

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

**Facultad de Ciencias de la Electrónica
Licenciatura en Electrónica**



**Manual de pruebas de CAN–Bus para arquitecturas
electrónicas de automóviles**

TESIS

que para obtener el grado de

Licenciado en Ciencias de la Electrónica

presenta

Juan Carlos Arroyo Núñez

Asesor

Dr. Pablo Sánchez Sánchez – (FCE)

CU, Puebla. Pue., Octubre 2022

Hoja de firmas

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
Facultad de Ciencias de la Electrónica
Licenciatura en Electrónica

El presente documento certifica que se ha leído el trabajo de Tesis que tiene por nombre **Manual de pruebas de CAN–Bus para arquitecturas electrónicas de automóviles** realizado por el alumno **Juan Carlos Arroyo Núñez** recomendando al Posgrado sea aceptada para cumplir con uno de los requisitos para obtener el grado de **Licenciado en Ciencias de la Electrónica**.

7 de noviembre de 2022

Presidente: _____
Dr. José Fernando Reyes Cortés

Secretario: _____
Dr. Roberto Carlos Ambrosio Lázaro

Vocal: _____
Dr. Luis Armando Moreno Coria

Lugar donde se realizó la tesis:

Laboratorio de Volkswagen de México SA de CV y Edificio de Posgrado de la Facultad de Ciencias de la Electrónica de la Benemérita Universidad Autónoma de Puebla (BUAP).

Agradecimientos

Primero me gustaría agradecerte a ti Dios, por bendecirme para llegar hasta donde he llegado, porque hiciste realidad este sueño.

A mis padres por haber confiado en mí y el haberme apoyado hasta el final de este proyecto que hoy concluyo.

A la Benemérita Universidad Autónoma de Puebla por darme la oportunidad de estudiar y ser un profesionista. A la planta académica de la Facultad de Ciencias de la Electrónica por su esfuerzo y dedicación, quienes, con sus conocimientos, sus experiencias, su paciencia y su motivación ha logrado en mí que pueda terminar mis estudios con éxito y en especial al Dr. Pablo Sánchez Sánchez de quien obtuve el apoyo para concluir este trabajo.

También me gustaría agradecer a la Volkswagen de México SA de CV, empresa en la cual me desarrollo profesionalmente, y que durante la estancia que llevo me ha dado el apoyo para salir adelante y concluir con esta etapa de mi vida profesional.

Juan Carlos Arroyo Núñez

Dedicatoria

Son muchas las personas que han formado parte de mi vida profesional. Son muchas las personas a las que me encantaría agradecerles su amistad, sus consejos, su apoyo, su ánimo y su compañía en los momentos más difíciles de mi vida. Algunas están aquí conmigo y otras en mis recuerdos y en mi corazón, sin importar en donde estén quiero darles las gracias por formar parte de mí, por todo lo que me han brindado y por todas sus bendiciones .

Índice general

Hoja de firmas	III
Agradecimientos	V
Dedicatoria	VII
1. Introducción	1
1.1. Antecedentes	1
1.2. Desarrollo del CAN Bus	3
1.3. Objetivos	4
1.3.1. Objetivo general	4
1.3.2. Objetivos específicos	4
1.4. Organización de la tesis	5
2. Generalidades del CAN Bus	7
2.1. Introducción	7
2.2. Sistema OBD–II	9
2.2.1. Modos de funcionamiento del sistema OBD–II	12
2.3. El protocolo CAN	14
2.4. Principales características de CAN-Bus	18
2.4.1. Capa de enlace	18
2.4.2. Capa física	19
2.4.3. Codificación	20
2.4.4. Niveles eléctricos del bus	21
2.4.5. Formato de tramas	22
2.4.5.1. Trama de datos	22
2.4.5.2. Inicio de trama (SOF)	23
2.4.5.3. Arbitraje	23
2.4.5.4. Control	23
2.4.5.5. Datos	24
2.4.5.6. CRC	24
2.4.5.7. Campo de reconocimiento (ACK)	24
2.4.5.8. Fin de trama (EOF)	24
2.4.5.9. Espaciado entre tramas (IFS)	24
2.4.5.10. Trama de datos extendida	24

2.4.5.11.	Trama Remota	25
2.4.5.12.	Trama de Error	26
2.4.5.13.	Espacio entre tramas	27
2.4.5.14.	Bus inactivo	28
2.4.5.15.	Trama de Sobrecarga	28
2.4.5.16.	Manejo de error	29
2.4.5.17.	Señalización de error	30
2.4.5.18.	Aislamiento de fallos	30
2.4.5.19.	Tolerancia del oscilador en el bus CAN	32
2.4.5.20.	Requerimientos para el tiempo de bit	32
2.4.5.21.	Tiempo nominal de bit.	33
2.4.5.22.	Sincronización	35
2.4.5.23.	Errores de fase	37
2.4.5.24.	Reglas de sincronización	37
2.5.	Protocolo de comunicaciones CAN-Bus	37
2.5.1.	Elementos que componen el sistema CAN-Bus	39
2.5.1.1.	Cables	39
2.5.1.2.	Elemento de cierre o terminador	40
2.5.1.3.	Controlador	41
2.5.1.4.	Transmisor-Receptor	42
2.5.2.	El CAN-Bus en los autos	42
2.5.3.	CAN Low Speed	44
2.5.4.	CAN High Speed	44
2.6.	CAN FD	48
2.6.1.	Beneficios del CAN FD	49
2.6.2.	CAN FD vs CAN	49
3.	Etapa física del CAN-bus	51
3.1.	Resistencia de cierre de CAN-HIGH	51
3.1.1.	Niveles de señal CAN High	53
3.1.2.	Pruebas de comunicación	56
3.1.2.1.	Estabilidad del anillo de comunicación	57
3.1.2.2.	Tranquilidad del BUS	59
3.1.2.3.	Formación del anillo de comunicación	61
4.	Diagnóstico	65
4.1.	Diagnóstico OBD	65
4.1.1.	OBD I	65
4.1.2.	OBD II	65
4.2.	Plataforma de diagnóstico	66
4.2.1.	Funcionamiento de software de diagnóstico	67
4.2.2.	Lectura de fallas en las unidades de control	68
4.2.3.	Borrado de fallas de la unidad de control	70
4.2.4.	Protocolo de diagnóstico	71

5. Análisis de plan de estudios ISA	73
5.1. Movilidad	73
5.2. Herramientas computacionales	74
5.3. Inmovilizador	74
5.4. Conectividad y encriptación	74
5.5. Buses de comunicación	75
5.6. Asignatura de Interfaces y Redes Vehiculares	75
5.7. Propuesta de práctica de análisis red automotriz	76
Conclusiones	81
A. Cálculo de la resistencia de cierre	81
B. Mediciones	83
C. Pruebas de comunicación	85
D. Instructivo Vector-CANoe	87
D.1. Interfaces (Hardware)	87
D.1.1. Piggies	87
D.1.2. VN8900	88
D.1.3. VN1600	89
D.1.4. VN2640	89
D.2. Uso del CANoe	90
D.2.1. Configuración del Hardware	90
D.2.2. Configuración de la Simulación	92
D.2.3. Configuración de una medición de CAN-Bus	93
D.2.4. Iniciar Medición	94
D.2.5. Finalizar Medición	95
D.2.6. Traces	96
E. Reconocimientos	99
Bibliografía	101

Índice de figuras

1.1. Sistemas electrónicos que utilizan el CAN-bus.	3
2.1. Centrales de control en un Automóvil.	7
2.2. Esquema de Bus de comunicación con 5 unidades de control.	8
2.3. Certificado OBD-II	10
2.4. Conector OBD-II y sus terminales.	11
2.5. Conexión sin bus CAN	15
2.6. Conexión con bus CAN	15
2.7. Esquema de selección de mensaje de las unidades de control.	17
2.8. Ejemplo de codificación NRZ	20
2.9. Datos sin relleno (arriba) y datos con relleno (abajo).	20
2.10. Niveles del bus.	21
2.11. Trama de datos estándar.	22
2.12. Campo de control.	23
2.13. Trama de datos extendida.	25
2.14. Trama Remota con identificador extendido. Sin campo de datos.	26
2.15. Trama de error.	27
2.16. Trama de sobrecarga.	28
2.17. Segmentos del tiempo de bit en CAN.	33
2.18. Re-sincronización el tiempo de bit.	36
2.19. Par trenzado de CAN Bus.	39
2.20. Valores recesivos y dominantes en CAN Bus.	40
2.21. Resistencia de cierre del CAN Bus.	41
2.22. Controlador de las unidades de control.	41
2.23. Transceiver de las unidades de control.	42
2.24. Unidad de control	43
2.25. Red de comunicación del BUS de Confort.	45
2.26. Red de comunicación del BUS de Infotainment.	46
2.27. Red de comunicación del BUS de Propulsión.	47
2.28. Red de comunicación del BUS de Clúster.	48
3.1. Circuito de resistencia de cierre en unidades de control.	51
3.2. Circuito de resistencia de cierre en unidad de control.	53
3.3. Diagrama a bloques de conexión para medición de los niveles de señal.	53
3.4. Ancho de pulso CAN High Speed (1.998, . . . , 2.002 μ s)	54

3.5. Nivel recesivo y dominante CAN H.	55
3.6. Nivel recesivo y dominante CAN L.	56
3.7. Diagrama a bloques de conexión para medición de estabilidad del anillo.	57
3.8. Gráfica de CANoe de permanencia del anillo en la prueba de estabilidad.	58
3.9. Imagen de la ventana de estadísticas del bus.	58
3.10. Diagrama a bloques de conexión para medición de tranquilidad del Bus.	60
3.11. Configuración estándar de CANoe.	60
3.12. Imagen de la ventana de estadísticas del bus.	61
3.13. CANExtender-AA.	62
3.14. Diagrama a bloques para medición de anillo de comunicación.	62
3.15. Simulación de switch de encendido.	63
3.16. Adquisición de datos.	63
3.17. Bit de formación del anillo de comunicación.	64
4.1. Centrales de control en un Automóvil.	66
4.2. Pantalla principal de plataforma de diagnóstico.	67
4.3. Softkey de la lista codificada de Gateway.	67
4.4. Lista de montaje en el Gateway.	68
4.5. Selección de unidad de control.	68
4.6. Lectura de memoria de eventos.	69
4.7. Eventos en la unidad de control.	69
4.8. Softkey regresar ventana anterior.	70
4.9. Borrar memoria de eventos.	70
4.10. Borrar memoria de eventos.	70
4.11. Memoria de eventos después del borrado.	71
4.12. Softkey Imprimir.	71
4.13. Protocolo de diagnóstico.	72
5.1. Fases de ISA.	73
A.1. Circuito de resistencia de cierre en unidades de control.	81
D.1. Piggies.	87
D.2. Piggies.	88
D.3. VN1600.	89
D.4. VN2640.	89
D.5. pantalla principal de CANoe	90
D.6. Configuración del Hardware.	90
D.7. menú de Configuration	91
D.8. Velocidad de la transmisión.	91
D.9. Panel.	92
D.10.ventana de CANoe	93
D.11.Rename.	93
D.12.visualizan de los mensajes.	94
D.13.Iniciar Medición.	94
D.14.Ventana de Gráficas.	95

D.15.opción de Add Signals	95
D.16.opción de Separate Views	96
D.17.configurar el tiempo de Pre-Trigger.	97
D.18.programas de CANoe /CANalyzer.	98
E.1. Reconocimiento	99

Índice de tablas

2.1. Terminales del conector OBD–II	12
2.2. Número de bytes de datos en el DLC. Dominante D y Recesivo R	24
2.3. Colores de cables de CAN Bus.	39
3.1. Medición de Resistencia total del Bus.	52
3.2. Ancho de pulso CAN High Speed.	54
3.3. Nivel de señal $UCAN_H$ para CAN High Speed	54
3.4. Nivel de señal $UCAN_L$ para CAN High Speed	55
3.5. Datos obtenidos por el programa CANoe.	57
3.6. Prueba de estabilidad del anillo de comunicación.	59
3.7. Consumo de corriente de Low Speed y High Speed.	59
3.8. Tranquilidad de Bus.	61
3.9. Formación del anillo de comunicación.	64
A.1. Medición de Resistencia total del Bus.	82
B.1. Ancho de pulso CAN High Speed.	83
B.2. Nivel de señal $UCAN_H$ para CAN High Speed	83
B.3. Nivel de señal $UCAN_L$ para CAN High Speed	83
C.1. Prueba de estabilidad del anillo de comunicación.	85
C.2. Tranquilidad de Bus.	86
C.3. Formación del anillo de comunicación.	86

Capítulo 1

Introducción

Al hablar de la comunicación electrónicas en el área automotriz, estamos hablando de una nueva era en el desarrollo de los autos. El desarrollo de nuevas unidades de control obliga a la introducción de nuevos protocolos de comunicación en las redes automotrices esto hace que la electrónica en el automóvil se haga más compleja.

El **CAN Bus** es el protocolo más usado hasta el día de hoy por los fabricantes de autos dejando atrás los protocolos propios desarrollados por las empresas automotrices de manera particular. **CAN (siglas del inglés Controller Area Network)** es un protocolo de comunicaciones desarrollado por la firma alemana Robert Bosch GmbH, basado en una topología bus para la transmisión de mensajes en entornos distribuidos. Además ofrece una solución a la gestión de la comunicación entre múltiples CPUs (unidades centrales de proceso).

Este protocolo ofrece alta inmunidad a las interferencias, habilidad para el autodiagnóstico y la reparación de errores de datos. Es un protocolo de comunicaciones normalizado, con lo que se simplifica y economiza la tarea de comunicar subsistemas de diferentes fabricantes sobre una red común o bus. El procesador anfitrión (host) delega la carga de comunicaciones a un periférico inteligente, por lo tanto el procesador anfitrión dispone de mayor tiempo para ejecutar sus propias tareas. Al ser una red multiplexada, reduce considerablemente el cableado y elimina las conexiones punto a punto, excepto en los enganches.

1.1. Antecedentes

El desarrollo del protocolo CAN comenzó en 1983 en la empresa Robert Bosch GmbH (comúnmente conocida como Bosh). El protocolo fue oficialmente lanzado en 1986 en el congreso de la Sociedad de Ingenieros Automotrices (SAE) en Detroit. Los primeros controladores CAN llegaron al mercado en 1987 de la mano de Intel y Philips.

Bosch publicó posteriormente varias versiones de la especificación CAN, siendo la última de ellas la especificación CAN 2.0, publicada en 1991. Esta especificación consta de dos partes; la parte A para el formato estándar y la parte B para el formato extendido.

Un dispositivo CAN que usa el formato estándar utiliza identificadores de 11 bits y es comúnmente referido como dispositivo CAN 2.0A. Un dispositivo CAN que usa el formato extendido usa identificadores de 29 bits y es comúnmente referido como dispositivo CAN 2.0B. Los estándares CAN 2.0A/B y otros documentos de referencia relacionados con CAN son de acceso libre a través de Bosch.

En 1993 se publicó el estándar **ISO 11898** del bus CAN y ha sido a partir de ese momento un estándar de la Organización Internacional para la Normalización. Actualmente el CAN-bus está estandarizado por las siguientes normas:

- **ISO 11898-1:2015**, Parte 1: Data link layer and physical signalling
- **ISO 11898-2:2016**, Parte 2: High-speed medium access unit
- **ISO 11898-3:2006**, Parte 3: Low-speed, fault-tolerant, medium-dependent interface. Este estándar ha sido revisado y confirmado en 2015
- **ISO 11898-4:2004**, Part 4: Time-triggered communication. Este estándar ha sido revisado y confirmado en 2013
- **ISO 11898-5:2007**, Part 5: High-speed medium access unit with low power mode
- **ISO 11898-6:2013**, Part 6: High-speed medium access unit with selective wake-up functionality
- **ISO 16845:2016**, Conformance test plan

En 2011 Bosch, en cooperación con los fabricantes de automóviles y otros expertos del CAN-bus, comenzó a desarrollar la siguiente generación del CAN.

El protocolo CAN FD (flexible data-rate). El CAN FD es compatible hacia atrás, es decir, un controlador CAN FD es capaz de comprender el mensaje CAN clásico (o CAN 2.0). Por el contrario, un controlador CAN clásico destruye un mensaje CAN FD emitiendo un mensaje de error. EL nuevo CAN FD es capaz de transmitir datos más rápido que 1 Mbps (la velocidad máxima del CAN clásico). Ambos protocolos, el CAN clásico y el CAN FD, están estandarizados en la norma **ISO/DIS 11898-1.34**

CAN (Área de Red Controlada) es un protocolo de comunicación de computadoras interconectadas aplicado a la industria automotriz. Actualmente está siendo usado por automóviles de Peugeot Soci t  Anonyme PSA, Volkswagen Aktiengesellschaft VAG, MERCEDES, BMW, VOLVO, RENAULT, FORD, General Motors GM, MAZDA, SAAB Automobile, entre otras marcas y es obligatorio para todos los autos a partir del 2008 en los pa ses desarrollados.

El sistema computarizado de los vehículos ha evolucionado desde una computadora que controla todo a un conjunto de computadoras, cada una con su propia área de responsabilidad. Este implica cambiar la forma en que una computadora se comunica con cada una de las otras para compartir información de los procesos que están controlando.

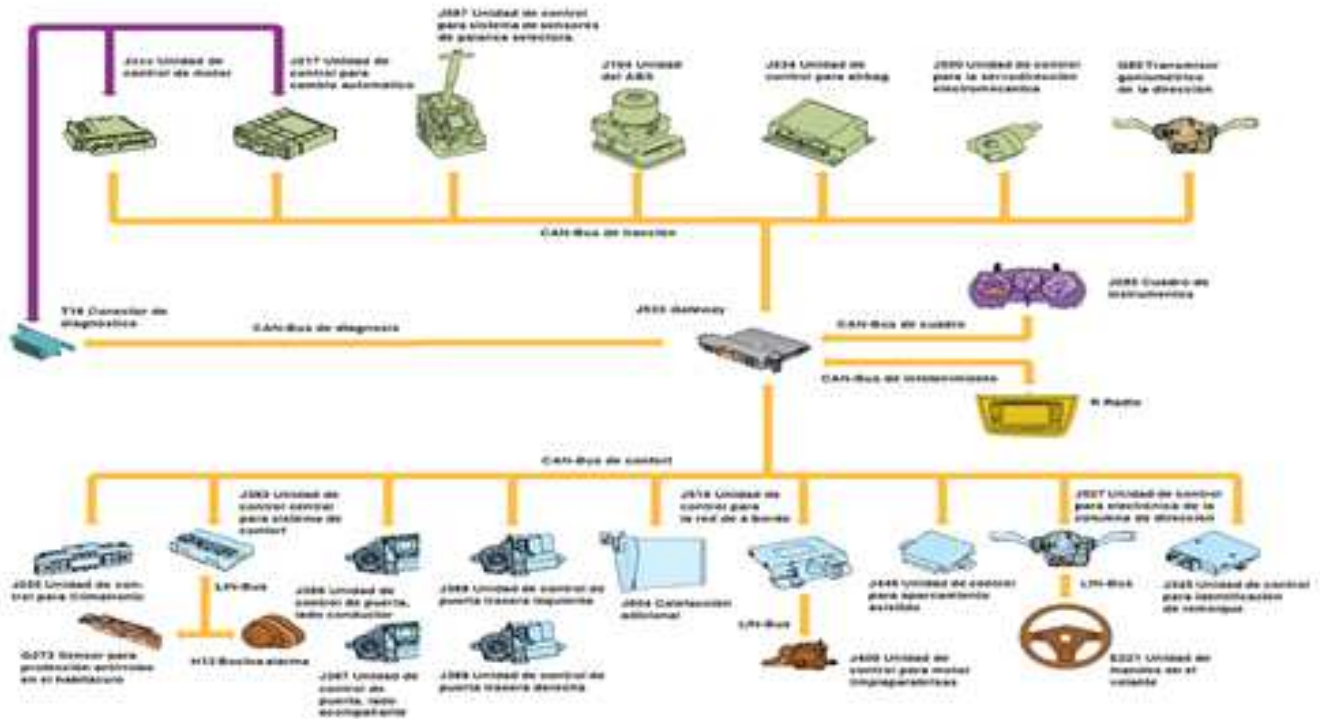


Figura 1.1: Sistemas electrónicos que utilizan el CAN-bus.

La solución la ofrece CAN el protocolo que permite a los fabricantes implementar en forma rápida y eficiente la interoperabilidad de las computadoras. CAN es un puerto de datos de alta velocidad que opera 50 veces más rápido que los protocolos usados en los vehículos anteriormente fabricados (VAN: Vehicle Area Network utilizados por PSA, J1850: Utilizado por FORD, CHRYSLER, GM Y FORD, BEAN: Body Electronic Area Network utilizado por Toyota). Esta velocidad, combinada con los nuevos parámetros definidos para CAN, da a los técnicos la habilidad de ver los datos más rápidamente y tener mayor acierto en el diagnóstico.

1.2. Desarrollo del CAN Bus

En los años 80 Robert Bosch (Alemania) desarrollo este protocolo de comunicación, originalmente para aplicaciones en automoción, pero las compañías automotrices le vieron como un protocolo robusto para aplicación automotriz.

Idea inicial: comunicar entre si varios micro controladores:

- Control de motores
- Control de transmisión automática
- Sistemas de frenos antideslizantes.

En el pasado, los protocolos usados por los fabricantes de autos fueron exclusivos por marca. Ello terminó en 1996 con la aplicación de la norma OBD-II en USA, la cual obligó a los fabricantes a seleccionar entre 4 diferentes Protocolos de Comunicación:

- J1859-PWM
- J1850-VPW
- ISO-9141
- ISO-14230

1.3. Objetivos

1.3.1. Objetivo general

Realizar un manual de pruebas de CAN-Bus de fácil aplicación para análisis e interpretación de fallas de comunicación dentro de los automóviles que trabajan con arquitecturas electrónicas bajo el protocolo de CAN Bus, así como también que sirva de instrumento de enseñanza para las nuevas generaciones que se desarrollan el ámbito de electrónica y lograr su fácil comprensión para su estudio.

1.3.2. Objetivos específicos

- Realizar un manual de pruebas rápidas de CAN-BUS.
- Aplicar este manual para el análisis de fallas en campo.
- Instrumento de enseñanza para las áreas de licenciatura en ciencias de la electrónica, ingeniería automotriz o carreras afines a al desarrollo automotriz.

1.4. Organización de la tesis

Este manual servirá para hacer pruebas de **CAN Bus** a los autos desarrollados bajo este protocolo de comunicación y está organizado de la siguiente forma: en el capítulo 1 se presentan brevemente los antecedentes históricos del **CAN Bus**, así como su definición y cuando fue desarrollado, mientras que los conceptos básicos y definiciones de las características del **CAN Bus** son presentados en el capítulo 2; posteriormente en el capítulo 3 se describe la manera de hacer la prueba de la etapa física del **CAN Bus**; el capítulo 4 describe la prueba del protocolo de comunicación para las unidades de control; finalmente el capítulo 5 es enfocado a la etapa de diagnóstico de las unidades de control misma que se hace mediante **CAN Bus**. En el Capítulo 1 se presentan los antecedentes, la motivación y las contribuciones de este trabajo.

Capítulo 2

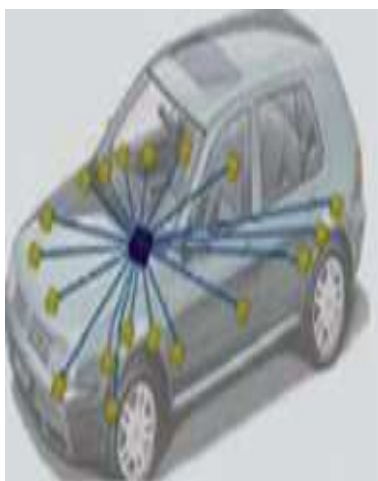
Generalidades del CAN Bus

En este capítulo se aborda de manera general cada uno de los conceptos relacionados con el protocolo CAN Bus.

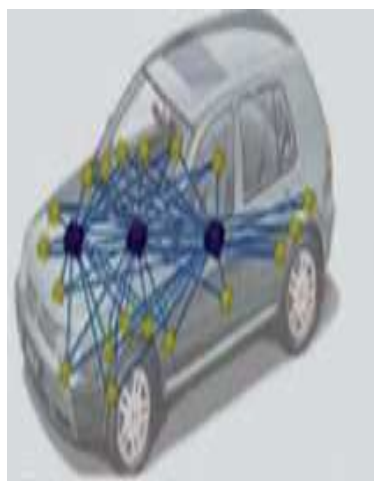
2.1. Introducción

En los sistemas antiguos, el hecho de comunicar y distribuir todas las señales necesarias para hacer funcionar todos estos sistemas implica un amplio y complejo cableado en todo el vehículo.

