (o base), definido como CAN 2.0A.
  - Extendido, definido como CAN 2.0B.
- La única diferencia entre ambos es el tamaño del campo identificador (ID):
  - 11 bits para el formato base y 29 bits para el formato extendido.
- La distinción se realiza mediante un bit que se transmite, denominado IDE (*Identifier Extension bit*):
  - Vale ‘0’ para el formato base y ‘1’ para el extendido.
- Los controladores CAN que soportan formato extendido también pueden enviar y recibir mensajes en formato base.
- El ID debe ser único para cada dispositivo del bus.

Tipos de tramas

- Las redes CAN soportan cuatro tipos de tramas:
  - Tramas de Datos: contiene datos a transmitir.
  - Tramas remotas: para solicitar la transmisión de un ID determinado.
  - Tramas de error: cuando algún nodo detecta un error.
  - Tramas de sobrecarga (*overload*): para insertar un retardo entre tramas de datos o remotas.
- Todas las tramas comienzan con un bit de inicio, denominado SOF (*Start of Frame*):
  - Que es igual a ‘0’.
- Entre dos tramas de datos o remotas se deja un campo de 3 bits denominado IFS (*Inter Frame Spacing*).
  - Todos iguales a ‘1’.

Trama de Datos – Formato base



- RTR (Remote Transmission Request): si es ‘0’ indica que envía datos sino que los solicita de otro nodo.
- Control:
  - IDE (1 bit).
  - DLC (Data Length Code, 4 bits): especifica la cantidad de bytes a transmitir, entre 0 y 8.
    - Es posible enviar un DLC mayor a 8, pero la cantidad de datos real es limitada a 8.
  - R0 (1 bit): reservado, se acepta que puede ser ‘0’ o ‘1’.
- ACK: para control de flujo.
  - El transmisor envía un ‘1’ y los receptores pueden poner un ‘0’.
- EOF: finalización de trama, todos bits en ‘1’.

Trama de Datos – Formato extendido

- Similar a la anterior, pero:
- IDE bit igual a 1.
- A continuación, los 18 bits restantes del ID.
- Dos bits reservados, r0 y r1.

Otros tipos de Tramas

- Tramas remotas: como las de Datos, con dos diferencias:
  - El bit RTR en 1 y que no tienen campo de datos.
  - El campo DLC indica la cantidad de bytes que se desea recibir.
- Tramas de error: consisten de dos campos
  - Flags de errores (entre 6 y 12 bits).
  - Delimitador de error (ocho ‘1’ consecutivos).
- Tramas de sobrecarga:
  - Flag de sobrecarga (seis ‘0’ consecutivos).
  - Delimitador de sobrecarga (ocho ‘1’ consecutivos).
  - Una trama de sobrecarga se inicia si se detecta un ‘0’ en medio del espaciado entre tramas.

Sincronización

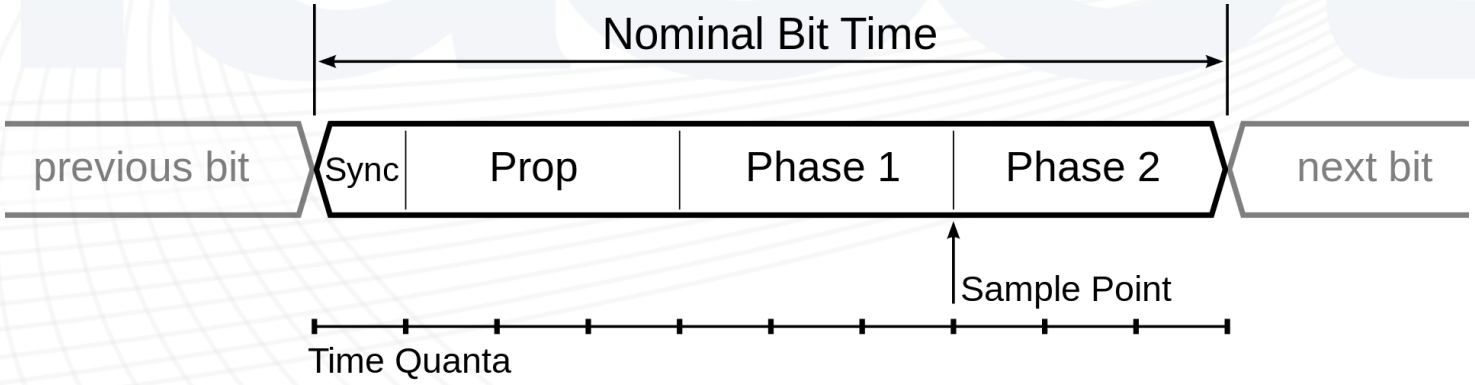
- Todos los nodos conectados deben operar a la misma velocidad de transmisión.
- La sincronización se hace con cada transición a un ‘0’ (bit dominante) dentro de una trama.
- Como para la codificación de línea se usa normalmente NRZ-L, se usa *bit stuffing* en caso de cinco bits consecutivos de la misma polaridad.
  - Con excepción de los campos ACK y EOF, que son de tamaño fijo.
- El controlador CAN espera que esta transición ocurra en un múltiplo del bit time.
  - Si no ocurre, se ajusta el bit time consecuentemente.