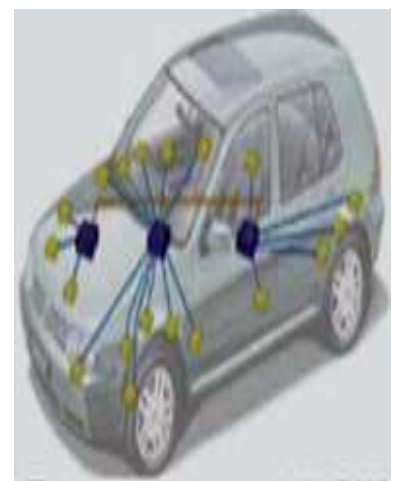
La instalación de los cables necesarios para conectar cada sistema con la **central de control** incrementaría los **problemas de flexibilidad en el mazo de cables**, pudiendo dar lugar a posibles cortes o daños muy difíciles de detectar y reparar; se podrían crear **interferencias** con partes de la carrocería u otras piezas del interior del habitáculo; así como un aumento del peso del vehículo lo que supone un peso de 50 Kilogramos equivalente a 2 Kilómetros de longitud lineales de cables, lo que incrementa la dificultad del mantenimiento.



(a) *Una central conectada*



(b) *3 centrales conectadas*



(c) *3 centrales con CAN Bus*

Figura 2.1: Centrales de control en un Automóvil.

Día con día crecen las exigencias en el automóvil con respecto a la contaminación, el confort de marcha, la seguridad en la conducción y el consumo de combustible. Estas exigencias obligan a un intercambio de información cada vez mayor entre las distintas unidades de mando. La solución a este problema fue el **CAN Bus** o Multiplexado en los automóviles.

El **CAN Bus** abarata considerablemente los costos de producción de soluciones electrónicas, ya que se eliminan sensores redundantes, cableado, etcétera; y se consiguen respuestas en tiempo real de pocos milisegundos. De ésta manera este protocolo le brinda a los técnicos la habilidad de ver los datos rápidamente y tener mayor acierto en el diagnóstico de alguna falla en el automóvil.

La función del protocolo es interconectar las diferentes unidades de control entre sí, de forma que pueden intercambiar gran cantidad de información en un tiempo muy reducido. Al igual que un autobús puede transportar un gran número de personas, el **CAN Bus** puede transportar una gran cantidad de información.

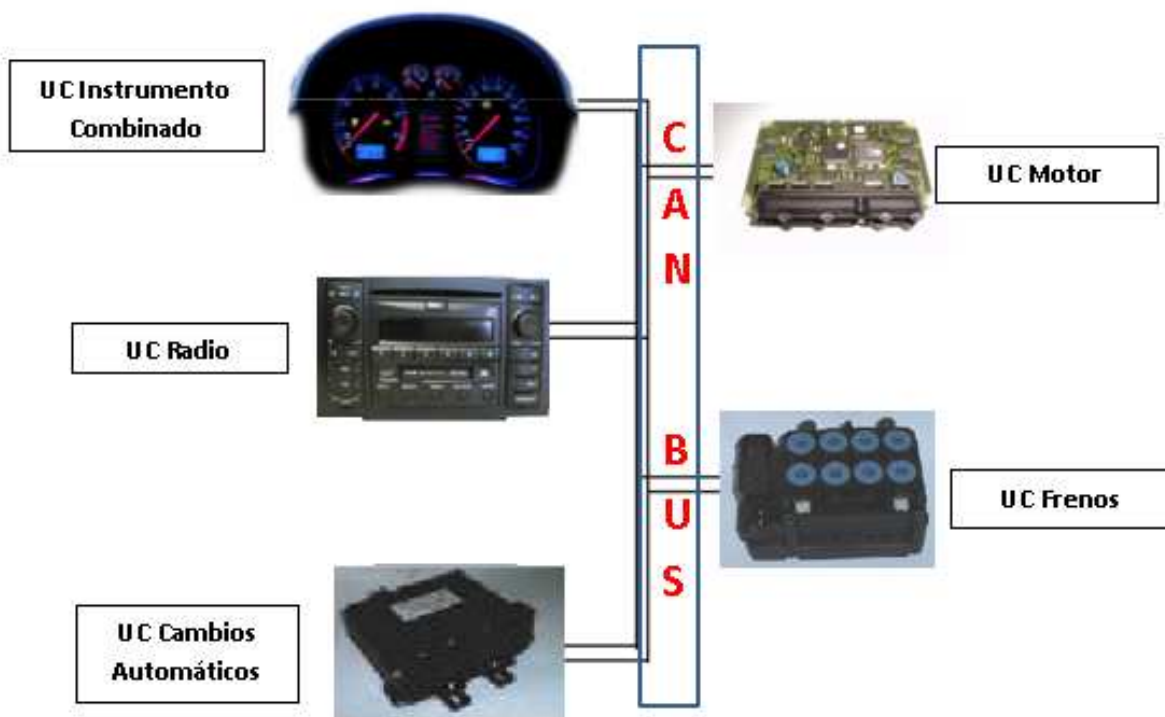


Figura 2.2: Esquema de Bus de comunicación con 5 unidades de control.

Esto permitió la llegada de la nueva generación de instrumentación con soluciones de tipo tecnológico como lo son:

- S1. Faros que se encienden solos.
- S2. Limpiaparabrisas que gradúan su velocidad en función de la lluvia.
- S3. Funciones que regulan el audio dependiendo de la velocidad.

S4. Unidades de telemática para acceso remoto.

S5. Encendidos remoto del automóvil.

S6. Localizadores satelitales.

No obstante, el uso del **CAN Bus** en el automóvil ha sido tan eficiente que no solo existe un solo bus de datos, sino hasta 6 buses de comunicación divididos por funciones específicas para las distintas arquitecturas electrónicas.

2.2. Sistema OBD-II

OBD (On Board Diagnostics) es un sistema de diagnóstico a bordo para automóviles y camiones; se emplean los estándares **OBD-II** (Estados Unidos), **EOBD** (Europa) y **JOBD** (Japón), los cuales permiten el monitoreo y control completo del motor y otros dispositivos del vehículo, detectan, almacenan, y evalúan fallos, guardan códigos de error (**DTC-Diagnostic Trouble Code**), por mencionar las más importantes.

En principio las fallas y sensores que se monitorean están relacionados con las emisiones contaminantes. El uso más importante que se le da a los DTC es diagnosticar un problema, sin necesidad de desmontar partes del automóvil; proceso que resultaría laborioso en la actualidad por la gran cantidad de sistemas mecánicos y electrónicos incluidos en los automóviles. Estos códigos se obtienen desde el sistema a través del Conector de datos (**DLC-Data Link Connector**).

Al sistema **OBD-II** llegan las señales de todos los sensores y salen señales hacia los actuadores repartidos por todo el vehículo, todo esto puede ser controlado por una o varias **ECU (Electrónica Control Unit)** que se comunican entre sí y con los dispositivos electrónicos a través de cableado. Cuando ocurre una falla el sistema **OBD-II** debe informar al conductor por medio de un testigo, en este caso, la **Lámpara de mal-funcionamiento (MIL - Malfunction Indicator Lamp)**.

OBD-II es una mejora del **OBD**, dicho sistema surgió por mandato del gobierno de California para controlar las emisiones contaminantes de los automóviles. Después de creada la **CARB (California Air Resources Board)** se comenzó a legislar el **OBD-I**, sistema que se empezó a utilizar en automóviles vendidos en California en 1988. Esta primera versión del sistema monitorea: el sistema de medición del combustible, el sistema **EGR (Exhaust Gas Recirculation)** y otras mediciones relacionadas con componentes eléctricos. Con las revisiones al decreto sobre aire limpio de 1990, la **CARB** desarrolló nuevas regulaciones para establecer la segunda generación de **OBD**, el **OBD-II**. Entonces, a partir de 1996, todos los camiones, camionetas y automóviles vendidos en Estados Unidos debían tener implementados el sistema **OBD-II**. Esta segunda generación de diagnóstico está diseñada para detectar fallas eléctricas, químicas y mecánicas, y se especializa en el monitoreo del motor, en el sistema **ABS (Anti-lock Break System)**, entre otros sistemas; de manera general el **OBD-II** se dedica a vigilar continuamente los componentes que intervienen en las emisiones contaminantes. Cabe mencionar que el funcionamiento de este sistema difiere en cierta medida en vehículos con motor a gasolina con respecto a los que usan Diesel.

Para saber si un vehículo determinado cuenta con el sistema **OBD-II**, podemos encontrar en el vehículo una etiqueta que lo indique, como la siguiente:



Figura 2.3: Certificado **OBD-II**.

Las funciones principales del sistema **OBD-II** en vehículos a gasolina o Diesel son:

- En vehículos con motor a gasolina:
 - Vigila el rendimiento del catalizador.
 - Diagnóstico del envejecimiento de las sondas lambda (sensores de oxígeno).
 - Prueba de voltaje en las sondas lambda.
 - Sistema de aire secundario (si el vehículo lo incorpora).
 - Sistema de recuperación de vapores de combustible del sistema de control de evaporaciones de gases (EVAP o cánister).
 - Prueba de diagnóstico de fugas.
 - Sistema de alimentación de combustible.
 - Fallos de combustión.
 - Funcionamiento del sistema de comunicación entre ECUs (Engine Control Unit) o PCM (Powertrain Control Module).
 - Control del sistema de gestión electrónica.
 - Sensores y actuadores del sistema electrónico que intervienen en la gestión del motor o que están relacionados con las emisiones contaminantes.
- En los motores Diesel:
 - Fallos de la combustión.
 - Regulación del comienzo de la inyección.
 - Regulación de la presión de sobrealimentación.

- Recirculación de gases de escape.
- Funcionamiento del sistema de comunicación entre ECUs.
- Control del sistema de gestión electrónica.

El sistema **OBD-II**, además de las mejoras añadidas, estandarizó los códigos de error para las diferentes funciones del vehículo y para todos los fabricantes; también permite acceder a esos códigos con equipos de diagnóstico universales, por lo tanto el conector de acceso al **OBD-II** está también normalizado. El conector para el sistema **OBD-II**, figura 2.2 debe ser accesible y estar en la zona del conductor (incluye el lado del copiloto). Además, debe tener asignado en sus terminales, de manera estandarizada, los protocolos por los cuales se accede al sistema **OBD-II**.

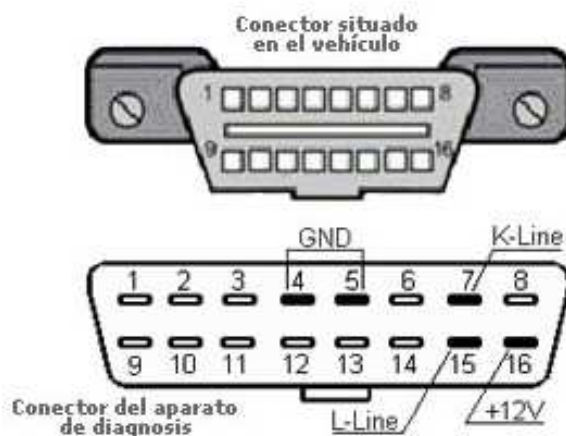


Figura 2.4: Conector **OBD-II** y sus terminales.

Es importante mencionar que antes del sistema **OBD-II**, cada fabricante establecía sus propios códigos, con el inconveniente de que se necesitaba un equipo para cada modelo de automóvil y un compendio diferente de códigos con su significado.

La tabla 2.1 enlista los protocolos presentes en cada terminal del conector.

Terminal	Uso	Nivel [V]
1	Reservado	—
2	J1850 PWM y VPW (+)	5 y 7
3	Reservado	—
4	Negativo de la batería	0
5	Tierra de la señal	0
6	CAN (high)	3.5
7	ISO 9141-2 e ISO 14230-4 (Línea K)	12
8	Reservado	—
9	Reservado	—
10	J1850 PWM (-)	0

Terminal	Uso	Nivel [V]
11	Reservado	—
12	Reservado	—
13	Reservado	—
14	CAN (low)	1.5
15	ISO 9141-2 e ISO 14230-4 (Línea L)	0
16	Positivo de la Batería	

Tabla 2.1: Terminales del conector **OBD-II**.

En el sistema **OBD-II** cuenta con cinco protocolos de comunicación, y un automóvil normalmente usa sólo uno de ellos, estos son:

- **J1850 PWM (Pulse Width Modulation)**, usado por Ford y Mazda
- **J1850 VPW (Variable Width Modulation)** usado por General Motors y camiones ligeros
- **ISO9141-2** el protocolo más viejo en vehículos Chrysler, europeos y asiáticos, entre 2000-2004
- **ISO14230-4 KWP2000 (Keyword Protocol 2000)** comúnmente usado en vehículos del 2003
- **ISO15765-4 CAN-Bus** primero introducido en 2004 y obligatorio en Estados Unidos en vehículos desde 2008.

2.2.1. Modos de funcionamiento del sistema **OBD-II**

El sistema **OBD-II** tiene varios modos de funcionamiento, también estandarizados, lo que significa que el modo con el que se acceda al sistema no dependerá del fabricante del vehículo. El total de modos de funcionamiento son nueve, aunque los fabricantes no están obligados a incluir todos, son:

Modo 1. Muestra datos en tiempo real. Identificación de Parámetro (PID - Parameter Identification), es el acceso a datos en tiempo real de valores analógicos o digitales de salidas y entradas a la ECU. Este modo es también llamado flujo de datos. Aquí es posible ver, por ejemplo, la temperatura de motor o el voltaje generado por una sonda lambda. El proyecto del que trata el presente trabajo se centra en este modo de funcionamiento.

Modo 2. Muestra los datos guardados en memoria (Freeze Frame). Los PID son los mismos que en el **Modo 1**, con el mismo significado, pero el acceso es al cuadro de datos congelados, con los valores de los sensores al momento de ocurrir una falla. Esta es una función muy útil del **OBD-II**, porque al recuperar estos datos, se pueden conocer las condiciones exactas en las que ocurrió dicho fallo. El cuadro de datos corresponde al primer fallo detectado.

Modo 3. Petición de códigos de error o fallo. Este modo permite extraer de la memoria de la ECU todos los códigos de fallo (DTC - Data Trouble Dode) almacenados. En talleres de mantenimiento y reparación de vehículos los escáneres trabajan principalmente con este modo, para detectar la posible zona de la falla.

Modo 4. Borrado de DTCs y valores almacenados. Con este modo se pueden borrar todos los códigos almacenados en la PCM, incluyendo los DTCs y el cuadro de datos grabados.

Modo 5. Solicitud de los diagnósticos de los sensores de oxígeno. El propósito de este modo es permitir el acceso a los resultados de las mediciones de voltaje hechas a los sensores de oxígeno del módulo de control del motor, y así determinar el funcionamiento de los mismos y la eficiencia del convertidor catalítico. La información de este modo no está disponible en vehículos que utilizan el sistema CAN. Para esos vehículos se requiere utilizar el modo 6.

Modo 6. Diagnóstico en el sistema OBD-II. El propósito de este modo es permitir el acceso a los resultados de las pruebas hechas a componentes que no están monitoreados continuamente, pero sólo en periodos de tiempo y condiciones de operación determinadas, estos resultados reportan rangos de valores máximos y mínimos. Este modo también sirve para verificar que todos los sensores y otros componentes de control de emisiones contaminantes estén funcionando adecuadamente. También muestra las pruebas hechas a los sensores de oxígeno de la misma manera que en el modo 5 pero sólo para CAN.

Modo 7. Diagnóstico de DTCs pendientes. Este modo permite leer de la memoria de la ECU todos los DTCs pendientes.

Modo 8. Prueba a bordo. Este modo nos permite realizar un control de operación a bordo de componentes y/o sistemas del vehículo, así, este modo permite realizar la prueba de actuadores, con lo que el mecánico puede activar y desactivar actuadores como bombas de combustible, válvula de ralentí, etcétera

Modo 9. Información del vehículo. Solicita información del vehículo como el VIN (Vehicle Identification Number). Este número es una secuencia de dígitos que identifica los vehículos de motor de cualquier tipo, y los remolques a partir de un cierto peso, es un código específico y único para cada unidad fabricada.

Para comunicarnos con el sistema **OBD-II** se debe utilizar alguno de sus nueve modos de funcionamiento. La información de petición de datos se envía dentro de una trama del protocolo que se use, esta trama incluye, el modo de funcionamiento con el cual se trabajará, así como cuál es la información que se requiere. Con la finalidad de especificar el dato que se pide se hace uso de los PID's, algún dispositivo del sistema **OBD-II** conectado en el bus reconoce ese PID como uno al que tiene que responder, y reporta el valor para ese PID por medio del mismo bus; donde el número de bytes enviados y leídos tanto como su codificación dependen del dato. La mayoría de los datos están codificados con una fórmula, otros están codificados de forma especial.

Aunque brindo muchas ventajas, el uso de 4 diferentes protocolos continúan ocasionado complicaciones en los procesos de inspección, mantenimiento y reparación. Desde 1992 se utiliza en coches, y la mayoría de las compañías de automoción europeas lo utilizan. Tal es el caso de Mercedes-Benz y otros han incluido redes CAN en sus vehículos para manejar la comunicación entre Controladores. El Protocolo CAN fue integrado a la normatividad **OBD-II** por un comité de la **Organización Internacional de Normalización (ISO)** y lo aceptaron como el Protocolo de Diagnóstico estandarizado para los Fabricantes. Adicionalmente, el **Buró de Calidad del Aire de California (CARB)** aceptó el Protocolo **CAN** y lo promovió obligatorio para todos los vehículos.

Los fabricantes de autos empezaron implementando **CAN** en el año 2003 y continuaron aplicándolo a todos sus vehículos, para una aplicación total en el año 2008.

La normatividad entorno a **CAN** también es obligatoria en los nuevos programas estatales de Emisiones, Inspección y Mantenimiento.

La industria automovilística ocupa el 90 % de la utilización de **CAN**. Pero en los últimos años se ha extendido a muchos otros campos como lo son:

- Sistemas de Control y monitoreo en la industria.
- Sistemas de monitoreo de la red eléctrica.
- Monitoreo de maquinaria para producción en serie.

2.3. El protocolo CAN

Desde principios de los 40's, los fabricantes continuamente han mejorado la tecnología de sus automóviles, integrando e incrementando la cantidad de sistemas electrónicos en ellos. Dichos sistemas tienen que comunicarse entre sí y pueden ser módulos de control, sensores, actuadores, etcétera. En forma general llamaremos **nodo** a cualquier sistema conectado al **bus CAN**.

La comunicación se establece de manera paralela o en serie; en un principio la comunicación se hacía punto a punto, lo que implicaba que al menos hubieran dos cables hacia cada lugar donde se requería comunicar.

Conforme la tecnología ha ido progresando, los vehículos se han vuelto más complejos; con el reemplazo de sistemas mecánicos por componentes electrónicos y al proporcionarle comodidades adicionales al usuario, conveniencia y características de seguridad se ha ido incrementando el cableado. Tener una cantidad muy grande de cableado tiene muchos problemas, como el incremento del peso del vehículo, menor maniobrabilidad, mayor interferencia, necesidad de más espacio, más propenso a averías, etcétera.

En la actualidad las señales que se transmiten y procesan son digitales, entonces una solución es utilizar comunicación serial, con un único bus de comunicación, que sea resistente al ruido.

El **bus CAN** es un bus automotriz desarrollado en 1980 por Robert Bosch. Es un estándar que rápidamente ha ganado una amplia aceptación dentro de la industria automotriz y aeroespacial. El **protocolo CAN** es un protocolo de bus serial para conectar sistemas y sensores individuales, como alternativa al cableado múltiple, esto permite a los componentes automotrices comunicarse sobre un bus de datos en red de uno o dos cables de hasta 1 Mbps. Gracias a este bus la cantidad de cableado necesario para interconectar todos los sistemas se redujo enormemente.

En la figura 2.5 vemos con bloques un ejemplo de conexión sin bus CAN.

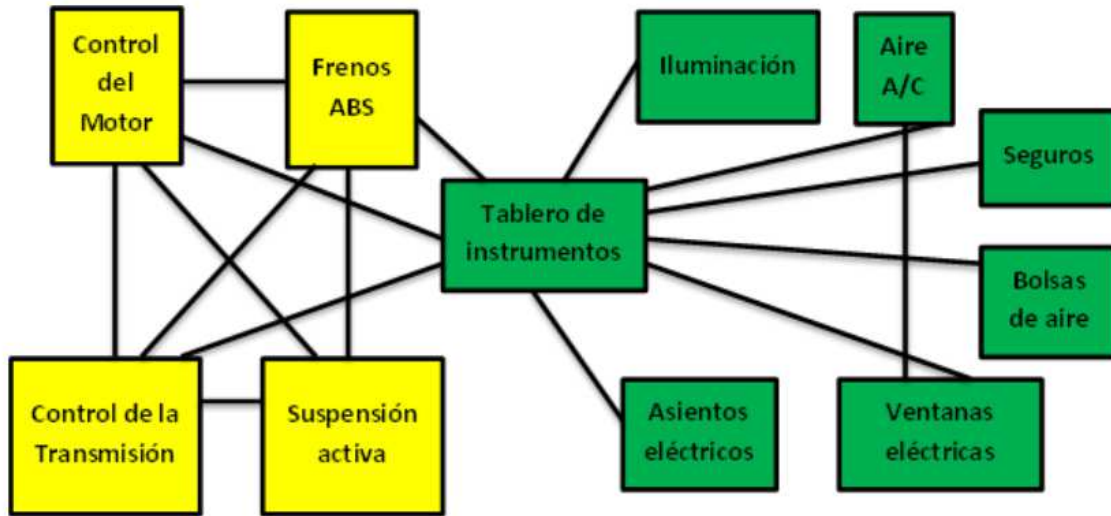


Figura 2.5: Conexión sin bus CAN.

En la figura 2.6 presentamos un ejemplo de conexión con CAN.

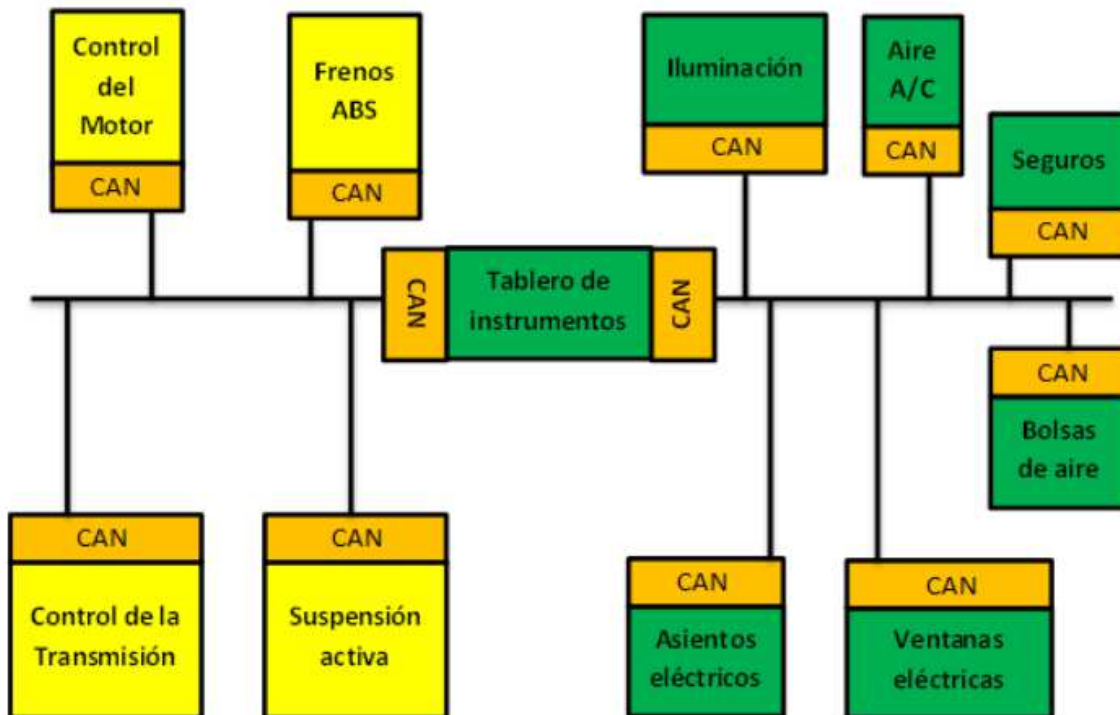


Figura 2.6: Conexión con bus CAN.

Al comparar estas dos imágenes nos damos idea de la gran importancia del uso del protocolo CAN, requerido de forma obligatoria en vehículos a partir del 2008 en Estados Unidos.

Las principales razones por las que se utiliza el **protocolo CAN** son:

- Es un estándar maduro o **CAN** es un protocolo usado por más de 20 años
- Existen numerosos productos y herramientas **CAN** en el mercado
- Implementación de hardware del protocolo
- Combinación del manejo de error y aislamiento de fallas con una alta velocidad de transmisión
- Medio de transmisión simple
- El estándar es un cable de par trenzado, pero incluso sólo con un alambre puede funcionar
- Puede trabajar con otros medios: enlaces ópticos o de radio
- Excelente manejo de error
- Mecanismo de detección de errores con CRC
- Aislamiento de fallas
- Característica incluida para prevenir que un nodo que falle bloquee al sistema
- Protocolo más usado en el mundo industrial y automotriz
- Buena relación precio-desempeño

Algunas de las desventajas que puede presentar el **bus CAN** son un mayor coste y la necesidad de conocimientos especializados a la hora del mantenimiento y reparación del vehículo.

Es importante señalar que el protocolo **CAN Bus** define su manera de trabajar sobre las primeras capas del modelo **OSI (Open System Interconnection)**; siendo éstas la capa de enlace de datos y parte de la capa física.

Estas capas están estandarizadas conforme el estándar **ISO-11898**, para velocidades de hasta 1 Mbps, denominado **CAN de alta velocidad (High-Speed CAN)** y el estándar **ISO-11519**, para velocidades de hasta 125 kbps, denominado **CAN de baja velocidad (Low-Speed CAN)** también llamado **tolerante a fallas o (Fault Tolerant Low-Speed CAN)**.

La información que circula entre las unidades de mando a través de los dos cables (bus) son paquetes de 0 y 1 (bit) con una longitud limitada y con una estructura definida de campos que conforman el mensaje.

Uno de esos campos actúa de identificador del tipo de dato que se transporta, de la unidad de mando que lo trasmite y de la prioridad para transmitirlo respecto a otros.

El mensaje no va direccionado a ninguna unidad de mando en concreto, cada una de ellas reconocerá mediante este identificador si el mensaje le interesa o no.

Todas las unidades de mando pueden ser transmisoras y receptoras, y la cantidad de las mismas abonadas al sistema puede ser variable (dentro de unos límites). Si la situación lo exige, una unidad de mando puede solicitar a otra una determinada información mediante uno de los campos del mensaje (trama remota o RDR).

Cualquier unidad de mando introduce un mensaje en el bus con la condición de que esté libre, si otra lo intenta al mismo tiempo el conflicto se resuelve por la prioridad del mensaje indicado por el identificador del mismo.

El sistema está dotado de una serie de mecanismos que aseguran que el mensaje es transmitido y recibido correctamente. Cuando un mensaje presenta un error, es anulado y vuelto a transmitir de forma correcta, de la misma forma una unidad de mando con problemas avisa a las demás mediante el propio mensaje, si la situación es irreversible, dicha unidad de mando queda fuera de servicio, pero el sistema sigue funcionando.

Selección de Mensajes

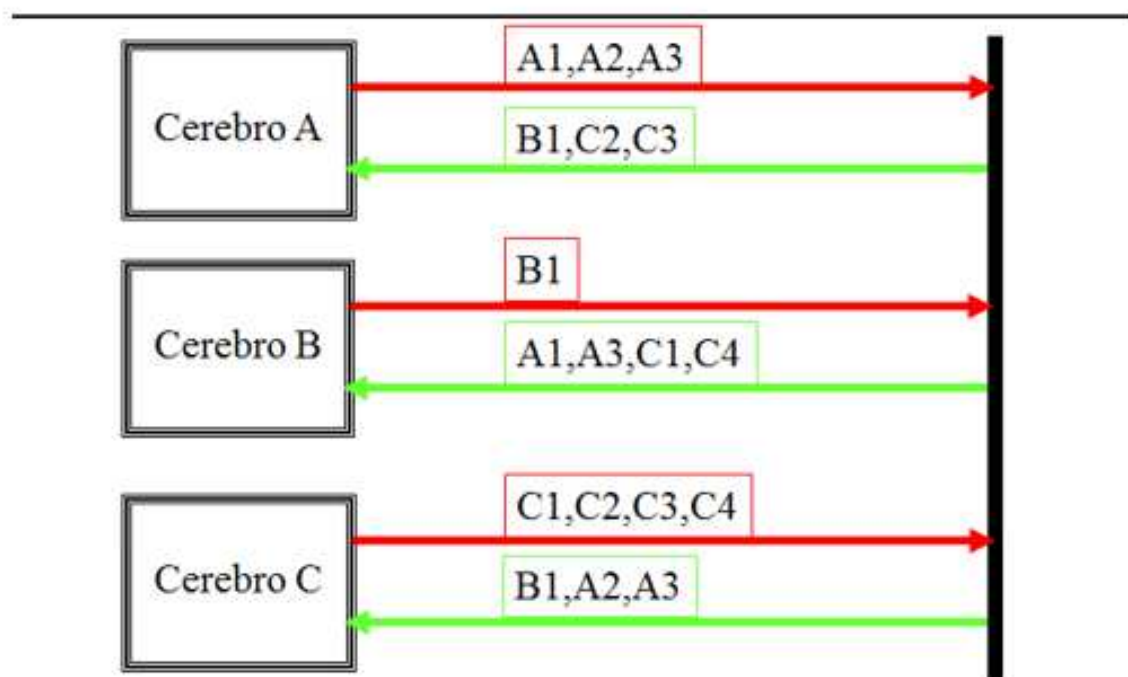


Figura 2.7: Esquema de selección de mensaje de las unidades de control.

2.4. Principales características de CAN-Bus

CAN se basa en el modelo productor/consumidor, el cual es un concepto, o paradigma de comunicaciones de datos, que describe una relación entre un productor y uno o más consumidores. CAN es un protocolo orientado a mensajes, es decir la información que se va a intercambiar se descompone en mensajes, a los cuales se les asigna un identificador y se encapsulan en tramas para su transmisión. Cada mensaje tiene un identificador único dentro de la red, con el cual los nodos deciden aceptar o no dicho mensaje. Dentro de sus principales características se encuentran:

- Prioridad de mensajes.
- Garantía de tiempos de latencia.
- Flexibilidad en la conexión.
- Recepción por multidifusión (multicast) con sincronización de tiempos.
- Sistema robusto en cuanto a consistencia de datos.
- Sistema multimaestro.
- Detección y señalización de errores.
- Retransmisión automática de tramas erróneas
- Distinción entre errores temporales y fallas permanentes de los nodos de la red, y desconexión autónoma de nodos defectuosos.

CAN fue desarrollado, inicialmente para aplicaciones en los automóviles y por lo tanto la plataforma del protocolo es resultado de las necesidades existentes en el área de la automoción. La Organización Internacional para la Estandarización (ISO, International Organization for Standardization) define dos tipos de redes CAN: una red de alta velocidad (hasta 1 Mbit/s), bajo el estándar **ISO 11898-2** destinada para controlar el motor e interconectar las **Unidades de Control Electrónico (ECU)**; y una red de baja velocidad tolerante a fallos (menor o igual a 125 kbit/s), bajo el estándar **ISO 11519-2/ISO 11898-3** dedicada a la comunicación de los dispositivos electrónicos internos de un automóvil como son control de puertas, techo corredizo, luces y asientos.

Las características principales de la operación de **CAN** en las capas mencionadas se describen a continuación:

2.4.1. Capa de enlace

Controla el enlace lógico de los nodos del bus, esta capa se encarga de los filtros con los cuales los nodos deciden qué mensajes recibidos se aceptan. Proporciona servicios durante la transferencia y petición de datos entre los nodos y proporciona medios para el restablecimiento y para notificar la sobrecarga del bus. Esta capa también controla el acceso al medio y representa el núcleo del **protocolo CAN**; presenta los mensajes recibidos al control de enlace lógico, también es responsable de la trama de mensajes, arbitraje, reconocimiento, señalización y detección de error. El control de acceso al medio decide si el bus está libre para comenzar una nueva transmisión o si la recepción acaba de comenzar.

2.4.2. Capa física

La capa física en **CAN** es responsable de la transferencia de bits entre los distintos nodos que componen la red. Define aspectos como niveles de señal, codificación, sincronización y tiempos en que los bits se transfieren al bus. En la especificación original de **CAN** la capa física no fue definida, permitiendo con ello diferentes opciones para la elección del medio y niveles eléctricos de transmisión. Las características de las señales eléctricas en el bus fueron establecidas más tarde por el estándar **ISO-11898**. La especificación **CiA (CAN in AUTOMATION)**, complementó las definiciones respecto al medio físico y conectores. El resto de la capa física (y para las demás capas del modelo OSI) no están especificadas para el **protocolo CAN** y pueden ser definidas por el diseñador del sistema. Como se ha mencionado que **CAN** es un protocolo de comunicación serial, está basado en una topología de bus para la transmisión de mensajes en ambientes distribuidos, es para aplicaciones de control en tiempo real, con una velocidad de comunicación de hasta 1 Mbit por segundo, y tiene excelente capacidad de detección y aislamiento de errores; además, ofrece una solución a la gestión de la comunicación entre múltiples **CPUs (unidades centrales de proceso)** como las **ECU** y **PCM**. El bus de comunicación de **CAN** consta de dos cables de par trenzado (aunque pueden ser otros medios), en los que viaja una señal diferencial, estas dos características hacen que la comunicación sea en gran medida inmune al ruido, tomando en cuenta que está sometido al ruido proveniente de la temperatura del motor, de la chispa de ignición y de todos los demás dispositivos con los que cuenta el automóvil. Todos los dispositivos que se comunican están conectados en estos dos cables, llamados respectivamente **CAN-High** (línea **H**) y **CAN-Low** (línea **L**), y todos los dispositivos conectados deben comunicarse a la misma velocidad. Estos cables o bus tiene una impedancia característica de 120 Ohms, por lo que debe tener una resistencia de terminación del mismo valor en cada extremo para evitar que se refleje la señal. El **protocolo CAN** es multi-maestro, esto permite que todos los dispositivos conectados tengan la misma oportunidad de acceder al bus para enviar mensajes. La comunicación se lleva a cabo por un **mecanismo de difusión**, el cual se basa en un protocolo de transmisión orientado a mensajes, esto quiere decir que define el contenido del mensaje en lugar de definir al nodo o a la dirección del nodo. Este mecanismo de comunicación permite que el acceso al bus se realice por prioridad del mensaje utilizando **CSMA/CR (Carrier Sense Multiple Access/Collision Resolution)**. **CSMA** significa que los nodos tienen la misma oportunidad de enviar mensajes (MA) y también que todos están escuchando los mensajes del bus (CS). En un determinado momento más de un nodo tratará de acceder al bus, y los mensajes colisionan, esta colisión se soluciona por arbitraje basado en prioridades (CR). Este método se basa en una topología eléctrica que aplica una función lógica determinista a cada bit; la prioridad se le da a un nivel de bit de tipo dominante. Se define como bit dominante al nivel lógico **0**, este bit fuerza su valor en el bus sin importar el valor de los demás bits; el valor lógico **1** es un bit recesivo, este valor permanece si todos los bits tienen este valor. Para la solución a una colisión de mensajes se aplica una **función AND** a todos los bits transmitidos simultáneamente. Cada transmisor escucha continuamente el valor presente en el bus; el transmisor se retira cuando el valor en el bus es dominante y éste ha enviado un bit recesivo. Mientras el valor del bus y el enviado coincidan, el transmisor permanecerá; finalmente el mensaje con identificador de máxima prioridad sobrevive, y los demás nodos reintentarán la transmisión lo antes posible. En concreto, la **resolución de colisión (CR)** hace que la comunicación sea no-destructiva y se garantiza que el mensaje que tenga la mayor prioridad no se detenga y se transmite sin perder tiempo.

La prioridad del mensaje se establece dentro de la misma trama que contiene al mensaje, en una sección dentro del campo **arbitraje** llamada identificador. Cada nodo, con base en el identificador, debe decidir si el mensaje debe atenderse o ignorarse.

2.4.3. Codificación

El **protocolo CAN** utiliza una codificación llamada **NRZ (Non Return to Zero)**, en la cual cada bit está representado por un solo nivel, esto quiere decir que el nivel de la señal permanece constante a lo largo del tiempo que dura un bit; un nivel lógico **0** se representa con un nivel de voltaje bajo, y un nivel lógico **1** se representa con un nivel de voltaje alto. Usar la **codificación NRZ** asegura mensajes compactos con un número mínimo de transiciones y una resistencia alta a perturbaciones externas. Un detalle de la **codificación NRZ** es que el nivel de la señal puede permanecer constante por un periodo de tiempo más largo; por lo tanto se deben tomar medidas para asegurar que no se exceda el intervalo máximo permisible entre dos flancos de la señal, esto es importante para propósitos de sincronización. En la figura 2.8 se muestra un ejemplo de la **codificación NRZ**, donde se ve que el nivel permanece constante al enviarse varios bits seguidos del mismo valor.



Figura 2.8: Ejemplo de **codificación NRZ**.

Dado que cada nodo conectado al bus tiene su propio oscilador pueden presentarse desfases entre ellos. Todos los nodos se sincronizan cuando alguno comienza la transmisión y se re-sincronizan con cada cambio de nivel, pero, debido a la característica de la **codificación NRZ**, cuando el nivel permanece constante mucho tiempo se aplica el relleno de bits (Bit Stuffing), insertando bits complementarios después de cinco bits del mismo valor; el receptor tiene que quitar estos bits de relleno para poder procesar el mensaje original. En la figura 2.9 podemos ver el relleno con bits.

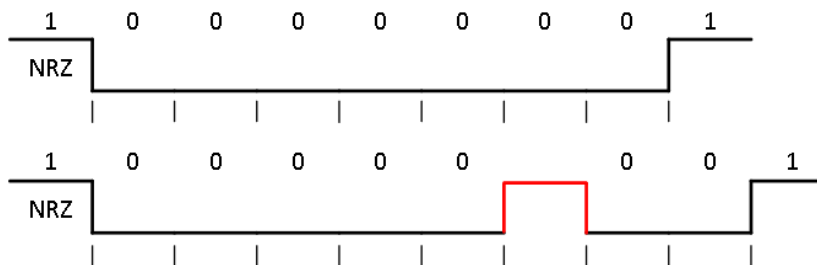


Figura 2.9: Datos sin relleno (arriba) y datos con relleno (abajo).

2.4.4. Niveles eléctricos del bus

El controlador **CAN** es el elemento encargado de la comunicación entre el microprocesador de la ECU y el transmisor-receptor **CAN**, este controlador trabaja acondicionando la información que entra y sale entre ambos componentes, y también es el que determina la velocidad de transmisión de los mensajes; además interviene en la necesaria sincronización entre los diferentes nodos para la correcta emisión y recepción de los mensajes. El transceptor **CAN** es el elemento que tiene la misión de recibir y de transmitir los datos, además de acondicionar y preparar la información para que pueda ser utilizada por los controladores; esta preparación consiste en situar los niveles de tensión de forma adecuada, amplificando la señal cuando la información se vuelca en la línea y reduciéndola cuando es recogida de la misma y suministrada al controlador; el transceptor en ningún caso interviene modificando el contenido del mensaje. Funcionalmente el transceptor está situado entre los cables que forman la línea **bus CAN** y el controlador. Hemos definido al bit **dominante (0)** y al bit **recesivo (1)**, estos bits tienen niveles TTL (5 V para alto ó **1** y 0 V para bajo ó **0**) y están presentes a la salida del controlador **CAN** hacia el transceptor **CAN**, en las líneas de Transmisión (Tx) y Recepción (Rx) correspondientes. Cuando se presenta un estado recesivo el voltaje nominal en las líneas **H** y **L** es de 2.5 V para cada una de ellas, lo que arroja un voltaje diferencial nominal de 0 V; para un estado dominante el voltaje nominal en **H** es de 3.5 V y para **L** es de 1.5 V, esto arroja un voltaje diferencial nominal de 2 V. Se ha de tomar en cuenta que la especificación **CAN** de Bosh no establece cómo se han de traducir estos niveles a variable física, (entendiéndose con esto si el bus es óptico, eléctrico, electromagnético, etcétera). Cuando se utiliza el par trenzado, según **ISO 11898**, el nivel dominante es un voltaje diferencial positivo en el bus; el nivel recesivo es ausencia de voltaje, o cierto valor negativo. En términos simples, los valores dominante y recesivo, descritos anteriormente, están definidos a nivel de protocolo y se presentan eléctricamente, generalmente con niveles TTL, entre el controlador y transceptor **CAN**; entonces, el valor dominante **0** provoca un voltaje diferencial positivo en el bus a través del transceptor (así como voltaje diferencial positivo provoca un valor dominante **0** a través del transceptor), y un valor recesivo **1** provoca un voltaje diferencial cero o negativo en el bus a través del transceptor (así como un voltaje diferencial cero o negativo provoca un valor recesivo **1** a través del transceptor). La figura 2.10 muestra los niveles en el bus antes descritos.

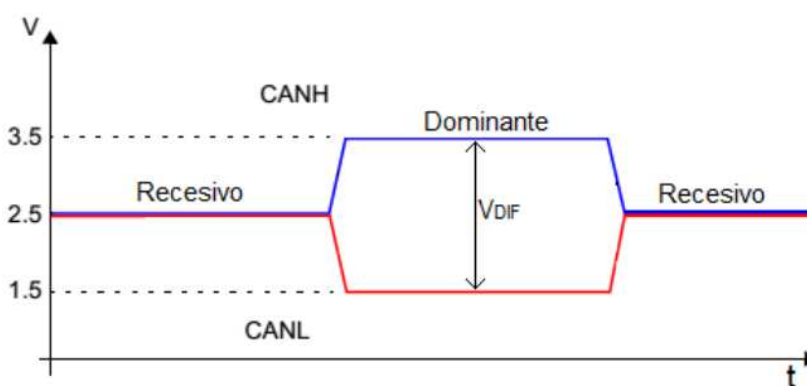


Figura 2.10: Niveles del bus.

2.4.5. Formato de tramas

CAN utiliza mensajes de estructura predefinida, tramas, para la gestión de la comunicación. Se distinguen entre dos variantes de CAN, el definido en **CAN 2.A** o **CAN Estándar** y el definido en **CAN 2.B** o **CAN Extendido**. En estas variantes de **CAN** los formatos de trama son análogos, diferenciándose básicamente en el número de bits que se utilizan para el identificador de mensaje: de 11 bits, dando 2048 identificadores diferentes; sin embargo la especificación dice que los 7 bits más significativos del identificador no pueden ser simultáneamente recesivos, por lo tanto debido a los 4 bits restantes hay 16 combinaciones en los que estos siete bits son todos recesivos y el número de identificadores se reduce a 2032 (2048-24) en **CAN** estándar; y de 29 bits que dan 536.870.912 identificadores para **CAN** Extendido.

Las tramas **CAN** son de longitud reducida, la trama más larga es de 130 bits en **CAN** Estándar y 154 bits en **CAN** Extendido. Los cuatro tipos de trama utilizados en las dos variantes son: trama de datos (Data Frame), lleva datos de un transmisor a los receptores, puede incluir entre 0 y 8 bytes de información útil; trama remota (Remote Frame), es transmitida por algún nodo del bus que requiere la transmisión de una trama de datos con el mismo identificador; trama de error (Error Frame), es transmitida por alguna unidad que detecte algún error; trama de sobrecarga (Overload Frame), permite que un nodo fuerce a los demás a alargar el tiempo entre transmisión de tramas sucesivas.

2.4.5.1. Trama de datos

Una trama de datos, figura 2.11, se compone de siete diferentes campos de bits: Inicio de trama (SOF-Start Of Frame), arbitraje, control, datos, CRC (Cyclic Redundancy Check), reconocimiento, fin de trama.

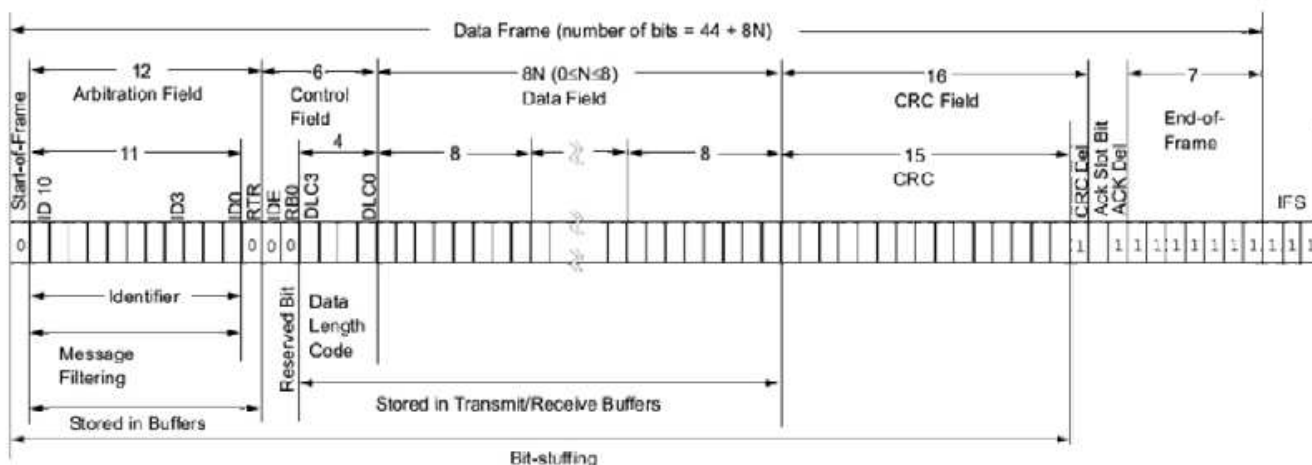


Figura 2.11: Trama de datos estándar.

2.4.5.2. Inicio de trama (SOF)

El inicio de trama es un campo de un solo bit siempre dominante, que indica el inicio de la transmisión. Los nodos receptores se sincronizan con el flanco de bajada de este bit.

2.4.5.3. Arbitraje

El campo de identificación está formado por el identificador de mensaje (11 bits) más el bit RTR (Remote Transmission Request). En una trama de datos el bit RTR es dominante, en una trama remota es recesivo. Los bits de identificador se transmiten en orden de más significativo a menos significativo.

2.4.5.4. Control

El campo de control, mostrado en la figura 2.12, está formado por dos bits reservados para uso futuro y cuatro bits adicionales que indican el número de bytes de datos.

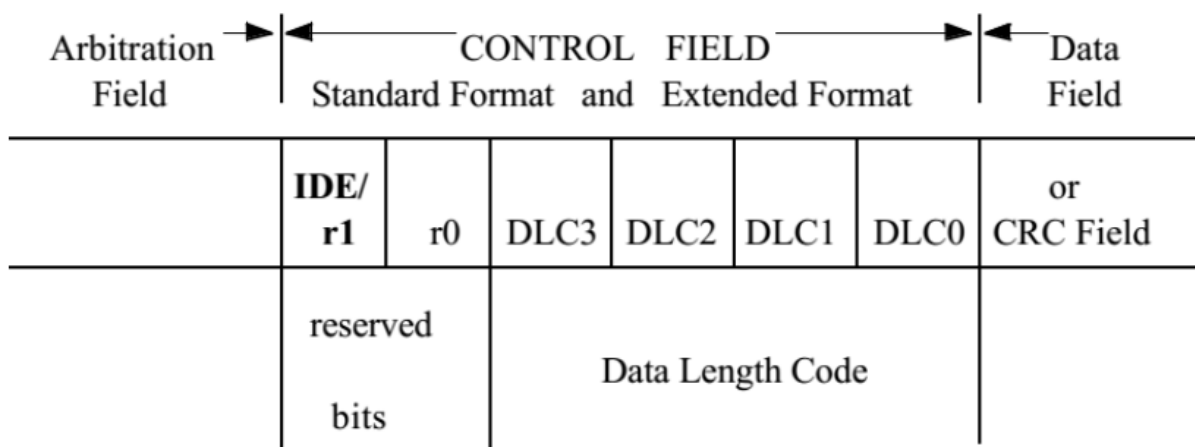


Figura 2.12: Campo de control.

En realidad el primero de estos bits, IDE (Identifier Extension), se utiliza para indicar si la trama es de **CAN** Estándar (IDE dominante) o Extendido (IDE recesivo). El segundo bit, RB0 (Reserved Bit Zero), siempre es recesivo. Los cuatro bits de código de longitud (DLC- Data Length Code) indican en binario el número de bytes de datos en el mensaje (0 a 8, otros valores no pueden ser usados).

La tabla 2.2 muestra el número de bytes de datos que se pueden obtener, cantidad codificada por los bits de código de longitud de datos, donde **D** es para **dominante** y **R** para **recesivo**.

Número de Datos	Código de longitud de datos			
	DLC3	DLC2	DLC1	DLC0
0	D	D	D	D
1	D	D	D	R

2	D	D	R	D
3	D	D	R	R
4	D	R	D	D
5	D	R	D	R
6	D	R	R	D
7	D	R	R	R
8	R	D	D	D

Tabla 2.2: Número de bytes de datos en el DLC. **Dominante D** y **Recesivo R**

2.4.5.5. Datos

Es un campo formado por 0 a 8 bytes de datos, es decir 0 a 64 bits. Cada byte se transmite con el bit más significativo primero.

2.4.5.6. CRC

Código de redundancia cíclica que genera el transmisor por la división módulo 2 de todos los bits precedentes del mensaje, incluyendo los de relleno si existen, por el polinomio generador:

$$x^{15} + x^{14} + x^8 + x^7 + x^4 + x^3 + x^1 + 1, \quad (2.1)$$

el residuo de esta división es el código CRC transmitido. Los receptores comprueban este código. Tras el código CRC se incluye un bit recesivo (delimitador de CRC).

2.4.5.7. Campo de reconocimiento (ACK)

Es un campo de dos bits que el transmisor envía como recesivos. El primero de estos bits es cambiado a un bit dominante por los nodos que han recibido el mensaje correctamente. El bit de ACK queda así insertado entre dos bits dominantes de delimitación.

2.4.5.8. Fin de trama (EOF)

Cierra la trama, consiste en 7 bits recesivos sucesivos.

2.4.5.9. Espaciado entre tramas (IFS)

Consta de un mínimo de 3 bits recesivos.

2.4.5.10. Trama de datos extendida

En la trama de datos extendida en CAN, mostrada en la figura 2.13, al bit SOF le sigue el campo de arbitraje, el cual consta de 32 bits.