Sincronización – Bit *timing*

- Cada bit es dividido en cuatro segmentos:
  - Sincronización, cuando ocurren las transiciones de ‘1’ a ‘0’.
  - Propagación, tiempo para compensar retardos de propagación en la línea.
  - Fase 1 y Fase 2, usados para la sincronización.
- El muestreo se realiza entre las fases 1 y 2.
  - Es común que el muestreo no se realice a mitad del bit, como en otras interfaces vistas.

Sincronización – Bit *timing*

- A su vez, también es dividido en intervalos de igual duración, denominados *quantas*.
  - La cantidad de intervalos es dependiente del controlador CAN.
- A cada segmento se le asignan una determinada cantidad de intervalos.
  - La cantidad de *quantas* para cada segmento depende de la velocidad de transmisión y de las condiciones de la red.
- Por ejemplo, si asumimos que el bit se divide en 10 *quantas*, una posible asignación sería:



Sincronización – Ajuste

- Si una transición a ‘0’ ocurre antes o después de lo esperado, el controlador CAN calcula la diferencia y modifica la cantidad de *quantas* asignados a las fases 1 y 2.
  - En el ejemplo anterior, si la transición llegase antes, el controlador cambiaría a 2 *quantas* para fase 1 y 4 *quantas* para fase 2.
- Este proceso se realiza con cada transición a ‘0’, asegurando que siempre los receptores estén sincronizados con el transmisor.
  - Y evita errores introducidos por ruido en la red.

Arbitraje del bus

- Todos los nodos conectados al bus deben estar sincronizados, para muestrear cada bit al mismo tiempo.
- Si dos nodos intentan transmitir al mismo tiempo, se impone el de mayor prioridad.
  - ***¿Cuál sería?***
  - La idea es que aquellos dispositivos que requieran que su mensaje llegue con menor retardo a destino tengan mayor prioridad, o sea, un ID más bajo (con más ‘0’s).
- El nodo de menor prioridad automáticamente deja de transmitir e intentará una retransmisión 6 ciclos de reloj después de terminada la transmisión del de mayor prioridad.
  - Si son más de dos nodos, el proceso es el mismo.
  - Se repite hasta que queda solamente un nodo transmitiendo.

Control de flujo

- Las tramas de datos contienen un campo de 2 bits para control de flujo.
- Cada nodo que reciba la transmisión sin errores debe insertar un ‘0’ en este momento.
  - El transmisor envía (mantiene) un ‘1’.
- Si el transmisor no detecta un ‘0’, sabe que el mensaje no fue recibido.
- En tal caso, lo más común es retransmitir la trama hasta que sea efectivamente recibida.

CAN FD

- *Flexible Data rate*, del 2012.
- Compatible con versiones anteriores.
- Velocidad de hasta 5 Mbps.
- Extiende la longitud de la sección de datos de la trama de 8 a 64 bytes.
- Permite aumentar la velocidad **luego del arbitraje**.
  - Multiplica la frecuencia por un factor que puede ser hasta 8x.
- También aumenta el tamaño del CRC a 21 bits, para mejorar la detección de errores.

Resumen

- **CAN**: *Controller Area Network*.
  - Estándar para comunicaciones que necesitan ser robustas.
  - Comunicación asincrónica, half dúplex, de tipo multicast.
  - Solamente dos líneas: CANH y CANL, balanceadas.
  - Dos esquemas de conexión, con diferentes velocidades de operación.
- Protocolo no tan simple como otras interfaces.
  - Varios tipos de tramas, mucho overhead.
- Permite arbitraje del bus, detección de errores y un control de flujo básico.

Agradecimientos

- Las diapositivas de este tema fueron basadas en las realizadas por el Ing. Nicolás Majorel Padilla.