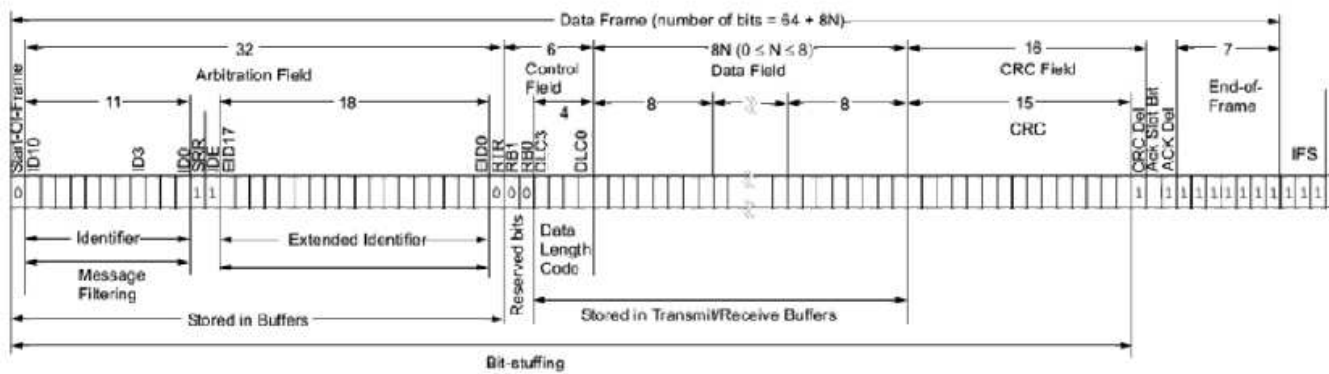


Figura 2.13: Trama de datos extendida.

Los primeros 11 bits son los bits más significativos (MSB, Most Significant Bits) (identificador base) de los 29 bits del identificador. Esos 11 bits son seguidos por el bit sustituto de petición remota (SRR-Substitutue Remote Request), el cual es recesivo y aparece en la posición del bit RTR de la trama estándar. Después del bit SRR está el bit IDE, éste es recesivo para indicar que se trata de una trama extendida.

Es importante hacer notar que si el arbitraje permanece sin resolverse después de la transmisión de los primeros 11 bits del identificador, y uno de los nodos involucrados en el arbitraje está enviando una trama **CAN** estándar (11 bits de identificación), la trama **CAN** estándar ganará el arbitraje debido a la aserción del bit dominante IDE.

El bit SRR en una trama **CAN** extendida debe ser recesivo para permitir la aserción del bit dominante RTR por el nodo que está enviando una trama remota estándar. A los bits SRR e IDE les siguen los restantes 18 bits del identificador (identificador extendido) y el bit de petición de transmisión remota.

Para habilitar el envío de tramas estándar y extendidas a través de una red compartida, los 29 bits de identificador de mensaje extendido se dividen en secciones de 11 bits (Más Significativos) y 18 bits (Menos Significativos). Esta división asegura que el bit IDE pueda permanecer en la misma posición de bit tanto en la trama estándar como en la trama extendida. Seguido del campo de arbitraje está el campo de control de 6 bits.

Los primeros 2 bits de este campo están reservados y deben ser dominantes. Los 4 bits restantes del campo de control son el DLC, el cual especifica el número de bytes de datos contenidos en el mensaje. La porción restante de la trama (campo de datos, campo del CRC, campo de reconocimiento, fin de trama y el espacio entre tramas) está construida de la misma manera que una trama de datos estándar.

2.4.5.11. Trama Remota

El formato de la trama remota es análogo a la trama de datos (estándar o extendida) pero con el bit RTR recesivo.

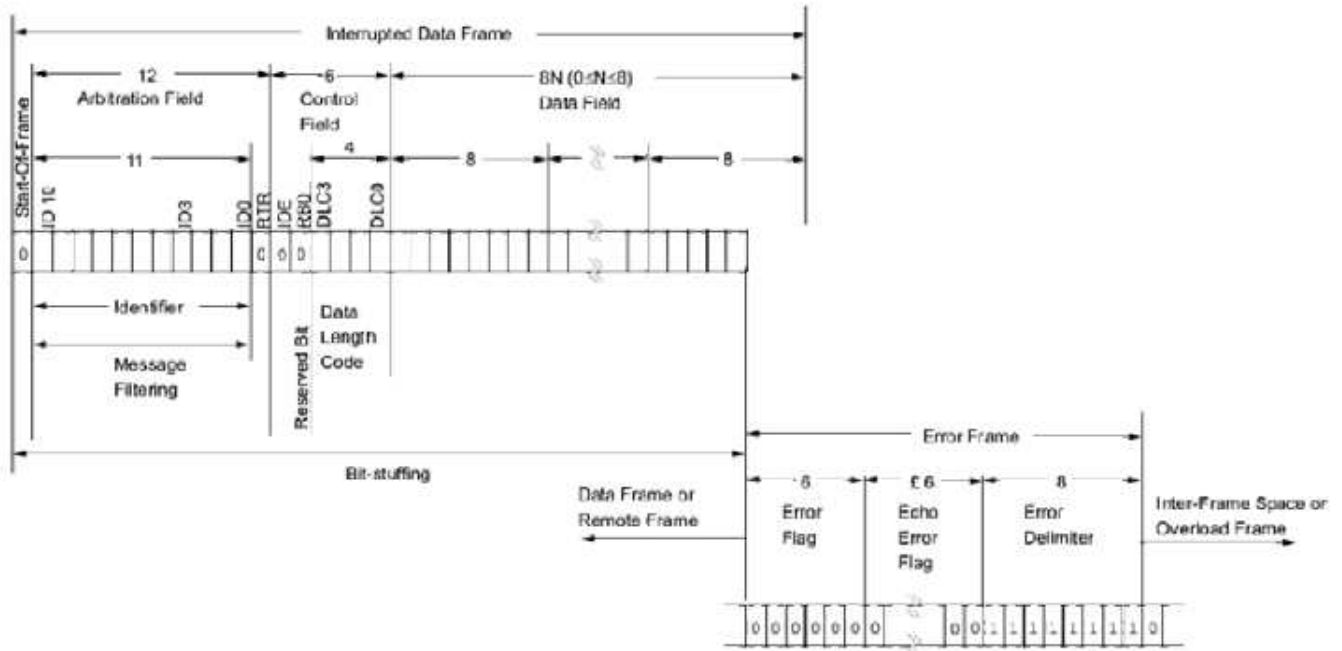


Figura 2.15: Trama de error.

Si un nodo en estado de error **Pasivo** detecta un error, el nodo transmite una **Bandera de error pasivo**, seguido, de nuevo, por el campo delimitador de error.

La Bandera de error de tipo pasivo consiste en 6 bits recessivos consecutivos y el delimitador de tipo pasivo es de 8 bit, por tanto, la trama de error para un nodo pasivo es una secuencia de 14 bits recessivos. De aquí se deduce que la transmisión de una trama de error de tipo pasivo no afectará a ningún nodo en la red, excepto cuando el error es detectado por el propio nodo que está transmitiendo.

En ese caso los demás nodos detectarán una violación de las reglas de bits de relleno y transmitirán a su vez tramas de error. Tras señalar un error por medio de la trama de error apropiada, cada nodo transmite bits recessivos hasta que recibe un bit también recessivo, luego transmite 7 bits recessivos consecutivos antes de finalizar el tratamiento de error.

2.4.5.13. Espacio entre tramas

El espacio entre tramas separa una trama (de cualquier tipo) de la siguiente trama de datos o interrogación remota. El espacio entre tramas ha de constar de, al menos, 3 bits recessivos. Esta secuencia de bits se denomina **intermisión**. Una vez transcurrida esta secuencia, un nodo en estado de error activo puede iniciar una nueva transmisión o el bus permanecerá en reposo. Para un nodo en estado de error pasivo la situación es diferente, éste deberá esperar una secuencia adicional de 8 bits recessivos antes de poder iniciar una transmisión. De esta forma se asegura una ventaja en el inicio de transmisión de información a los nodos en estado activo frente a los nodos en estado pasivo.

2.4.5.14. Bus inactivo

Este estado del bus puede ser de una longitud arbitraria. Los nodos reconocen al bus como libre y cualquiera de ellos que tenga algo que transmitir puede acceder al bus. El mensaje que esté pendiente de transmitir durante la transmisión de otro mensaje, comienza en el primer bit al terminar la intermisión.

2.4.5.15. Trama de Sobrecarga

La trama de sobrecarga se muestra en la parte inferior derecha de la figura 2.16, después del final de una trama de datos.

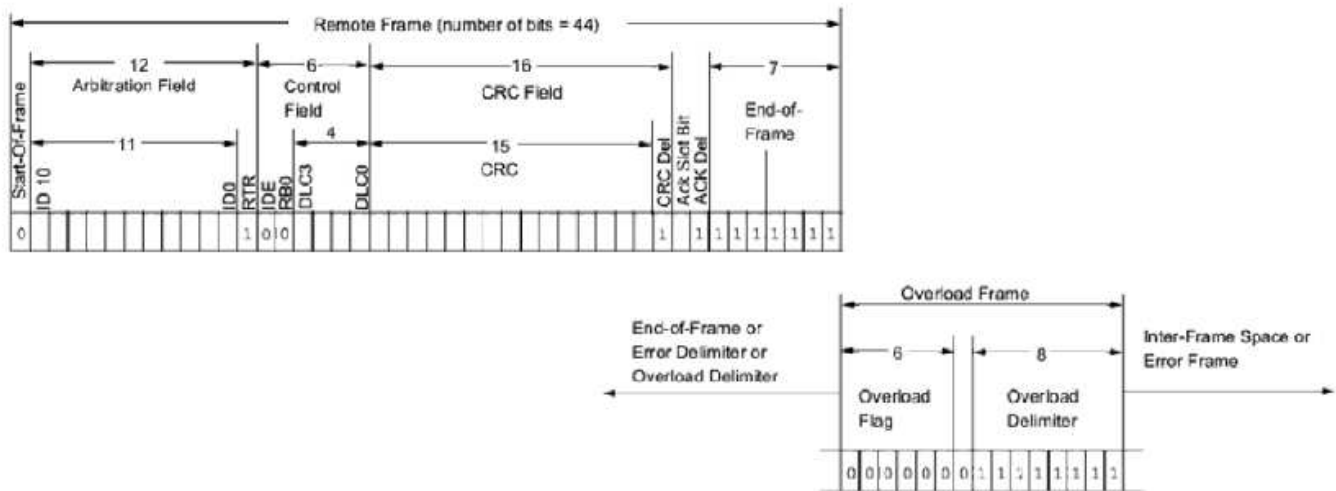


Figura 2.16: Trama de sobrecarga.

Una trama de sobrecarga tiene el mismo formato que una trama de error activo. Sin embargo, la trama de sobrecarga sólo puede generarse durante el espacio entre tramas. De esta forma se diferencia de una trama de error, que sólo puede ser transmitida durante la transmisión de un mensaje. La trama de sobrecarga consta de dos campos, la bandera de Sobrecarga, y el delimitador. La bandera de sobrecarga consta de 6 bits dominantes que pueden ser seguidos por los generados por otros nodos, dando lugar a un máximo de 12 bits dominantes. El delimitador es de 8 bits recesivos.

Una trama de sobrecarga puede ser generada por cualquier nodo como resultado de dos condiciones:

Condición 1. Debido a sus condiciones internas el nodo no puede iniciar la recepción de un nuevo mensaje. De esta forma retrasa el inicio de transmisión de un nuevo mensaje. Un nodo puede generar como máximo 2 tramas de sobrecarga consecutivas para retrasar un mensaje.

Condición 2. Por la detección por cualquier nodo de un bit dominante en los 3 bits de intermisión.

Por todo ello una trama de sobrecarga generada por un nodo dará normalmente lugar a la generación de tramas de sobrecarga por los demás nodos, dando lugar, como se ha indicado, a un máximo de 12 bits dominantes para la bandera de sobrecarga.

2.4.5.16. Manejo de error

El **protocolo CAN** está diseñado para la transferencia segura de mensajes y para proveer mecanismos para la detección de errores, señalización y autodiagnóstico, incluyendo medidas para el aislamiento de fallos, el cual previene que nodos defectuosos afecten el estado de la red. Las medidas generales para la detección de errores se basan en la capacidad de cada nodo de controlar la difusión de las transmisiones sobre el bus, si se trata de un transmisor o de un receptor, y para señalar las condiciones de error resultantes de varias fuentes. Los mensajes dañados se marcan por cualquier nodo que detecta un error, tales mensajes se abortan y serán retransmitidos automáticamente. Pasos en el tratamiento de errores en CAN

1. Se detecta un error global o local
2. Será transmitida una **Bandera de Error** (globalización del error)
3. En el caso de un error local la **Bandera de Error** procederá a sobreponerse a la **Bandera de Error** anterior seguido por el **Delimitador de Error**
4. El mensaje será descartado por cada nodo
5. Los contadores de error de cada nodo del bus son incrementados
6. La transmisión del mensaje interrumpido será repetida automáticamente

Detección de error Hay 5 tipos diferentes de error (los cuales no son mutuamente excluyentes):

- **ERROR DE BIT.** El nodo que esté enviando un bit sobre el bus también lo monitorea. Un **ERROR DE BIT** debe ser detectado en el tiempo que dura ese bit, cuando el valor del bit que se monitorea es diferente al valor del bit enviado. Como excepciones están los bits recesivos enviados como parte del proceso de arbitraje o el espacio de ACK.
- **ERROR DE RELLENO.** Un error de relleno se detecta en el sexto bit consecutivo del mismo valor en el campo de mensaje, el cual debe estar codificado por el método de relleno (bit stuffing).
- **ERROR DE CRC.** La secuencia del CRC consiste del resultado del cálculo del CRC por el transmisor. El receptor calcula el CRC de la misma forma que el transmisor. Un **ERROR DE CRC** se detecta si el resultado del cálculo no es el mismo que el recibido en la secuencia CRC.
- **ERROR DE FORMA.** Este error se detecta cuando la forma fija de un campo de bits contiene uno o más bits ilegales.
- **ERROR DE RECONOCIMIENTO.** Un **ERROR DE RECONOCIMIENTO** se detecta por el transmisor cuando no detecta un bit dominante durante el espacio ACK.

2.4.5.17. Señalización de error

Un nodo que detecta una condición de error lo señala transmitiendo una bandera de error. La trama de error no es una trama verdadera, en realidad es el resultado de una secuencia de señalización de error que consiste en la superposición de banderas de error, transmitido desde diferentes nodos, posiblemente en diferentes momentos, seguido por un campo delimitador de error. Siempre que un nodo detecta un ERROR DE BIT, un ERROR DE RELLENO, un ERROR DE FORMA o un ERROR DE RECONOCIMIENTO, se empieza la transmisión de una bandera de error en el respectivo nodo en el siguiente bit.

Siempre que se detecta un ERROR DE CRC, la transmisión de una bandera de error empieza en el siguiente bit al delimitador ACK, al menos que la bandera de error ya haya comenzado debido a otra condición. Para un nodo en error activo es una bandera de ERROR ACTIVO, el cual consta de 6 bits **dominantes**, para un nodo en error pasivo es una bandera de ERROR PASIVO, que consiste de 6 bits recesivos consecutivos.

Las banderas de **ERROR** violan la regla de relleno de bits o destruye la forma fija del campo de reconocimiento o el de fin de trama. Como consecuencia, todos los demás nodos detectan una condición de error y comienzan la transmisión de una bandera de ERROR por su parte. Así que, la secuencia de bits dominantes que pueden ser monitoreados por los niveles del bus, es una superposición de diferentes banderas de error transmitidos por los nodos individuales.

La longitud total de esta secuencia varía entre un mínimo de seis y un máximo de doce bits. La bandera de error pasivo enviado por un nodo en error pasivo no tiene efecto sobre el bus. Sin embargo, el nodo de señalización todavía tendrá que esperar seis bits consecutivos de igual nivel, que comienza en el principio de la bandera de error pasivo antes de continuar. El tiempo de recuperación desde la detección de un error hasta el comienzo del siguiente mensaje, la mayoría de las veces, es de 31 bits.

2.4.5.18. Aislamiento de fallos

El **protocolo CAN** tiene un protocolo de aislamiento de fallos que detecta unidades defectuosas y las pone en estados pasivos o desactivadas, de manera que no puedan afectar el estado del bus con sus salidas. El protocolo asigna a cada nodo uno de tres estados:

- Error activo
- Error pasivo
- Fuera de bus

Las unidades cambian sus estados de acuerdo al valor de dos contadores enteros:

Una unidad en ERROR ACTIVO normalmente puede tomar parte de las comunicaciones del bus y envía una bandera de ERROR ACTIVO cuando se ha detectado un error. Una unidad en ERROR PASIVO no debe enviar una bandera de ERROR ACTIVO. Éste toma parte de las comunicaciones en el bus, pero cuando un error ha sido detectado, sólo se envía una bandera de ERROR PASIVO.

También después de una transmisión, una unidad en ERROR PASIVO esperará antes de iniciar una nueva transmisión.

A una unida en **fuera de bus** no se le permite tener alguna influencia sobre el bus. Para el aislamiento de fallos están implementados dos contadores en cada unidad en el bus:

- Contador de error de transmisión
- Contador de error de recepción

Esos contadores se modifican de acuerdo a las siguientes reglas:

- Cuando un **RECEPTOR** detecta un error, el CONTADOR DE ERROR DE RECEPCIÓN se incrementará en 1, excepto cuando el error detectado fue un ERROR DE BIT, durante el envío de una BANDERA DE ERROR ACTIVO o de una BANDERA DE SOBRECARGA.
- Cuando un **RECEPTOR** detecta un bit **dominante**, como el primer bit después de enviar una BANDERA DE ERROR, el CONTADOR DE ERROR DE RECEPCIÓN se incrementará en 8.
- Cuando un **TRANSMISOR** envía una BANDERA DE ERROR, el CONTADOR DE ERROR DE TRANSMISIÓN se incrementa en 8.

Excepción 1. Si el transmisor está en ERROR PASIVO y detecta un ERROR DE RECONOCIMIENTO, porque no detecta un ACK **dominante** y no detecta un bit **dominante** al enviar su BANDERA DE ERROR PASIVO.

Excepción 2. Si el **TRANSMISOR** envía una BANDERA DE ERROR porque ocurrió un ERROR DE RELLENO durante el **ARBITRAJE**, y debe haber sido **recesivo**, y ha sido enviado como **recesivo** pero monitoreado como **dominante**.

En las excepciones 1 y 2 el CONTADOR DE TRANSMISIÓN no cambia.

- Si un **TRANSMISOR** detecta un ERROR DE BIT mientras envía una BANDERA DE ERROR ACTIVO o una BANDERA DE SOBRECARGA, el CONTADOR DE ERROR DE TRANSMISIÓN se incrementa en 8.
- Si un **RECEPTOR** detecta un ERROR DE BIT mientras envía una BANDERA DE ERROR ACTIVO o una BANDERA DE SOBRECARGA, el CONTADOR DE ERROR DE RECEPCIÓN se incrementa en 8.
- Cualquier nodo tolera hasta 7 bits **dominantes** consecutivos después de enviar una BANDERA DE ERROR ACTIVO, BANDERA DE ERROR PASIVO o BANDERA DE SOBRECARGA. Después de detectar el décimo cuarto bit **dominante** consecutivo (en el caso de una BANDERA DE ERROR ACTIVO o de una BANDERA DE SOBRECARGA) o después de detectar el octavo bit **dominante** consecutivo seguido de una BANDERA DE ERROR PASIVO, y después de cada secuencia de ocho bits **dominantes** consecutivos adicionales cada **TRANSMISOR** incrementa su CONTADOR DE ERROR DE TRANSMISIÓN en 8 y cada **RECEPTOR** incrementa su CONTADOR DE ERROR DE RECEPCIÓN en 8.

- Después de una transmisión exitosa de un mensaje (obtener un ACK y sin error hasta finalizar **FIN DE TRAMA**). El CONTADOR DE ERROR DE TRANSMISIÓN se disminuye en 1, al menos que ya esté en 0.
- Después de una recepción exitosa de un mensaje (recepción sin error hasta el **ESPACIO ACK** y el envío exitoso del bit ACK), el CONTADOR DE ERROR DE RECEPCIÓN se disminuye en 1, si estaba entre 1 y 127. Si el CONTADOR DE ERROR DE RECEPCIÓN estaba en 0, se queda en 0, y si era mayor a 127, entonces se establecerá en un valor de entre 119 y 127.
- Un nodo está en ERROR PASIVO cuando el CONTADOR DE ERROR DE TRANSMISIÓN es igual o mayor a 128, o cuando el CONTADOR DE ERROR DE RECEPCIÓN es igual o mayor a 128. Una condición de error que deja que un nodo se convierta a ERROR PASIVO provoca que el nodo envíe una BANDERA DE ERROR ACTIVO.
- Un nodo esta en FUERA DE BUS cuando el CONTADOR DE ERROR DE TRANSMISIÓN es mayor o igual a 256.
- Un nodo en ERROR PASIVO se vuelve a ERROR ACTIVO otra vez cuando, tanto el CONTADOR DE ERROR DE TRANSMISIÓN como el CONTADOR DE ERROR DE RECEPCIÓN son menores o igual a 127.
- A un nodo que esté FUERA DE BUS se le permite convertirse a ERROR ACTIVO (ya no más FUERA DE BUS) con sus dos contadores de error puestos en 0, y después de 128 apariciones de 11 bits **recesivos** consecutivos que han sido monitoreados en el bus.

Nota: Una cuenta de error mayor de alrededor a 96 indica una fuerte perturbación en el bus. Puede tener ventaja proveer medios para poner a prueba esta condición.

Nota: Inicio: Si durante la puesta en marcha del sistema sólo está en línea un nodo, y éste nodo transmite algún mensaje, no recibirá ningún reconocimiento, detecta un error y repite el mensaje. Este nodo se pone en ERROR PASIVO pero no en FUERA DE BUS debido a la razón mencionada.

2.4.5.19. Tolerancia del oscilador en el bus CAN

La tolerancia máxima permitida del oscilador en el **bus CAN** es de 1.58%. La oscilación está dada por un resonador cerámico, por lo tanto se requiere de su uso en un bus con una velocidad de hasta 125 kbits/s como regla de oro. Para una mayor velocidad y mejor precisión del bus para el **protocolo CAN** se necesita de un oscilador de cuarzo.

2.4.5.20. Requerimientos para el tiempo de bit

Todos los dispositivos en el **bus CAN** deben tener la misma velocidad de bits. Sin embargo, no se requiere que todos los nodos tengan la misma frecuencia del oscilador del reloj maestro. Por las diferencias en las frecuencias de los relojes de cada nodo, la velocidad de bits tiene que ser ajustada estableciendo apropiadamente el pre-escalador del **Baud Rate** y el número de cuantos de tiempo en cada segmento.

El tiempo de los bits en **CAN** está compuesto por segmentos no superpuestos. Cada uno de esos segmentos está compuesto por unidades llamadas **Cuantos de Tiempo** (TQ-Time Quanta). La velocidad nominal de bits (NBR-Nominal Bit Rate) está definida en **CAN** como el número de bits transmitidos por segundo por un transmisor ideal sin re-sincronización. Se puede describir con la ecuación 2.1:

$$NBR = f_{\text{bit}} = \frac{1}{t_{\text{bit}}} \quad (2.2)$$

2.4.5.21. Tiempo nominal de bit.

El tiempo nominal de bit (NBT) (t_{bit}) está compuesto por segmentos no superpuestos, mostrados en la figura 2.17.

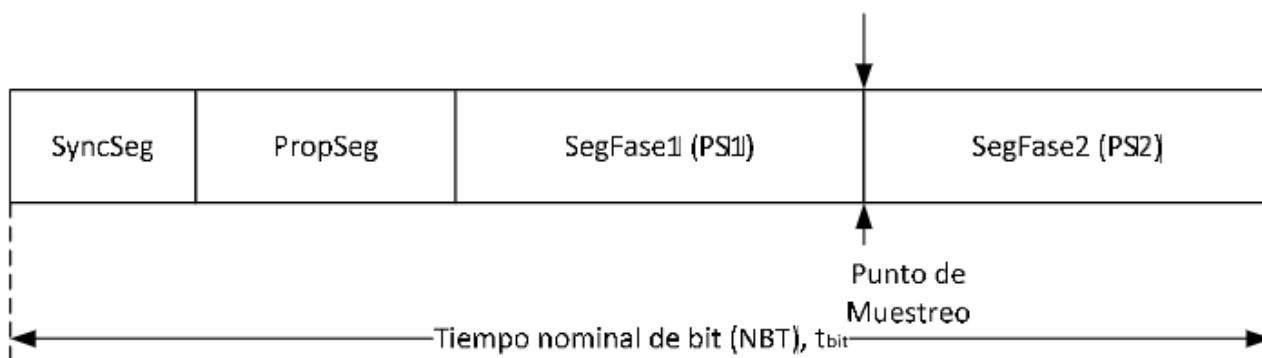


Figura 2.17: Segmentos del tiempo de bit en CAN.

El NBT es la suma de los siguientes segmentos:

$$t_{\text{bit}} = t_{\text{SecSinc}} + t_{\text{SecProp}} + t_{PS1} + t_{PS2} \quad (2.3)$$

Asociado con el NBT está el punto de muestreo, el **Ancho de Salto de sincronización (SJW-Synchronization Jump Width)** y el **Tiempo de Procesamiento de la Información (IPT-Information Processing Time)**. Se requiere que los nodos se sincronicen en los flancos de los bits, así que cada nodo concuerda con el valor del bit transmitido actualmente en el bus. Para sincronizarse, cada nodo implementa un protocolo de sincronización que mantiene la velocidad de bits del receptor alineado con la velocidad actual de los bits transmitidos.

El protocolo de sincronización usa los flancos de transición para re-sincronizar a los nodos. Por lo tanto, se deben evitar las largas secuencias sin transición de bits para evitar desfasamientos entre los bits de los nodos. Esta es la razón por la que el protocolo emplea la técnica llamada **bit de relleno**, lo que fuerza a complementar el flujo después de transmitir 5 bits del mismo tipo. Los bits de relleno son insertados automáticamente por el nodo transmisor y removidos en el lado receptor antes del procesamiento del contenido de la trama.

La transmisión síncrona de los bits permite el protocolo de arbitraje en **CAN** y simplifica el manejo del flujo de bits, pero también requiere un sofisticado protocolo de sincronización. La sincronización de los bits se realiza primero al recibir el bit de inicio disponible con cada transmisión asíncrona. Después, para permitir que los receptores lean correctamente el contenido del mensaje se requiere de una continua re-sincronización.

Otras características del protocolo influyen en la definición de la temporización del bit. Para propósito de arbitraje del bus, reconocimiento del mensaje y señalización de error, el protocolo requiere que los nodos puedan cambiar el estado de un bit transmitido de recesivo a dominante, informando del cambio de estado del bit a todos los demás nodos de la red antes de finalizar la transmisión. Esto significa que el tiempo del bit debe ser al menos lo suficiente grande para acomodar la propagación de la señal desde cualquier transmisor a cualquier receptor y de regreso al transmisor.

El tiempo del bit incluye un segmento de retardo de la propagación que toma en cuenta la propagación de la señal en el bus, así como los retardos causados por la transmisión y recepción en los nodos. En la práctica esto significa que la señal de propagación se determina por los dos nodos dentro del sistema que están más apartados entre sí. El flanco del bit de un nodo transmisor alcanza a un nodo receptor, después la señal se propaga por todo el camino desde los dos nodos.

En este punto el receptor puede cambiar su valor de recesivo a dominante, pero el nuevo valor no alcanzará el transmisor hasta que la señal se propague por todo el camino de regreso; sólo entonces el primer nodo puede decidir si el nivel de su propia señal (en este caso recesivo) es el valor actual en el bus o si éste ha sido remplazado por un nivel dominante por otro nodo. Considerando el protocolo de sincronización y a la necesidad de que todos los nodos estén de acuerdo con el valor del bit, el tiempo nominal del bit puede definirse como la composición de cuatro segmentos. Estos segmentos son:

- **Segmento de sincronización (SyncSeg).** Es el primer segmento en el NBT y se usa para sincronizar a los nodos en el bus. Se espera que los flancos de los bits ocurran dentro de este segmento.
- **Segmento de propagación (PropSeg).** Este segmento sirve para compensar los retardos físicos dentro de la red. El retardo de propagación se define como dos veces la suma del tiempo de propagación de la señal en el cable del bus, el retardo del comparador de entrada y el retardo del controlador de salida.
- **Segmento de fase 1 y 2 (PS1 y PS2 respectivamente).** Los dos segmentos de fase se usan para compensar el error de fase de los bordes de los bits dentro del bus. PS1 se puede alargar o PS2 acortar en la re-sincronización.
- **Punto de muestreo (SP).** El punto de muestreo es el punto en el cual se lee e interpreta el nivel del bus como el valor respectivo del bit. El punto de muestro se localiza al final de PS1.
- **Tiempo de Cuanta (TQ).** El tiempo de cuanta es una unidad de tiempo fija derivada del período del oscilador. Hay un pre-escalador programable, con valores enteros, que van desde 1 a 32. Empezando con el **TIEMPO MÍNIMO DE CUANTA**.

El **TIEMPO DE CUANTA** puede tener una longitud de:

$$\text{TIEMPO MÍNIMO DE CUANTA} = m \times \text{TIEMPO MÍNIMO DE CUANTA} \quad (2.4)$$

donde m es el valor del pre-escalador.

El **TIEMPO DE CUANTA** es la mínima resolución en la definición del tiempo del bit y el error máximo asumido por el protocolo de sincronización orientado a bits. La longitud de cada segmento queda:

- SyncSeg es de 1 **TQ** de longitud.
- PropSeg se programa para ser de longitud $1, 2, \dots, 8\text{TQ}$.
- PS1 se programa para ser de longitud $1, 2, \dots, 8\text{TQ}$.
- PS2 con un máximo de entre PS1 y del **TIEMPO DE PROCESAMIENTO DE LA INFORMACIÓN**, éste último debe ser menor o igual a 2 **TQ**.

El número total de **TQ** en el tiempo de un bit puede programarse desde al menos 8 hasta 25.

2.4.5.22. Sincronización

Para compensar los desplazamientos de fase entre las frecuencias de los osciladores de cada nodo del bus, cada controlador **CAN** debe ser capaz de sincronizarse con el flanco relevante de la señal entrante. Cuando se detecta un flanco en el dato transmitido, la lógica comparará la ubicación del flanco en el momento previsto (SyncSeg). Entonces el circuito ajustará los valores de PS1 y PS2 como sea necesario.

Hay dos mecanismos de sincronización:

- **Hard Sincronización.** Toma lugar al comienzo de la trama, cuando el bit de inicio de trama SOF cambia el estado del bus de recesivo a dominante. Después de la detección del flanco correspondiente, el tiempo de bit se reinicia al final del segmento de sincronización. Por lo tanto **hard synchronization** fuerza que el flanco del bit de inicio se encuentre dentro del segmento de sincronización del tiempo de bit reiniciado.
- **Re-sincronización.** Como resultado de la re-sincronización el PS1 puede alargarse o el PS2 acortarse, ver figura 2.18.

La cantidad de alargamiento o acortamiento de **PS1** y **PS2** tiene un límite superior dado por **SJW**, el cual debe ser programado entre 1TQ y 4TQ . La información de sincronización sólo se puede derivar de la transición de un valor de bit a otro. Por lo tanto, la posibilidad de volver a sincronizar un nodo del bus con el flujo de bits durante una trama depende de la propiedad de que exista un intervalo máximo de tiempo entre dos transiciones de bit (aquí la importancia del protocolo de bits de relleno). El diseñador del dispositivo puede programar los parámetros del tiempo de bit en el controlador **CAN** por medio de registros apropiados. Tomar en cuenta que, dependiendo del tamaño del segmento de propagación, se puede determinar la longitud máxima posible del bus a una velocidad de datos específica (o la velocidad máxima de datos con una longitud de bus específica).

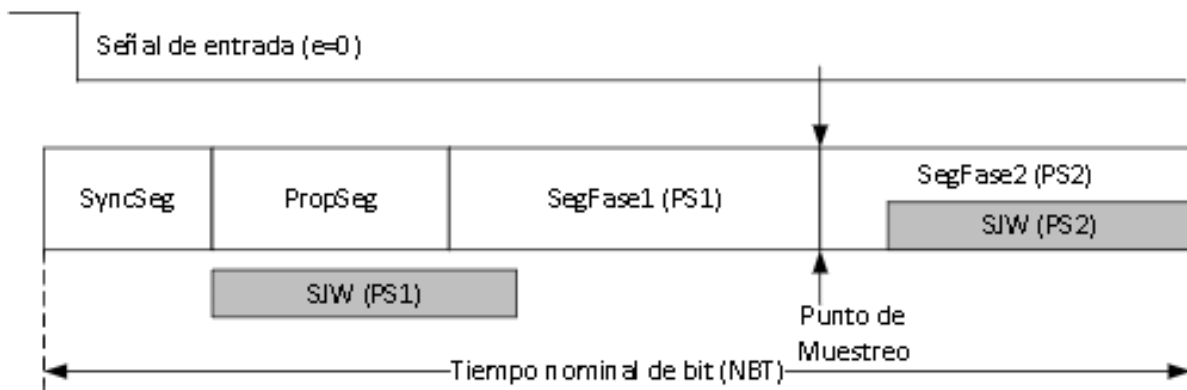
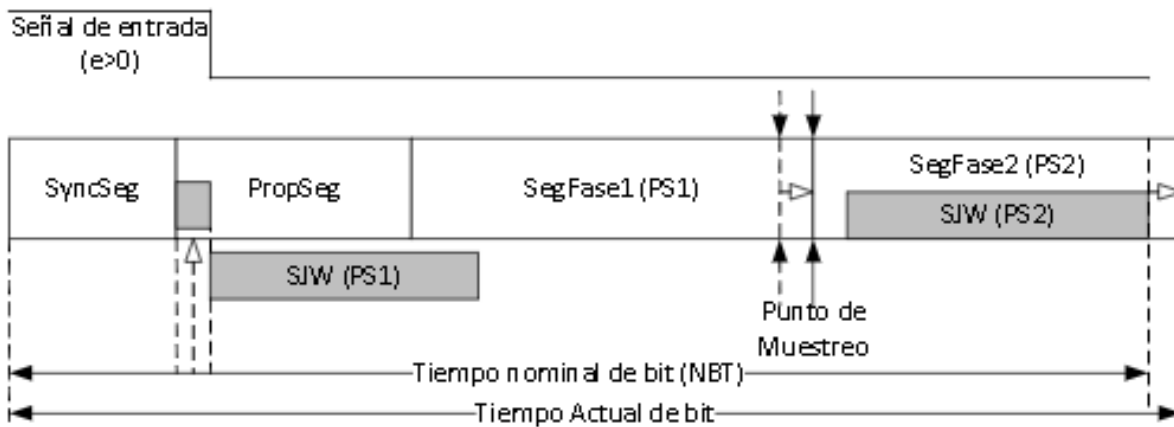
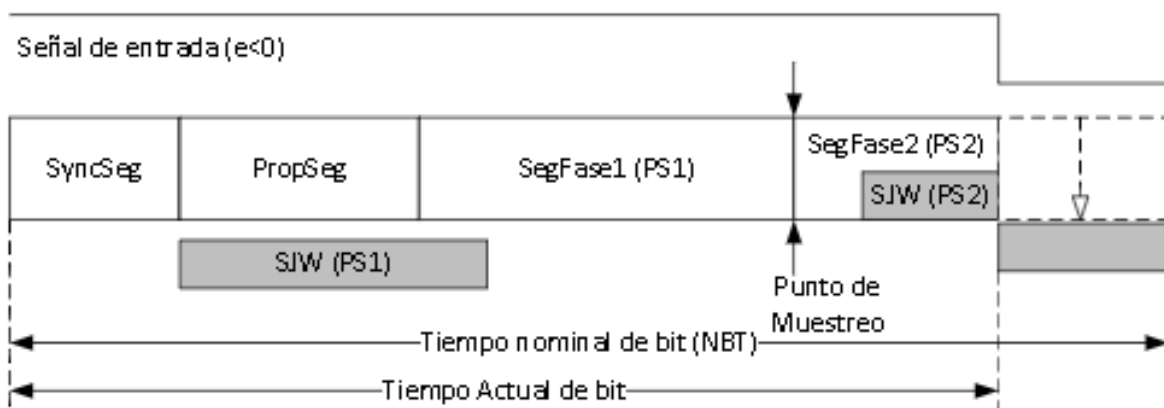
(a) Sin resincronización ($e = 0$)(b) Resincronización a un transmisor mas lento ($e > 0$)(c) Resincronización a un transmisor mas rápido ($e < 0$)

Figura 2.18: Re-sincronización el tiempo de bit.

2.4.5.23. Errores de fase

El ERROR DE FASE de un flanco está dado por la posición del flanco con relación al SyncSeg, se mide en **TQ**. El signo del ERROR DE FASE se define como sigue:

- **Sin error de fase $e = 0$** : Si la magnitud del error de fase es menor o igual que el valor programado del **SJW**, el efecto de la re-sincronización es el mismo que en **hard synchronization**. $e = 0$ si el flanco se encuentra dentro del SyncSeg.
- **Error de fase positivo ($e > 0$)**: Si la magnitud del error de fase es mayor que el **SJW** y, si el error de fase es positivo, **PS1** es alargado una cantidad igual al **SJW**. $e > 0$ si el flanco se encuentra antes del punto de muestreo (se añaden **TQ** al **PS1**).
- **Error de fase negativo ($e < 0$)**: Si la magnitud del error de fase es mayor que el **SJW**, y el error de fase es negativo, **PS2** es acortado por una cantidad igual al **SJW**. $e < 0$ si el flanco se encuentra después del punto de muestreo (se sustraen del **PS2**).

2.4.5.24. Reglas de sincronización

1. Sólo se usarán los flancos de recesivo a dominante para sincronización.
2. Se permite sólo una sincronización dentro de un tiempo de bit.
3. Un flanco podrá ser usado para sincronización sólo si el valor detectado en el punto de muestreo previo difiere del valor del bus inmediatamente después del flanco.
4. Un nodo transmisor no se sincronizará en un error de fase positivo ($e > 0$).
5. Si la magnitud absoluta del error de fase es mayor que el **SJW**, el segmento de fase apropiado se ajustará por una cantidad igual al **SJW**.

2.5. Protocolo de comunicaciones CAN-Bus

Como ya se ha mencionado, CAN es un protocolo de comunicaciones serie que soporta control distribuido en tiempo real con un alto nivel de seguridad y multiplexación. El establecimiento de una red CAN para interconectar los dispositivos electrónicos internos de un vehículo tiene la finalidad de sustituir o eliminar el cableado (como lo comentamos anteriormente). Las ECU, sensores, sistemas antideslizantes, etcétera, se conectan mediante una red CAN a velocidades de transferencia de datos de hasta 1 Mbps.

De acuerdo al modelo de referencia OSI (Open Systems Interconnection, Modelo de interconexión de sistemas abiertos), la arquitectura de protocolos CAN incluye tres capas: física, de enlace de datos y aplicación, además de una capa especial para gestión y control del nodo llamada capa de supervisor.

- **Capa física**: define los aspectos del medio físico para la transmisión de datos entre nodos de una red CAN, los más importantes son niveles de señal, representación, sincronización y tiempos en los que los bits se transfieren al bus. La especificación del protocolo CAN no define una capa física, sin embargo, los estándares ISO 11898 establecen las características que deben cumplir las aplicaciones para la transferencia en alta y baja velocidad.

- **Capa de enlace de datos:** define las tareas independientes del método de acceso al medio, además debido a que una red CAN brinda soporte para procesamiento en tiempo real a todos los sistemas que la integran, el intercambio de mensajes que demanda dicho procesamiento requiere de un sistema de transmisión a frecuencias altas y retrasos mínimos. En redes multimaestro, la técnica de acceso al medio es muy importante ya que todo nodo activo tiene los derechos para controlar la red y acaparar los recursos. Por lo tanto, la capa de enlace de datos define el método de acceso al medio, así como los tipos de tramas para el envío de mensajes.

Cuando un nodo necesita enviar información a través de una red CAN, puede ocurrir que varios nodos intenten transmitir simultáneamente. CAN resuelve lo anterior al asignar prioridades mediante el identificador de cada mensaje, donde dicha asignación se realiza durante el diseño del sistema en forma de números binarios y no puede modificarse dinámicamente. El identificador con el menor número binario es el que tiene mayor prioridad.

El método de acceso al medio es el de **Acceso Múltiple por Detección de Portadora, con Detección de Colisiones y Arbitraje por Prioridad de Mensaje**. De acuerdo con este método, los nodos en la red que necesitan transmitir información deben esperar a que el bus esté libre (detección de portadora); cuando se cumple esta condición, dichos nodos transmiten un bit de inicio (acceso múltiple). Cada nodo lee el bus bit a bit durante la transmisión de la trama y comparan el valor transmitido con el valor recibido; mientras los valores sean idénticos, el nodo continúa con la transmisión; si se detecta una diferencia en los valores de los bits, se lleva a cabo el mecanismo de arbitraje. CAN establece dos formatos de tramas de datos (data frame) que difieren en la longitud del campo del identificador, las tramas estándares (standard frame) con un identificador de 11 bits definidas en la especificación CAN 2.0A, y las tramas extendidas (extended frame) con un identificador de 29 bits definidas en la especificación CAN 2.0B.

Para la transmisión y control de mensajes CAN, se definen cuatro tipos de tramas: de datos, remota (remote frame), de error (error frame) y de sobrecarga (overload frame). Las tramas remotas también se establecen en ambos formatos, estándar y extendido, y tanto las tramas de datos como las remotas se separan de tramas precedentes mediante espacios entre tramas (interframe space). En cuanto a la detección y manejo de errores, un controlador CAN cuenta con la capacidad de detectar y manejar los errores que surjan en una red. Todo error detectado por un nodo, se notifica inmediatamente al resto de los nodos.

- **Capa de supervisor:** La sustitución del cableado convencional por un sistema de bus serie presenta el problema de que un nodo defectuoso puede bloquear el funcionamiento del sistema completo. Cada nodo activo transmite una bandera de error cuando detecta algún tipo de error y puede ocasionar que un nodo defectuoso pueda acaparar el medio físico. Para eliminar este riesgo el protocolo CAN define un mecanismo autónomo para detectar y desconectar un nodo defectuoso del bus, dicho mecanismo se conoce como aislamiento de fallos.
- **Capa de aplicación:** Existen diferentes estándares que definen la capa de aplicación; algunos son muy específicos y están relacionados con sus campos de aplicación. Entre las capas de aplicación más utilizadas cabe mencionar CAL, CANopen, DeviceNet, SDS (Smart Distributed System), OSEK, CANKingdom.

2.5.1. Elementos que componen el sistema CAN-Bus

2.5.1.1. Cables

La información circula por dos cables trenzados que unen todas las unidades de control que forman el sistema. Esta información se transmite por diferencia de tensión entre los dos cables, de forma que un valor alto de tensión representa un 1 y un valor bajo de tensión representa un 0. La combinación adecuada de unos y ceros conforman el mensaje a transmitir.

En la Figura 2.19 se presenta el par trenzado.



Figura 2.19: Par trenzado de CAN Bus.

Los colores **CAN High** y **CAN Low** son de colores definidos por normas de cada marca, un ejemplo lo muestra la siguiente tabla:

Bus de Comunicación	Color CAN
CAN Low	Naranja-Café
CAN High	Naranja-Negro

Tabla 2.3: Colores de cables de CAN Bus.

El par trenzado bajo una norma puede cumplir las siguientes características:

- Longitud de trenza $20 \text{ mm} \pm 2 \text{ mm}$
- Longitud transversal 0.35 mm

En un cable los valores de tensión oscilan entre 1.4 V . (valor dominante). y 4.7 V (Valor recesivo) lo que se denomina cable L (Low) y en el otro, el cable H (High) lo hacen entre 0.3 V . (valor recesivo) y 3.6 V . (valor dominante) para buses de comunicación de baja velocidad.

Mientras que para los buses de alta velocidad los valores de voltaje en el cable H (High) oscilan entre 2.6 V. (valor recesivo) y 4.5 V. (valor dominante) Y para el cable L (Low) 2.6 V. (valor recesivo) y 2.25 V. (valor dominante). Ver figura 2.20.

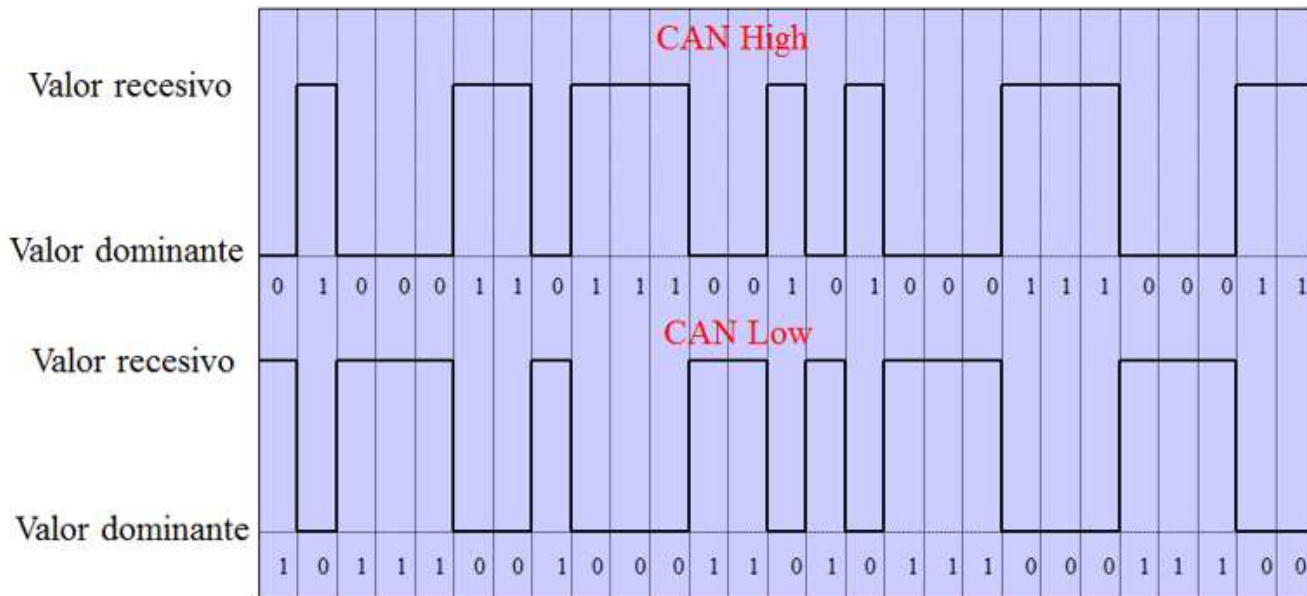


Figura 2.20: Valores recesivos y dominantes en CAN Bus.

En caso de que se interrumpa la línea H o que se derive a masa, el sistema trabajará con la señal de Low con respecto a masa, en el caso de que se interrumpa la línea L, ocurrirá lo contrario. Esta situación permite que el sistema siga trabajando con uno de los cables cortados o comunicados a masa, incluso con ambos comunicados también sería posible el funcionamiento (solo para buses de baja velocidad), quedando fuera de servicio solamente cuando ambos cables se cortan.

Para buses de alta velocidad la no existe comunicación en modo unifilar ya que por ser un bus de seguridad al quedar unidos los cables (CAN High y CAN Low) existe lo denominado como BUS OFF, el auto seguirá funcionando, pero desplegará testigos de falla en el instrumento combinado.

Es importante tener en cuenta que el trenzado entre ambas líneas sirve para anular los campos magnéticos, por lo que no se debe modificar en ningún caso ni el paso ni la longitud de dichos cables ya que la resistencia de cierre está calculada para las distancias del arnés del automóvil.

2.5.1.2. Elemento de cierre o terminador

Son resistencias conectadas a los extremos de los cables High y Low. Sus valores se obtienen de forma empírica y permiten adecuar el funcionamiento del sistema a diferentes longitudes de cables y número de unidades de control abonadas, ya que impiden fenómenos de reflexión que pueden perturbar el mensaje.

Estas resistencias están alojadas en el interior de algunas de las unidades de control del sistema por cuestiones de economía y seguridad de funcionamiento

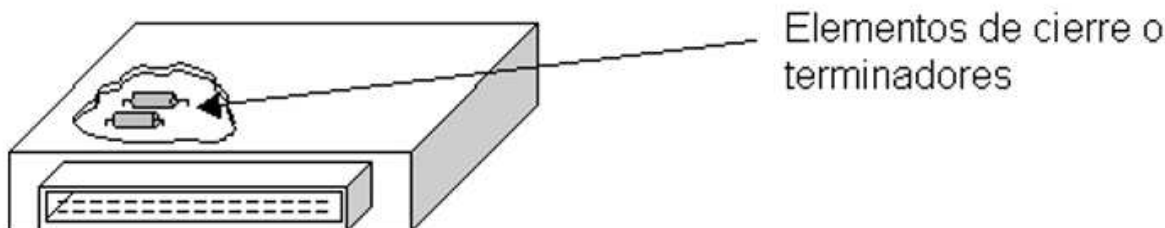


Figura 2.21: Resistencia de cierre del CAN Bus.

2.5.1.3. Controlador

Es el elemento encargado de la comunicación entre el microprocesador de la unidad de control y el transmisor-receptor. Trabaja acondicionando la información que entra y sale entre ambos componentes. El controlador está situado en la unidad de control, por lo que existen tantos como unidades estén conectadas al sistema. Este elemento trabaja con niveles de tensión muy bajos y es el que determina la velocidad de transmisión de los mensajes, que será más o menos elevada según el compromiso del sistema. Así, en la línea de CAN-Bus del motor-frenos-cambio automático es de 500 K baudios, y en los sistemas de confort hasta de 100 K baudios. Este elemento también interviene en la necesaria sincronización entre las diferentes unidades de mando para la correcta emisión y recepción de los mensajes.

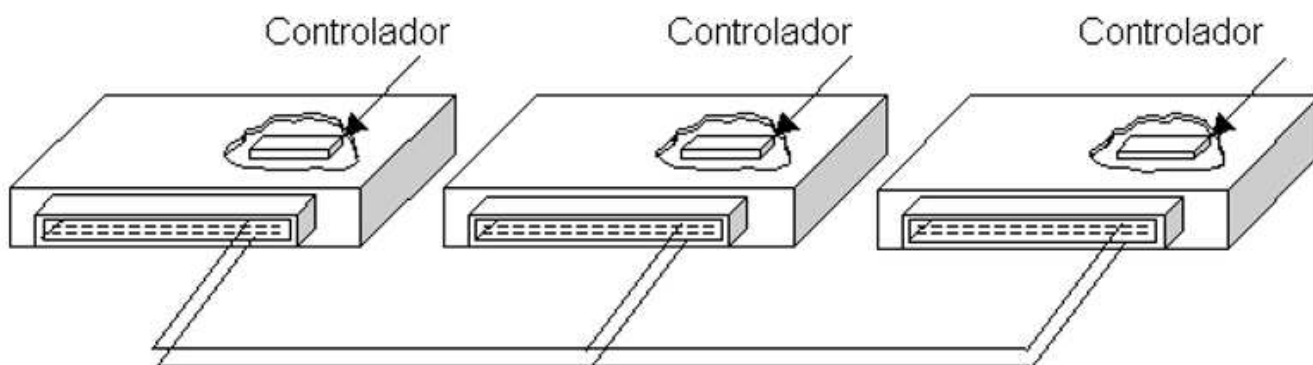


Figura 2.22: Controlador de las unidades de control.

2.5.1.4. Transmisor-Receptor

El transmisor-receptor es el elemento que tiene la misión de recibir y de transmitir los datos, además de acondicionar y preparar la información para que pueda ser utilizada por los controladores. Esta preparación consiste en situar los niveles de tensión de forma adecuada, amplificando la señal cuando la información se vuelca en la línea y reduciéndola cuando es recogida de la misma y suministrada al controlador.

El transmisor-receptor es básicamente un circuito integrado que está situado en cada una de las unidades de control abonadas al sistema, trabaja con corrientes aproximadas a 0.5 A y en ningún caso interviene modificando el contenido del mensaje. Funcionalmente está situado entre los cables que forman la línea CAN-Bus y el controlador.

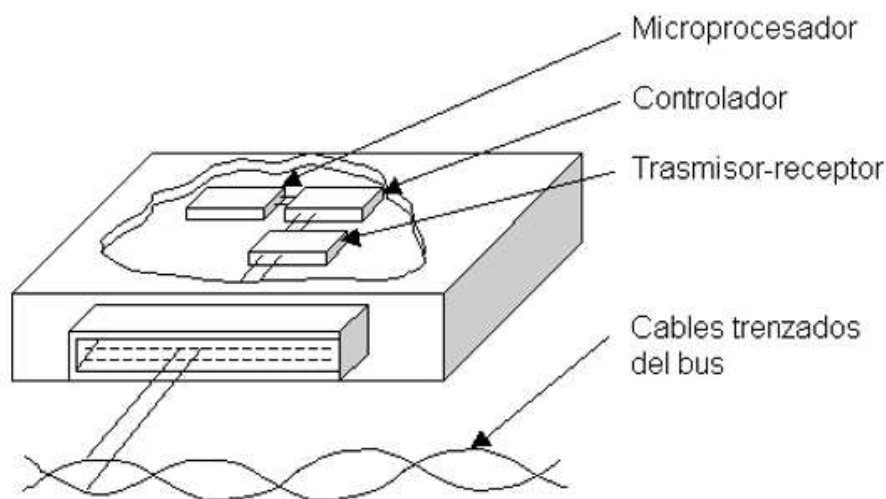


Figura 2.23: Transceiver de las unidades de control.

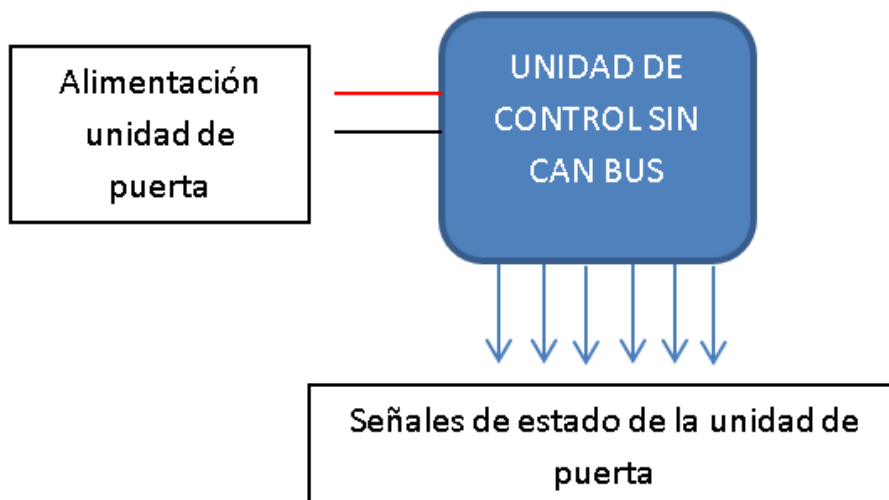
2.5.2. El CAN-Bus en los autos

El empleo de CAN-Bus, como se ha comentado anteriormente, aumenta la posibilidad de utilizar más unidades de control electrónicas en el automóvil y reducir el número de los cables y con eso también el número de las terminales en los conectores de salidas físicas de las unidades.

Los datos con la diferente información de los sensores del Auto son intercambiados entre las unidades de control para sus funciones específicas.

La programación o actualizaciones y un diagnóstico para corroborar el estado de las unidades de control se hace también por medio del CAN-Bus en los automóviles. Las funciones de confort son cada vez más exigentes en el mercado y obligan el cambio de software de las unidades de control. Resultado de esto los requisitos especiales de los largos cables queda únicamente limitado para la alimentación de las unidades de control o sensores inteligentes montados en el Auto.

En la Figura 2.24(a) se presentan la Unidad de Control sin CAN-Bus, mientras que en 2.24(b) se muestra la Unidad de Control con CAN-Bus.



(a) *Unidad de Control sin CAN-Bus*



(b) *Unidad de Control con CAN-Bus*

Figura 2.24: Unidad de control

Desde el año 1998 el consorcio Volkswagen (por ejemplo) empezó a utilizar el CAN-Bus (unidades de control de puertas) en una de sus arquitecturas electrónicas con velocidades de transmisión de 100 K baudios y dejando atrás las transmisiones de 64.5 K baudios. Al decir que se utilizó CAN BUS en las unidades de puerta del automóvil Fabia de Volkswagen fue un gran avance ya que el cableado se redujo, anteriormente se utilizaban varias señales para informar a las de más electrónicas el estado de las unidades de puertas y con esto se redujo el cableado a dos hilos de comunicación y dos de alimentación.

2.5.3. CAN Low Speed

La descentralización de las unidades de control en el automóvil, normas ISO, aumento de funciones, llevo a la tendencia de utilizar no solo un bus de comunicación sino más buses para los automóviles más equipados, es por eso que se desarrollaron Buses de baja velocidad (100 K baudios) y alta velocidad (500 K baudios), dependiendo la función de la unidad de control en cuanto a función y seguridad.

Con el paso del tiempo los transceivers de baja velocidad se dejaron de fabricar y se pasó a la utilización de transceivers de alta velocidad para los autos actuales los cuales trabajan a velocidades de 500 K baudios actualmente.

2.5.4. CAN High Speed

El Bus de comunicación de alta velocidad en el cual nos enfocaremos debido a su utilización hoy en día trabaja a una velocidad de transmisión de 500 K baudios 5 veces mayor que el de baja velocidad (obsoleto), esto debido a la necesidad más procesamiento de más mensajes lo cual amplía la cantidad de funciones eléctrico electrónicas que pueden llevar los autos, pero también hacen más complejas de las arquitecturas electrónicas.

Las arquitecturas electrónicas en los automóviles se pueden dividir hoy en 1 o más buses de comunicación, en este caso hablaremos de una arquitectura de 4 buses de comunicación High Speed los cuales describiremos a continuación:

- **CAN Confort:** Bus de comunicación de 500 K baudios y en el cual van montadas las unidades de control que brindan ayuda electromecánica dentro del Automóvil.

Las unidades de control montadas en este Bus son las siguientes:

- **ECU Puerta conductor.** Unidas de control de puerta de conductor.
- **ECU Puerta acompañante.** Unidad de control de puertas de acompañante.
- **ECU Aire acondicionado.** Unidad de control de Clima.
- **ECU Remolque.** Unidad de control de remolque.
- **ECU Conmutador.** Unidad de control de conmutador (direccionales y botoneras en volante).
- **ECU unidad de apertura.** Unidad de control de llaves para encendido remoto o detección de llave.

- **ECU Bloqueo de dirección.** Unidad de control de bloqueo de la columna de dirección.
- **ECU Confort.** Unidad de control de Confort (control de luces, alarma, sensores de lluvia, sensores de proximidad).

NOTA 1: Aparte de la comunicación CAN se tiene otro protocolo ligado al CAN-Bus como lo es el LIN-BUS, el cual usa velocidades de transmisión de 19.2 K baudios y funciona como esclavo de las unidades de control electrónicas de CAN Bus y puede ser utilizado para los sensores o actuadores de baja relevancia.

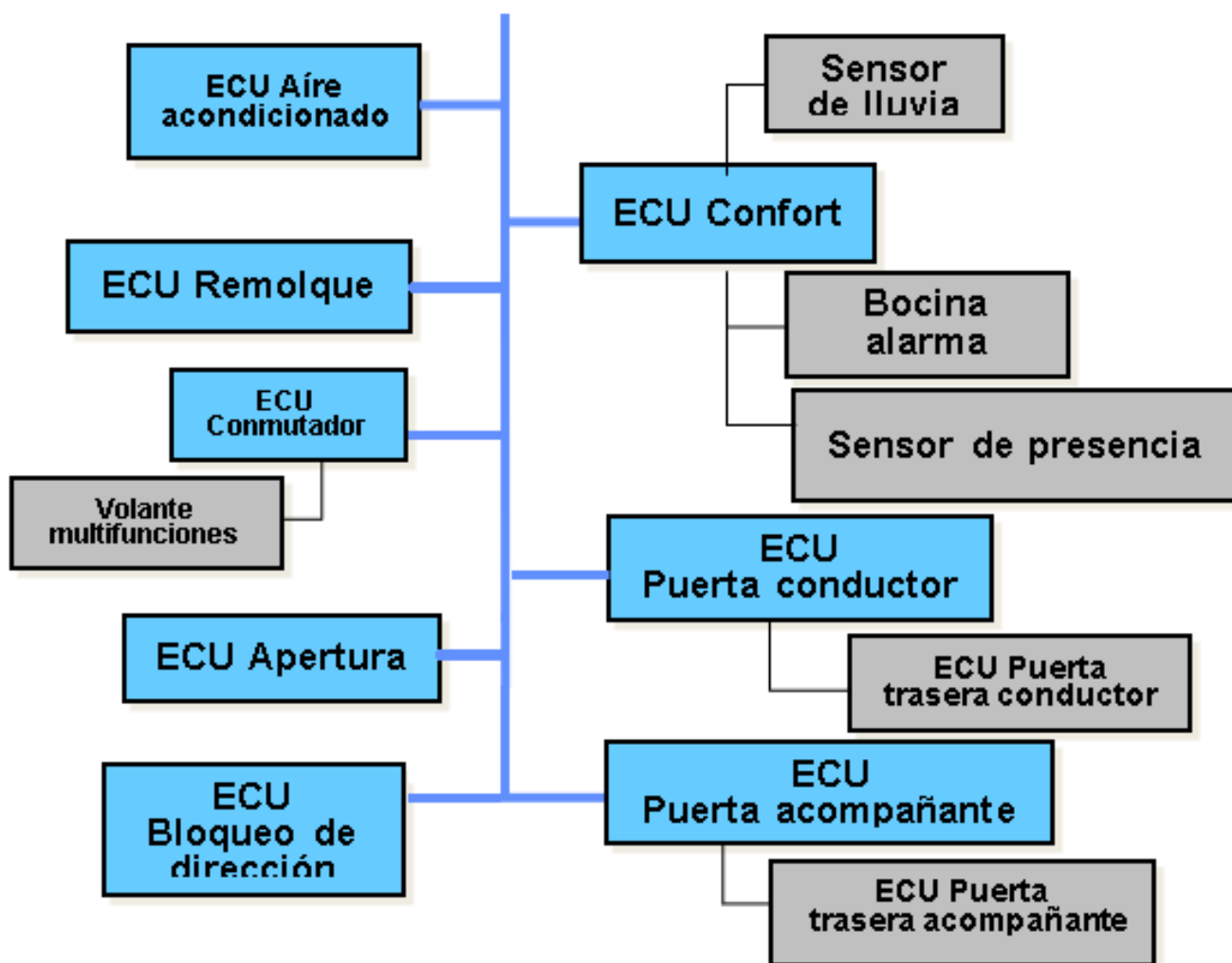


Figura 2.25: Red de comunicación del BUS de Confort.

- **CAN Infotainment.** Bus de comunicación de 500 K baudios y en el cual van montadas las unidades de control que brindan entretenimiento dentro del Automóvil.

Las unidades de control montadas en este Bus son las siguientes:

- **ECU Radio.** Unidad de control de Radio para sintonía de estaciones de radio o reproductores de música.
- **ECU Media.** Unidad de control de media para conexión de sistemas de almacenamiento masivo o teléfonos vía USB.
- **ECU Conectividad.** Unidad de control para conectividad vía GSM/GPRS.
- **ECU Amplificador.** Amplificador de sonido con 8 bocinas y 1 subwoofer.
- **ECU Radio/Navegación.** Variante de la unidad de control de radio con sistema de navegación y radio satelital incluido.

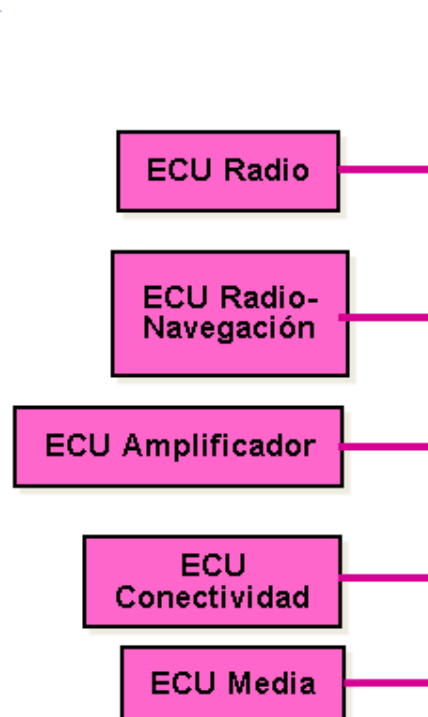


Figura 2.26: Red de comunicación del BUS de Infotainment.

- **CAN Propulsión.** Bus de comunicación de 500 K baudios y en el cual van montadas las unidades de control que brindan propulsión y seguridad en el Auto.

Las unidades de control montadas en este Bus son las siguientes:

- **ECU Motor.** Unidad de control de Motor es la encargada de hacer los ajustes por medio de sus actuadores y sensores dependiendo del tipo de manejo que se tenga.
- **ECU Transmisión.** Unidad de control de caja de velocidades encargada de hacer los cambios electrónicamente dependiendo del tipo de manejo.
- **ECU Palanca de transmisión.** Palanca de velocidades con comunicación CAN, esta palanca solo se usa en las cajas tipo doble embrague.
- **ECU Airbag.** Unidad de control del sistema de seguridad de bolsas de Aire.
- **ECU ABS/ESP.** Unidad de control de frenos con sistemas de sistemas para frenados.
- **ECU Angulo de giro.** Unidad de control del censado de ángulo de giro.
- **ECU Aparcamiento.** Unidad de control para la ayuda de estacionamiento.

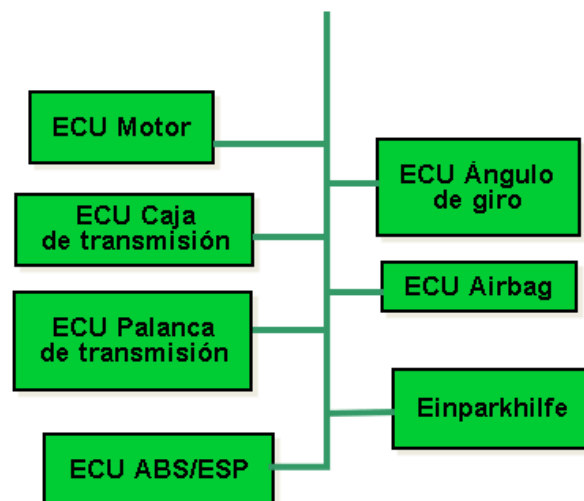


Figura 2.27: Red de comunicación del BUS de Propulsión.

- **CAN Clúster.** Bus de comunicación de 500 K baudios y en el cual va montado la unidad que comúnmente conocemos como cuadro de instrumentos o cuadrante. Las unidades de control montadas en este Bus es la siguiente:
 - **Clúster.** Unidad de control que muestra la información del vehículo y testigos de posibles averías en el auto.

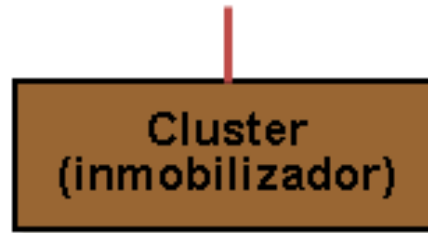


Figura 2.28: Red de comunicación del BUS de Clúster.

Dentro de esta unidad de control existe la función de inmovilizador la cual es encargada de la seguridad de arranque del auto por medio de una adaptación por medio de sistemas informáticos entre Llaves-ECU Motor-ECU Bloqueo de dirección.

2.6. CAN FD

CAN FD es la abreviación de **Controller Area Network Flexible Data-Rate** por sus siglas en inglés. Es un protocolo de comunicación de datos o también conocido como **protocolo CAN FD**, comúnmente utilizado para transmitir datos de sensores e información de control.

El **protocolo CAN FD** se usa típicamente en vehículos modernos de alto rendimiento. Por lo tanto, algunos expertos explican CAN FD como un Lamborghini entre los Renault. CAN FD es una extensión del CAN Bus clásico (**ISO 11898-1**). El **protocolo CAN FD** fue desarrollado en 2011 y lanzado por Bosch en 2012. El motivo para desarrollar el protocolo fue la creciente necesidad de una transferencia de datos más rápida y, por lo tanto, se desarrolló para ser capaz de transferir datos a una velocidad hasta 5 veces más rápida y con tramas/mensajes más grandes.

El **protocolo CAN FD** está diseñado para recibir y transmitir de manera eficiente datos de sensores, controlar comandos y detectar errores de datos entre dispositivos de sensores electrónicos y micro-controladores CAN FD. Al principio, el protocolo se desarrolló para vehículos de alto rendimiento, sin embargo, debido a su capacidad mejorada de comunicación de datos, lo veremos en muchas otras aplicaciones, como robótica, automatización industrial, vehículos marinos, equipos médicos, etcétera.

2.6.1. Beneficios del CAN FD

Hay grandes beneficios de implementar el **protocolo CAN FD**. Mencionaremos tres beneficios principales a continuación:

- Mayor confiabilidad.
- Mayor velocidad.
- Mayor longitud de bytes de datos.

En términos generales, CAN FD puede mejorar el ancho de banda de la red de 3 a 8 veces más que un CAN clásico. Como ya mencionamos antes, CAN FD puede manejar más datos a un ritmo más rápido. Algunas de las aplicaciones comunes se pueden ver a continuación: Vehículos Eléctricos (VE), camiones y autobuses, conducción segura en modos autónomos.

2.6.2. CAN FD vs CAN

La principal diferencia entre CAN FD y CAN son los datos flexibles. Significa que al usar CAN FD, las unidades de control electrónico (ECU) pueden cambiar fácilmente a diferentes velocidades de datos con tamaños de mensaje más grandes o más pequeños. CAN FD ofrece una velocidad de datos más rápida y más mejoras en la capacidad de datos que el CAN clásico. Esto generalmente se aplica a las empresas que necesitan usar muchos datos, donde el CAN clásico puede causar algunos problemas. CAN FD se convierte en la solución para muchas empresas, debido a sus necesidades específicas de transferir más cantidades de datos.

Capítulo 3

Etapa física del CAN-bus

En este capítulo se describirá brevemente la etapa física del CAN-bus.

3.1. Resistencia de cierre de CAN-HIGH

Las unidades de control de alta velocidad (500 K baudios) tienen una resistencia de cierre entre las terminales de CAN High y CAN Low, ésta puede ser medida directamente con un Ohmetro entre CAN High y CAN Low. La resistencia total de cierre del bus es medida con el auto sin alimentación de batería.

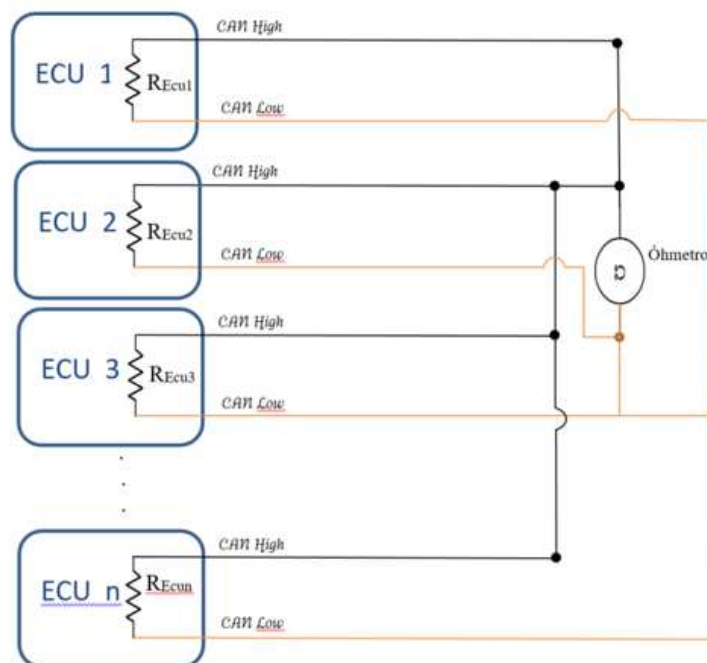


Figura 3.1: Circuito de resistencia de cierre en unidades de control.

donde R_{Ecu} es la resistencia interna de cierre de cada unidad de control dentro del bus.

Matemáticamente podemos calcularla de la resistencia total del bus de la siguiente manera:

$$R_{\text{CANH-CANL}} = \left(\frac{1}{\frac{1}{R_{\text{ECU}_1}} + \frac{1}{R_{\text{ECU}_2}} + \frac{1}{R_{\text{ECU}_3}} + \dots + \frac{1}{R_{\text{ECU}_n}}} \right) \quad (3.1)$$

donde R_{ECU} es la resistencia interna de cierre de cada unidad de control dentro del bus, la cual se obtiene el valor teórico dentro de la especificación de desarrollo de cada unidad de control.

Ejemplo. Tomando como ejemplo 4 unidades de control montadas en el CAN-bus.

Cada unidad de control montada en el CAN-bus tiene el siguiente valor de resistencia

$$R_{\text{ECU}_1} = 120.00\Omega \quad (3.2)$$

$$R_{\text{ECU}_2} = 2600.0\Omega \quad (3.3)$$

$$R_{\text{ECU}_3} = 2600.0\Omega \quad (3.4)$$

$$R_{\text{ECU}_4} = 120.00\Omega \quad (3.5)$$

Resolviendo con respecto a (3.1) se tiene

$$\begin{aligned} R_{\text{CANH-CANL}} &= \left(\frac{1}{\frac{1}{R_{\text{ECU}_1}} + \frac{1}{R_{\text{ECU}_2}} + \frac{1}{R_{\text{ECU}_3}} + \dots + \frac{1}{R_{\text{ECU}_n}}} \right) \\ &= \left(\frac{1}{\frac{1}{120.00\Omega} + \frac{1}{2600.0\Omega} + \frac{1}{2600.0\Omega} + \frac{1}{120.00\Omega}} \right) \\ &= \left(\frac{1}{0.00833\Omega + 0.000384\Omega + 0.000384\Omega + 0.0083\Omega} \right) \\ &= \left(\frac{1}{0.017369\Omega} \right) \\ &= 57.5730\Omega \end{aligned} \quad (3.6)$$

Los resultados obtenidos son ¹:

Resistencia	Debe ser	Es	Correcto/Incorrecto
$R_{\text{CANH-CANL}}$	57.5730	56.93	Correcto

Tabla 3.1: Medición de Resistencia total del Bus.

¹Ver **Apéndice A** para el llenado de datos para la prueba

Si los valores de resistencia del bus no coinciden con lo especificado en la información de desarrollo de la unidad de control, se deben de realizar mediciones de manera individual de cada unidad de control y así verificar cual está fuera de norma.

Esto se hace de la siguiente manera:

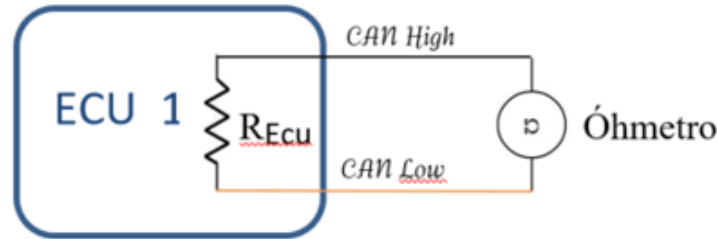


Figura 3.2: Circuito de resistencia de cierre en unidad de control.

3.1.1. Niveles de señal CAN High

La prueba se realiza utilizando una herramienta de la firma Vector llamada **CANScope High Speed** la cual es un osciloscopio de CAN-bus o en su defecto podemos utilizar un osciloscopio.

El diagrama de conexión es el siguiente:

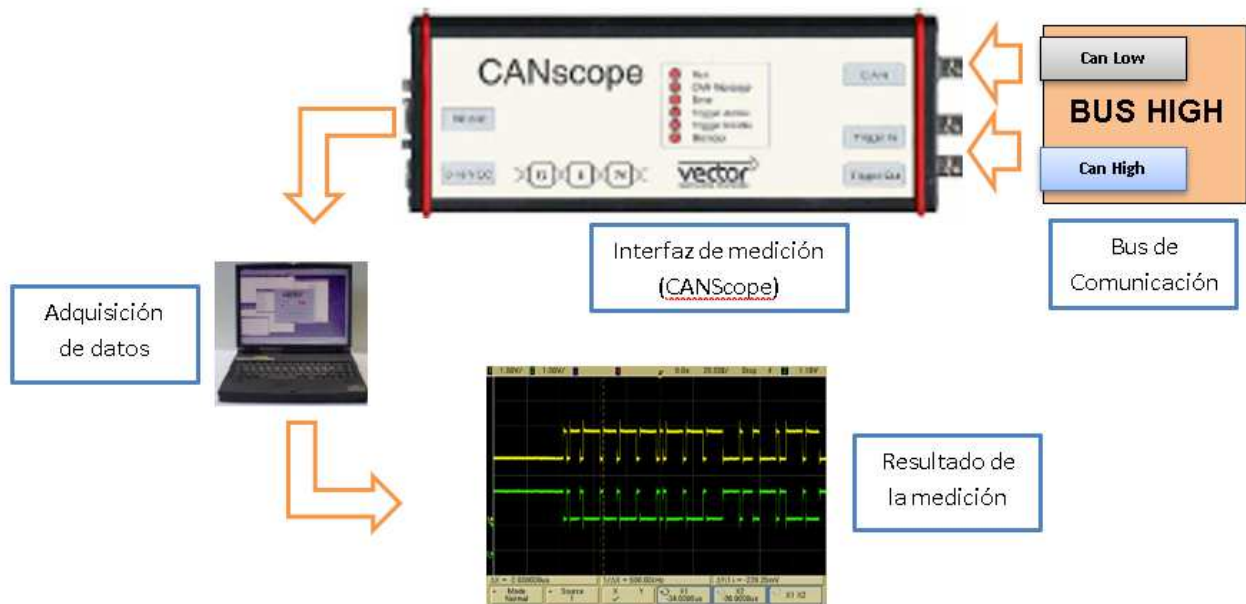


Figura 3.3: Diagrama a bloques de conexión para medición de los niveles de señal.

Los datos obtenidos son llenados en las siguientes tablas:

	Debe ser	Es	Correcto/Incorrecto
Ancho de pulso	1.998 . . . 2.002	2.00	Correcto

Tabla 3.2: Ancho de pulso CAN High Speed.

A continuación se muestran las señales obtenidas

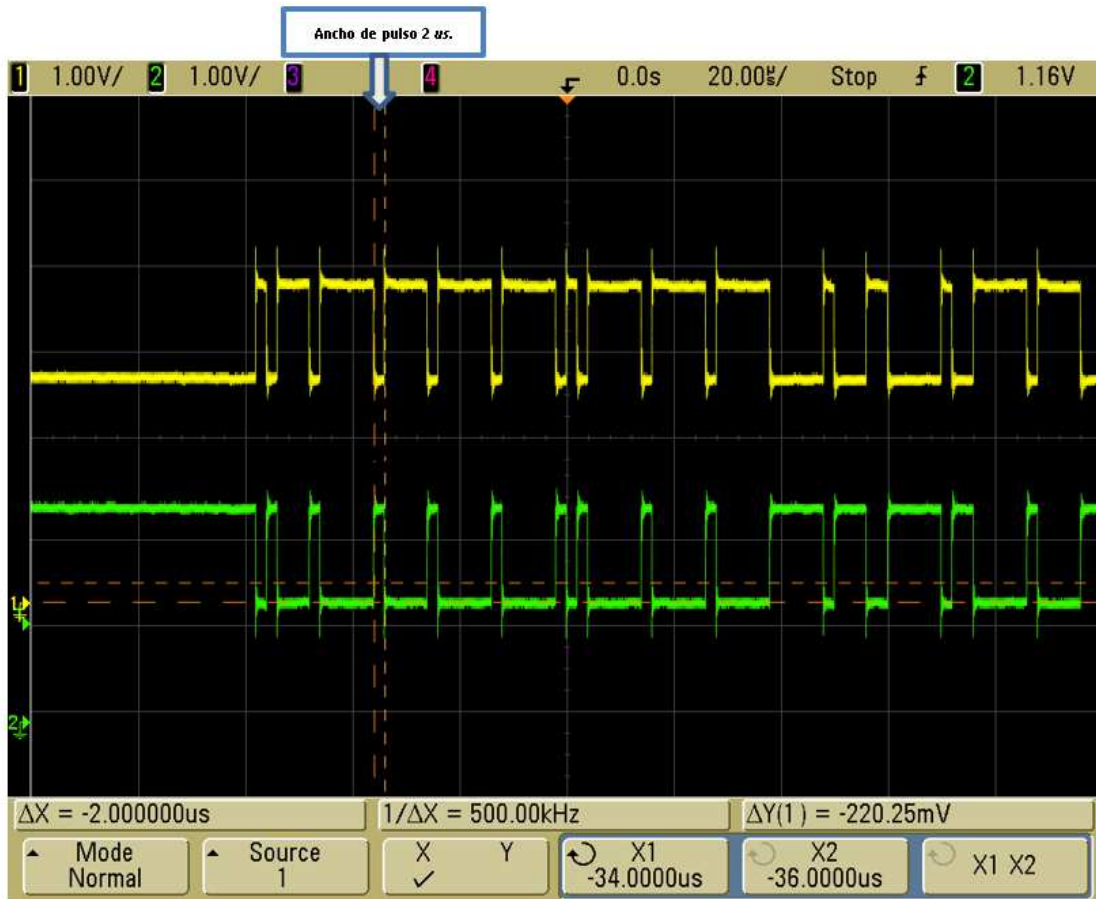


Figura 3.4: Ancho de pulso CAN High Speed (1.998, . . . , 2.002 μ s)

La tabla de mediciones es la siguiente

Nivel de señal	$UCAN_H$		Correcto/Incorrecto
	DEBE [V]	ES[V]	
Nivel de señal recesiva	2.40, . . . 2.6	2.50	Correcto
Nivel de señal dominante	2.75, . . . 4.5	3.55	Correcto

Tabla 3.3: Nivel de señal $UCAN_H$ para CAN High Speed

A continuación se muestran las señales obtenidas

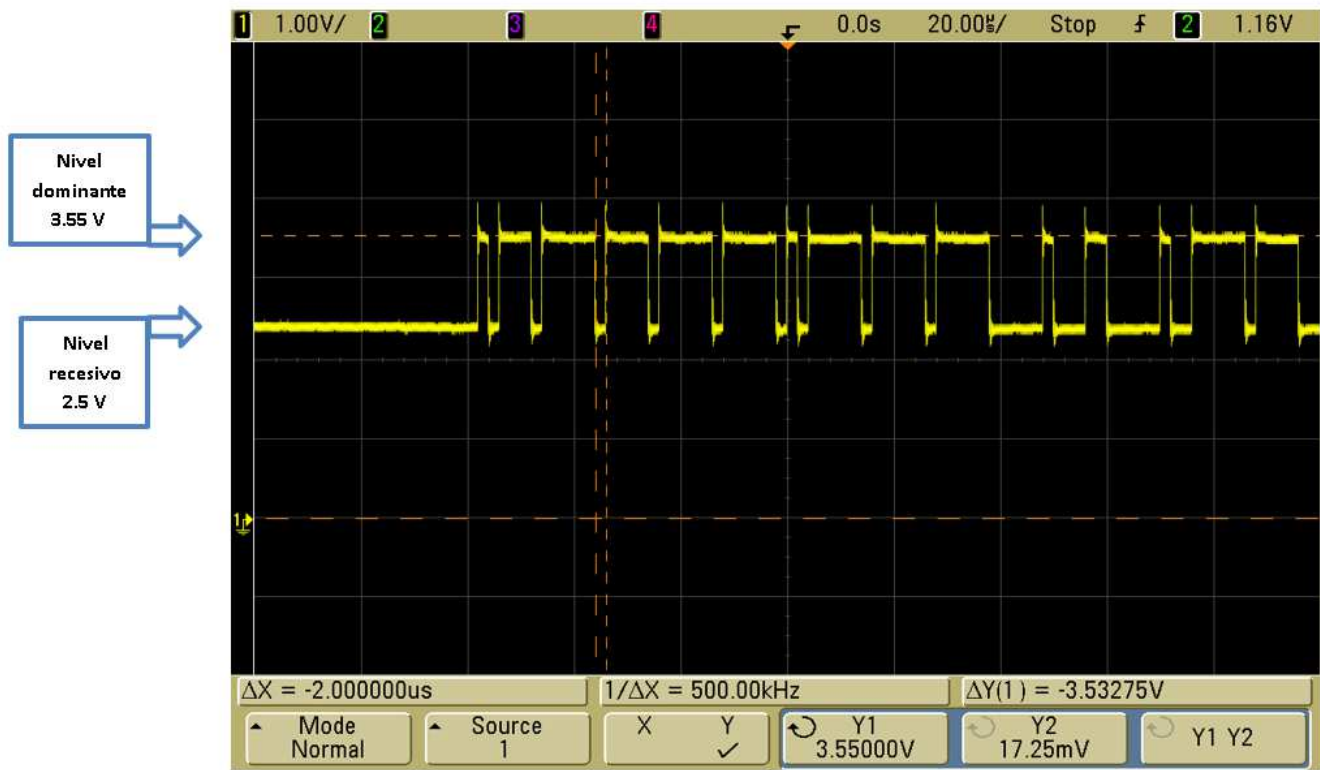


Figura 3.5: Nivel recesivo y dominante CAN H.

La tabla de mediciones es la siguiente

Nivel de señal	UCAN _L		Correcto/Incorrecto
	DEBE [V]	ES[V]	
Nivel de señal recesiva	2.40, ... 2.60	2.55	Correcto
Nivel de señal dominante	0.50, ... 2.25	1.41	Correcto

Tabla 3.4: Nivel de señal UCAN_L para CAN High Speed

A continuación se muestran las señales obtenidas

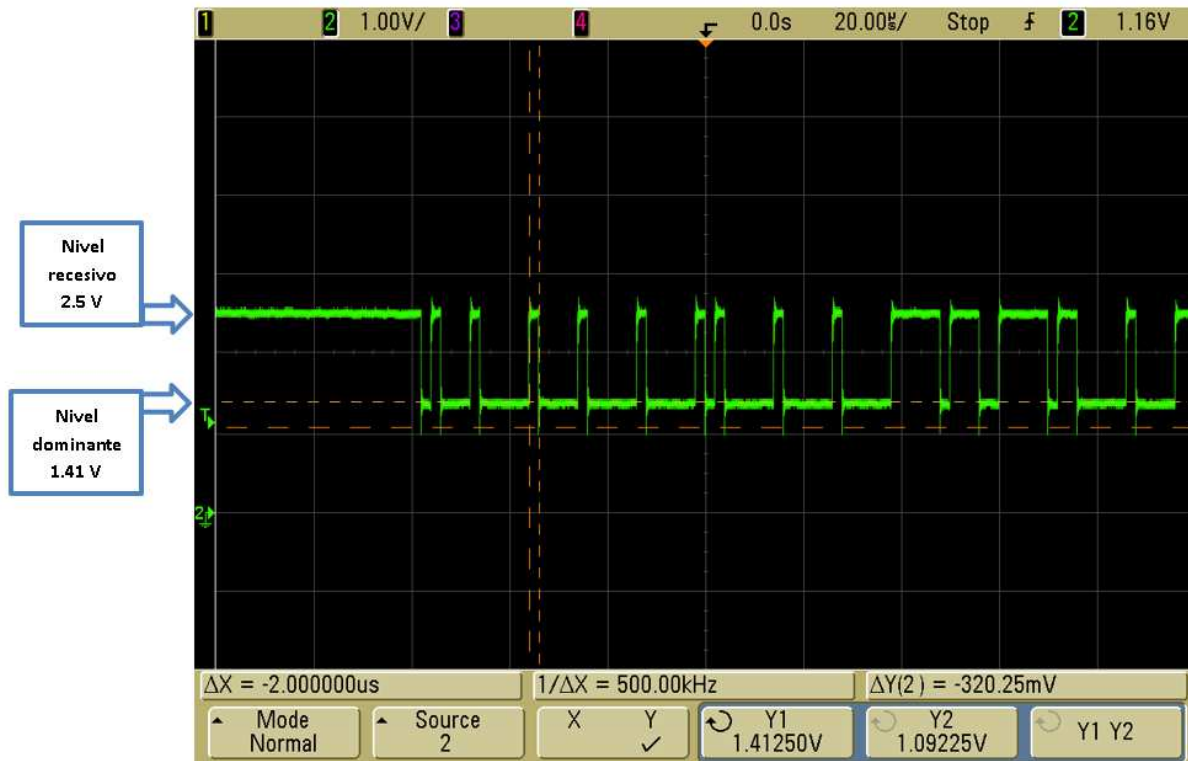


Figura 3.6: Nivel recesivo y dominante CAN L.

Ver el **Apéndice B** para el llenado de datos para la prueba.

3.1.2. Pruebas de comunicación

Cuando se reconoce una causa de **despertar (Wake up)** las unidades de control empiezan a transmitir datos, debido a que la unidad que sufre el **Wake up** envía un mensaje al Bus de comunicación llamado **alive message**, este mensaje detona la comunicación CAN de todo el Bus y depende de las funciones realizadas es el tiempo de activación. Existen diferentes causas de despertar de las unidades de control, dependiendo de la función que tengan o la reacción que tengan bajo determinada función. Algunos **Wake up** son los siguientes:

- Activación de switch de encendido.
- Encendido de luces.
- Encendido de intermitentes.

Al existir la comunicación CAN en el Bus se forma el denominado anillo de comunicación es aquí que cada mensaje de administración de la red pone su bit de ring en **1** y se forma el anillo lógico. Después

de que la función por la cual despertó el Bus deja de ser utilizada y no existe alguna otra función como consecuencia de la anterior que mantenga la comunicación activa las unidades de control ponen su **Sleep-Bit** en **1** indicando con este que no tienen ningún otro requerimiento que las mantenga activas y puede ir al denominado **bus de tranquilidad**.

3.1.2.1. Estabilidad del anillo de comunicación

Mediante la prueba de estabilidad del anillo de comunicación se comprueba la correcta comunicación de todos los participantes montados en el sistema. El anillo tiene que permanecer estable durante la medición completa, la prueba se debe de realizar con el switch del auto encendido, sin ningún funcionamiento de emergencia activado y con ninguna trama de error en el Bus. El examen se realiza con una configuración estándar de Software CANoe y una interfaz denominada **CANCase** para la adquisición de datos, la prueba debe ser durante un periodo de 60 minutos. El diagrama de conexión es el siguiente:



Figura 3.7: Diagrama a bloques de conexión para medición de estabilidad del anillo.

Los datos son obtenidos por el programa CANoe se describen de la siguiente manera:

	Unidad 1	Unidad 2	Unidad 3	Unidad 4
Dirección	RX	RX	RX	RX
ID	X1	X2	X3	X4
DLC	8	8	8	8
BYTE 0	11	16	1B	10
BYTE 1	0 1	0 1	0 1	0 1
BYTE 2	11	13	12	20
BYTE 3	10	11	15	10
BYTE 4	10	21	22	21
BYTE 5	24	20	10	20
BYTE 6	10	10	20	20
BYTE 7	10	10	10	10

Tabla 3.5: Datos obtenidos por el programa CANoe.

En el **BYTE 1 bit 1** nos muestra la permanencia del anillo de comunicación, lo podemos observar en el resultado anterior que el anillo permanece constante. Gráficamente lo podemos ver de la siguiente manera.

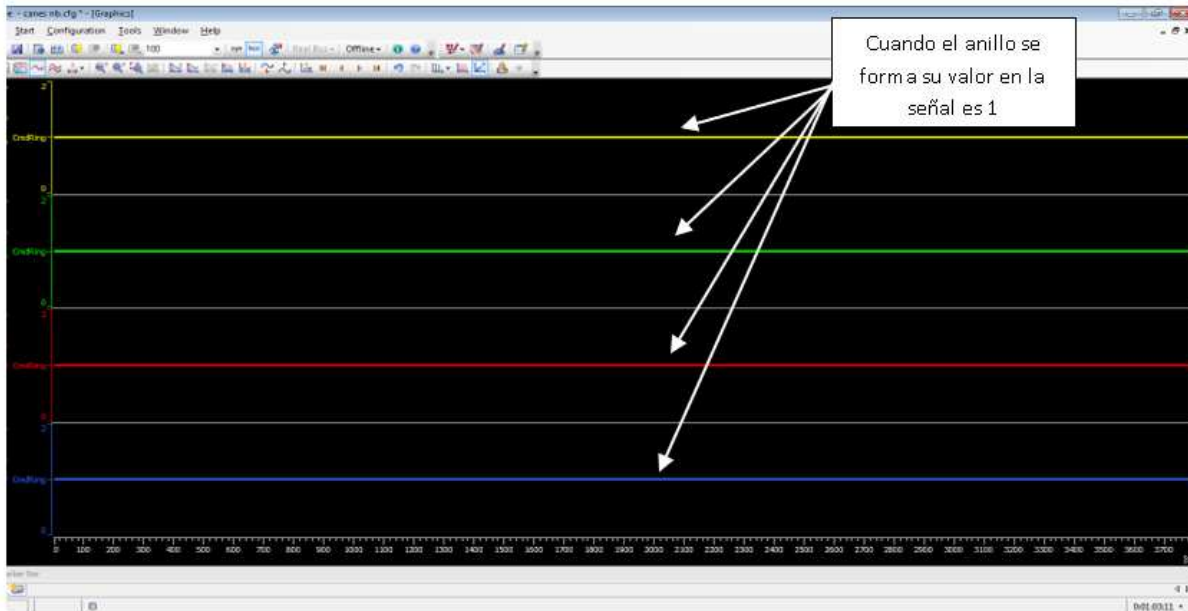


Figura 3.8: Gráfica de CANoe de permanencia del anillo en la prueba de estabilidad.

La ventana de Bus statistic nos muestra los datos de **Errorframes** y **Alive Messages (Ext. Data)**

	CAN 1	CAN 2	CAN 3	CAN 4
Bus load [%]	37.75	0.00	0.00	0.00
Peak load [%]	38.90	0.00	0.00	0.00
Std. Data [fr/s]	22	0	0	0
Std. Data [total]	80760	0	0	0
Ext. Data [fr/s]	0	0	0	0
Ext. Data [total]	0	0	0	0
Std. Remote [fr/s]	0	0	0	0
Std. Remote [total]	0	0	0	0
Ext. Remote [fr/s]	0	0	0	0
Ext. Remote [total]	0	0	0	0
Error frame [fr/s]	0	0	0	0
Error frames [total]	0	0	0	0
Chip state	Active	Active	Active	Active

Figura 3.9: Imagen de la ventana de estadísticas del bus.

Los datos de la prueba de estabilidad del anillo de comunicación se presentan en la siguiente tabla

	Debe ser	Es	Correcto/Incorrecto
Errorframes	0	0	Correcto
Alive Messages	0	0	Correcto

Tabla 3.6: Prueba de estabilidad del anillo de comunicación.

Ver el **Apéndice B** para el llenado de los datos para la prueba

3.1.2.2. Tranquilidad del BUS

La tranquilidad de Bus es de suma importancia para los automóviles con comunicación CAN ya que si no se tiene un correcto funcionamiento las unidades de control pueden tener **Wake up** no definidos y tener consumos de corriente en cada acontecer de despertar acabando con la fuente de alimentación del automóvil (batería).

Bus de Comunicación	Consumo de corriente
High Speed	400 mA
Low Speed	500 mA

Tabla 3.7: Consumo de corriente de Low Speed y High Speed.

Con esta prueba se comprueba la estabilidad de la tranquilidad bus de comunicación durante un período largo. El Bus completo debe ser llevado a tranquilidad (NO COMUNICACIÓN CAN) y todas las funciones que despiertan el Bus como, por ejemplo:

- Luces exterior.
- Intensidad de iluminación interior.
- Intermitentes.
- Direccionales.
- Switch de encendido cerrado
- Radio encendido

Deben de estar desactivadas ya que el tener alguna de estas u otras funciones activas nos puede ocasionar el despertar del bus después de tiempos definidos en el desarrollo de cada unidad de control. Es por eso que el Bus debe de cumplir con la tranquilidad durante un periodo mínimo de 60 minutos para que la prueba sea satisfactoria. La realización de esta prueba se realiza mediante una configuración estándar de CANoe para averiguar qué participante (unidad de control) vuelve a despertar Bus sin una función definida.

El diagrama de conexión para la Medición de tranquilidad del bus es el siguiente:



Figura 3.10: Diagrama a bloques de conexión para medición de tranquilidad del Bus.

Realizamos una configuración estándar para poder guardar los datos del Bus en caso que alguna unidad de control despierte después de apagar el switch de encendido del automóvil.

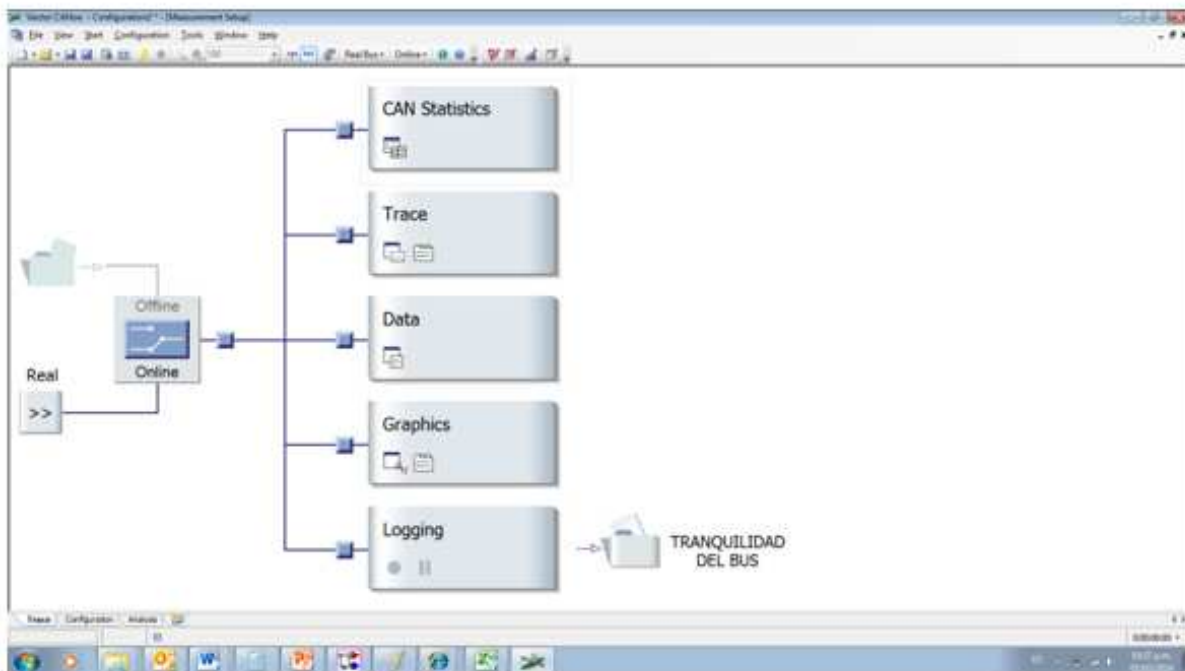


Figura 3.11: Configuración estándar de CANoe.

Después de realizar la medición de por lo menos de 60 minutos de duración obtenemos en la ventana de estadísticas de Bus el último porcentaje de Bus load para que la prueba sea calificada correctamente el Bus load debe de ser 0%.

	CAN 1	CAN 2	CAN 3	CAN 4
Bus load [%]	0.00	0.00	0.00	0.00
Peak load [%]	0.00	0.00	0.00	0.00
Std. Data [fr/s]	0	0	0	0
Std. Data [total]	0	0	0	0
Ext. Data [fr/s]	0	0	0	0
Ext. Data [total]	0	0	0	0
Std. Remote [fr/s]	0	0	0	0
Std. Remote [total]	0	0	0	0
Ext. Remote [fr/s]	0	0	0	0
Ext. Remote [total]	0	0	0	0
Error frame [fr/s]	0	0	0	0
Error frames [total]	0	0	0	0
Chip state	Active	Active	Active	Active

Figura 3.12: Imagen de la ventana de estadísticas del bus.

La tabla de valores de Tranquilidad de Bus se presenta a continuación:

	Constante	No constante	Correcto/Incorrecto
Tranquilidad del Bus	SI		Correcto

Tabla 3.8: Tranquilidad de Bus.

Ver **Apéndice C** para el llenado de los datos para la prueba

***Nota:** La prueba es aceptada como correcta si tiene una duración mínima de 60 minutos, pero es recomendable realizar durante 8 horas (durante la noche).*

3.1.2.3. Formación del anillo de comunicación

Para esta prueba se debe analizar primero que ninguna unidad de control tenga algún error de comunicación CAN en su memoria de averías. Si se llegara a tener una falla estática por algún problema de adaptación se debe de documentar antes de iniciar la prueba.

Es necesario utilizar una interfaz para activar/desactivar el switch de encendido del auto, esto lo realizamos por medio de un **CANExtender** el cual es un equipo controlado por CAN Bus mediante el software CANoe que cuenta con entradas y salidas analógicas y pueden ser utilizadas para el fin requerido.



Figura 3.13: CANExtender-AA.

La prueba consiste en hacer varios ciclos de activación y desactivación del switch de encendido del automóvil tomando como premisa la espera de la tranquilidad del Bus para volver a activar el switch de encendido y así analizar por medio de un CANTrace la formación correcta del anillo de comunicación, deben todas las unidades de control responder el mismo número de ciclos de activación y antes de iniciar con la prueba se debe de sacar un diagnóstico de la prueba y otro diagnóstico después de la prueba.

El diagrama de conexión para la medición de la formación del anillo de comunicaciones el siguiente:



Figura 3.14: Diagrama a bloques para medición de anillo de comunicación.

La configuración utilizada para esta prueba se llama **Análisis de anillo** la cual consta de una etapa de simulación de activación/desactivación de switch de encendido

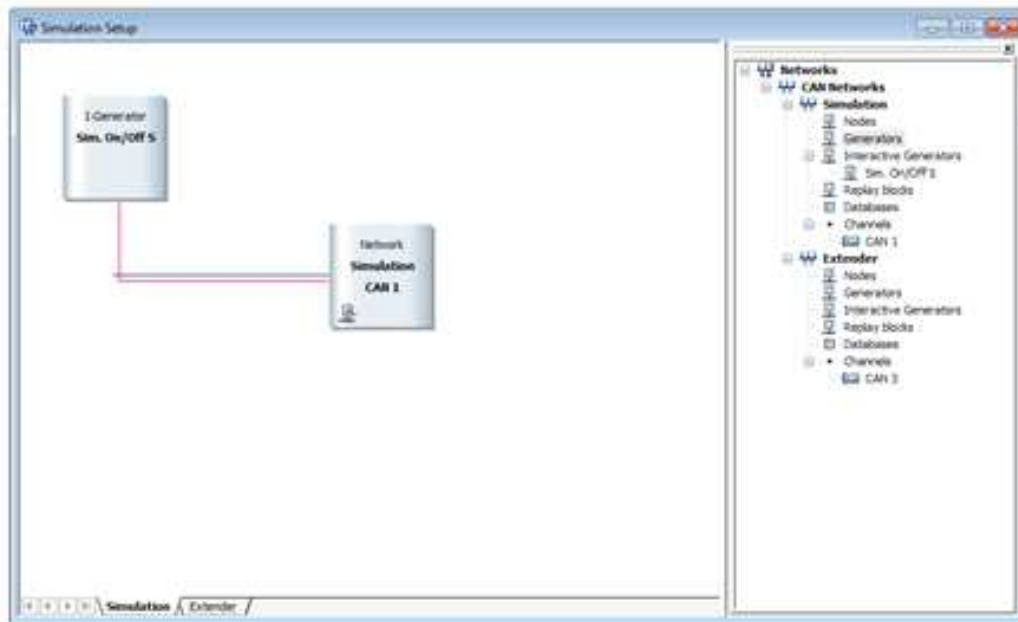


Figura 3.15: Simulación de switch de encendido.

y una segunda etapa de adquisición de datos.

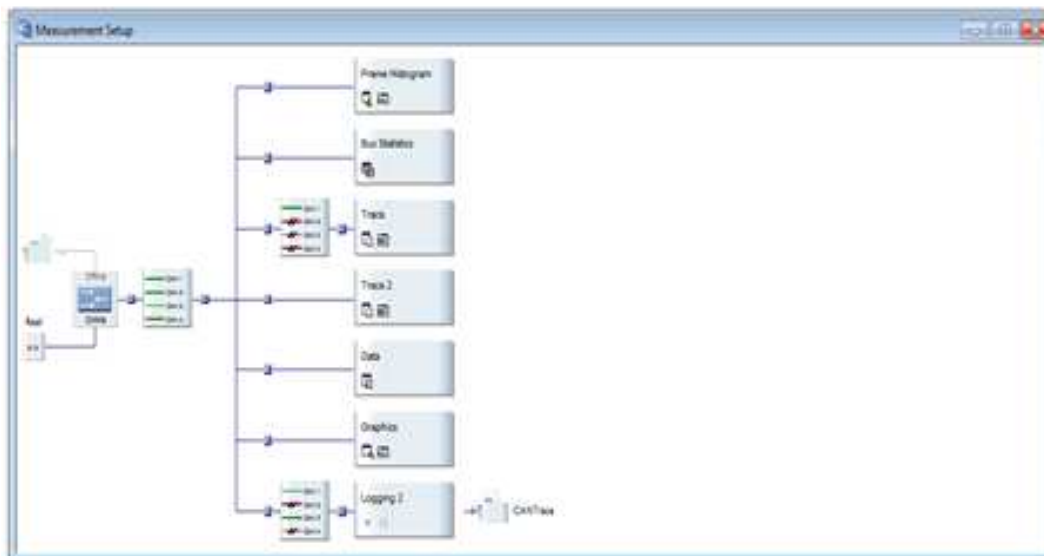


Figura 3.16: Adquisición de datos.

Analizando el resultado del CANTrace generador por la configuración podemos observar el comportamiento del bit de formación del anillo de comunicación de las 4 unidades de control medidas.

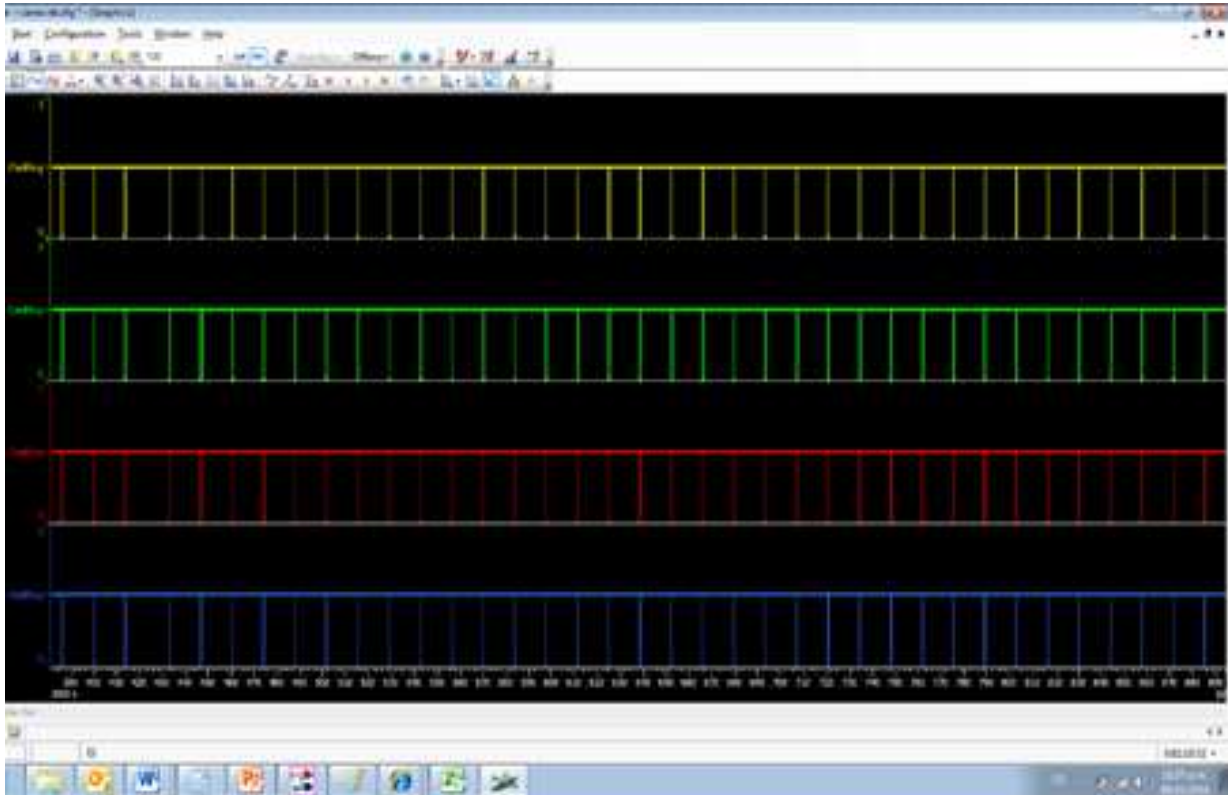


Figura 3.17: Bit de formación del anillo de comunicación.

Esta prueba se debe realizar durante un período de 8 horas (durante la noche). Después de terminar el Test se debe de sacar un diagnóstico de todas las unidades de control donde se corrobora que ninguna unidad reporto alguna falla de comunicación.

Los resultados de la prueba se muestran en la siguiente tabla:

	Debe se	Es	Correcto/Incorrecto
Fallas antes de la prueba	ninguna	ninguna	Correcto
Fallas después de la prueba	ninguna	ninguna	Correcto
Respondieron las unidades de control	Todas/Siempre	Todas/Siempre	Correcto

Tabla 3.9: Formación del anillo de comunicación.

Ver **Apéndice D** para el llenado de los datos para la prueba

Capítulo 4

Diagnóstico

En este capítulo se describe el proceso de diagnóstico

4.1. Diagnóstico OBD

OBD quiere decir por sus siglas en inglés **On board Diagnostics**, lo que quiere decir en español, dentro del contexto, **Sistema de Diagnósticos a Bordo**; el sistema consiste en la utilidad de reporte y auto-diagnóstico del vehículo, entregando información acerca del estado del automóvil y sus sistemas. Por ejemplo, en sus formas más sencillas, en los automóviles antiguos se podían encontrar pequeñas ampollitas indicadoras de fallos o alertas, como un bajo nivel de aceite. El sistema OBD no es más que una sofisticación del concepto, en donde el automóvil mediante un sistema de códigos informa lo que anda mal. Los tipos de OBD son OBD I y OBD II.

4.1.1. OBD I

Fue el primer estándar ampliamente difundido, y su limitación consistía en que entregaba muy pocos códigos, o sea información demasiado general para poder identificar el problema en cuestión. Surge en 1987 con las especificaciones requeridas en el estado de California para los automóviles nuevos. Antes de esta fecha, existían diferentes sistemas y propuestas para sistemas de diagnóstico.

4.1.2. OBD II

Surge nuevamente como requerimiento de estándar para vehículos nuevos en California, y desde 1996 que es un requerimiento para todos los automóviles vendidos en Estados Unidos. La principal diferencia con el sistema anterior, es que el OBD II indica muchos más códigos, indicando específicamente el sistema involucrado en el fallo y otros detalles. Para poder leer o interpretar estos códigos, se emplea un escáner de diagnóstico, que puede ser un sistema independiente (stand alone), u operado a través de interfaz USB mediante un notebook. En el caso de no tener un aparato para el **scan**, la mayoría de los OBD en los automóviles pueden entregar los códigos de fallo mediante alguna luz en el panel.

4.2. Plataforma de diagnóstico

El diagnosticador utilizado para el análisis de fallas de las unidades de control dentro de las diferentes marcas automotrices es exclusivo de cada marca. En este caso lo representaremos de manera gráfica por cuestiones de confidencialidad.



(a) *Interfaz de diagnóstico*



(b) *Puerto de diagnóstico OBD*

Figura 4.1: Centrales de control en un Automóvil.

El software de diagnóstico o escáner automotriz por sí solo no puede trabajar ya que necesita de una interfaz para comunicarse con el vehículo, este equipo se conecta por medio del puerto OBD localizado en el automóvil y tiene la capacidad de realizar:

- Funciones guiadas.
- Localización de averías.
- Codificaciones.
- Adaptaciones.
- Ajustes básicos.

Es compatible con diferentes protocolos de diagnóstico del auto como lo es:

- La línea K (ISO9141-2)
- CAN de alta velocidad de la Norma (ISO 11898-2)
- CAN baja velocidad de la Norma (ISO 11898-3)
- J1850 (SAE J1850)

Los protocolos de los vehículos son manejados directamente en las plataformas de cada marca esto garantiza tiempos de respuesta rápidos y un comportamiento fiable en tiempo real independientemente de la PC ó sistema operativo. El software de diagnóstico o escáner se sincroniza con el vehículo por medio de la interfaz de diagnóstico y una vez que se hace la conexión Software-Automóvil, las unidades de control o Gateway son las encargadas de enviar los datos al sistema de diagnóstico por medio de requerimientos específicas a cada una de ellas.

4.2.1. Funcionamiento de software de diagnóstico

Al iniciar la plataforma de diagnóstico se reconoce a primera vista el tipo de arquitectura del vehículo (Proyecto) o automóvil a diagnosticar.

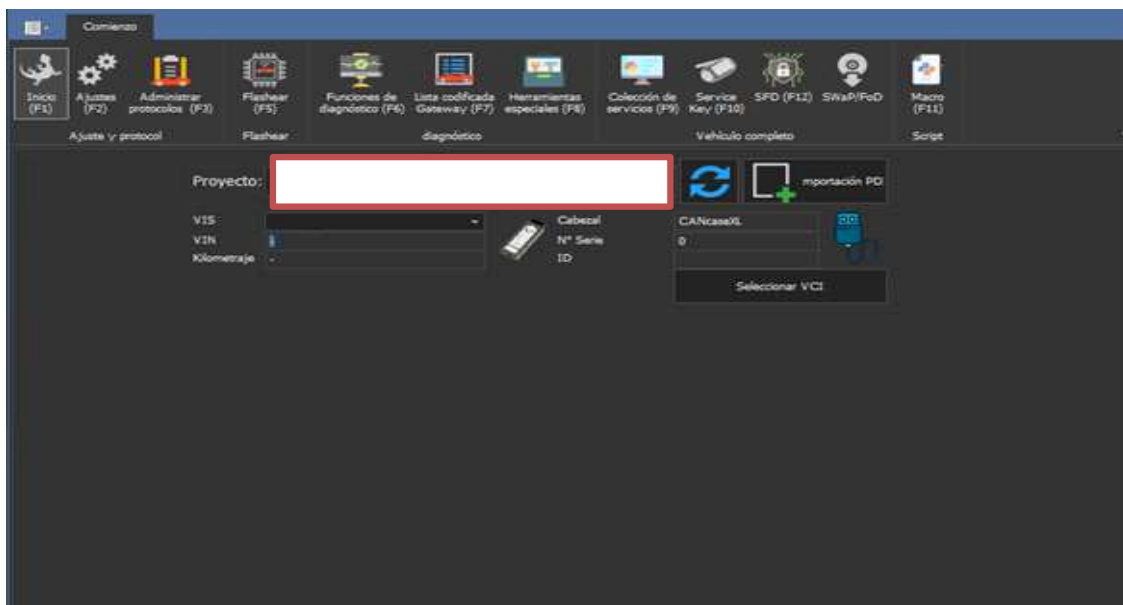


Figura 4.2: Pantalla principal de plataforma de diagnóstico.

Nota: En caso de que la arquitectura no sea la correcta puede llevarnos al no diagnóstico de algunas unidades de control que a pesar de estar montadas en el auto no podrán ser visualizadas ya que estamos en una arquitectura diferente a la del vehículo a diagnosticar.

Ahora que tenemos la arquitectura correcta seleccionada pasamos a la ventana que nos mostrara las unidades de control montadas en el vehículo que es tomada de una lista de montaje del Gateway (unidad de control de acceso al diagnóstico). Seleccionando la opción (Lista codificada de Gateway)



Figura 4.3: Softkey de la lista codificada de Gateway.

Posteriormente nos saldrá la lista de montaje de las unidades de control en auto

Identificador	Nombre de la unidad de control	Protocolo de comunicación	Estado
0001	Electrónica del motor	CAN-Trip de potencia	OK
0002	Regulación de la batería memoria adicional I	CAN-Comfort	No instalado
0003	Actuador para supresión de resonancias	CAN-Comfort	No instalado
0004	Actuador para sonido exterior	CAN-Comfort	No instalado
0005	Electrónica del cambio	CAN-Trip de potencia	OK
0006	Electrónica de los frenos	CAN-Chassis	OK
0007	Ajuste de asiento lado acompañante	Irreconocido	No instalado
0008	Electrónica del eje control de estabilidad	CAN-Comfort	OK
0009	Control de dirección electrónica	CAN-Comfort	OK
0010	ACC	CAN-Electrónica	OK
0011	Electrónica de la antipinchadura de ruedas	Irreconocido	No instalado
0012	Asiento	CAN-Trip de potencia	OK
0013	Electrónica de la columna de dirección	Irreconocido	No instalado
0014	Cambio de instrumentos	CAN-Comfort	OK
0015	Calificación adicional/independiente	Irreconocido	No instalado
0016	Asistencia lateral carrilera	Irreconocido	No instalado
0017	Electrónica de la tracción total	Irreconocido	No instalado
0018	Servofreno	Irreconocido	No instalado
0019	Mirador de la luz izquierda	Irreconocido	No instalado
0020	Bi-xenon columna dirección	CAN-Comfort	OK
0021	Sistema electrónico de bloqueo	Irreconocido	No instalado
0022	Regulación del asiento del lado del conductor	CAN-Comfort	OK

Figura 4.4: Lista de montaje en el Gateway.

La lista de montaje muestra todas las unidades que son diagnosticables por medio de CAN Bus (System), así como también que unidades tiene alguna falla en su memoria de averías.

4.2.2. Lectura de fallas en las unidades de control

Para hacer el diagnóstico de manera independiente de las unidades de control debemos seleccionar la unidad que deseamos diagnosticar con un **doble clic**, en este caso lo vamos a hacer con la unidad de Motor (0001-Electrónica de Motor).

Identificador	Nombre de la unidad de control	Protocolo de comunicación	Estado
0001	Electrónica del motor	CAN-Trip de potencia	OK
0002	Regulación de la batería memoria adicional I	CAN-Comfort	No instalado
0003	Actuador para supresión de resonancias	CAN-Comfort	No instalado
0004	Actuador para sonido exterior	CAN-Comfort	No instalado
0005	Electrónica del cambio	CAN-Trip de potencia	OK
0006	Electrónica de los frenos	CAN-Chassis	OK
0007	Ajuste de asiento lado acompañante	Irreconocido	No instalado
0008	Electrónica del eje control de estabilidad	CAN-Comfort	OK
0009	Control de dirección electrónica	CAN-Comfort	OK
0010	ACC	CAN-Electrónica	OK
0011	Electrónica de la antipinchadura de ruedas	Irreconocido	No instalado
0012	Asiento	CAN-Trip de potencia	OK
0013	Electrónica de la columna de dirección	Irreconocido	No instalado
0014	Cambio de instrumentos	CAN-Comfort	OK
0015	Calificación adicional/independiente	Irreconocido	No instalado
0016	Asistencia lateral carrilera	Irreconocido	No instalado
0017	Electrónica de la tracción total	Irreconocido	No instalado
0018	Servofreno	Irreconocido	No instalado
0019	Mirador de la luz izquierda	Irreconocido	No instalado
0020	Bi-xenon columna dirección	CAN-Comfort	OK
0021	Sistema electrónico de bloqueo	Irreconocido	No instalado
0022	Regulación del asiento del lado del conductor	CAN-Comfort	OK

Figura 4.5: Selección de unidad de control.

Posteriormente nos desplegará la siguiente ventana donde nos aparecen diferentes opciones a realizar dentro de esta unidad de control:



Figura 4.6: Lectura de memoria de eventos.

En este caso realizaremos la opción leer memoria de eventos en la cual podemos observar que se encuentran 4 fallas:

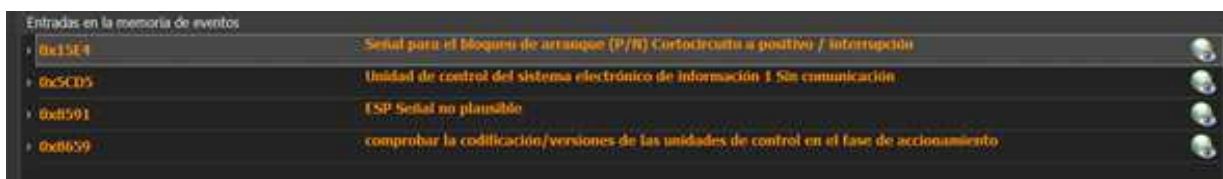


Figura 4.7: Eventos en la unidad de control.

Nota: Estas fallas fueron generadas para fines de enseñanza y se pudiera ver como es que son desplegadas en la memoria de eventos de la unidad de control de motor.

4.2.3. Borrado de fallas de la unidad de control

Ahora realizaremos el borrado de estas fallas, para eso damos en la función atrás

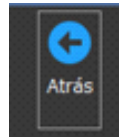


Figura 4.8: Softkey regresar ventana anterior.

La cual nos regresa a la pantalla principal de diagnóstico de la unidad de motor. Ahora seleccionamos borrar memoria de eventos.

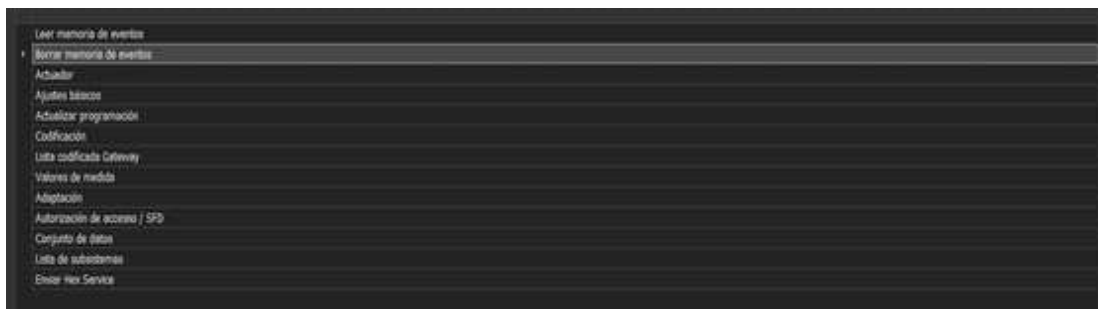


Figura 4.9: Borrar memoria de eventos.

Posterior al borrado de memoria de eventos realizamos de nuevo la lectura de memoria de eventos para corroborar que ya no existen fallas en la memoria de eventos.



Figura 4.10: Borrar memoria de eventos.

Al rectificar la memoria de eventos podemos visualizar que se encuentra vacía, ya que las fallas anteriormente mostradas fueron solucionadas y borradas.

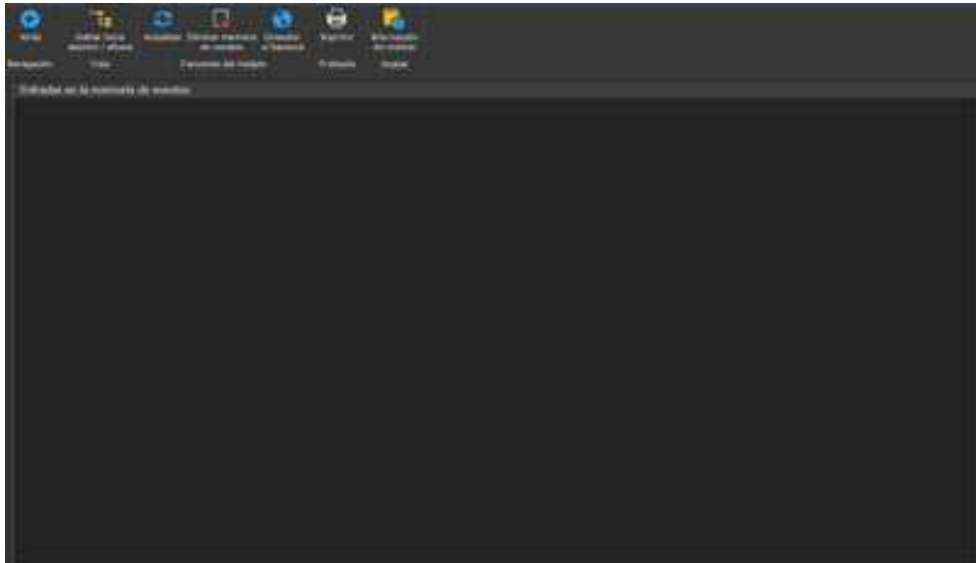


Figura 4.11: Memoria de eventos después del borrado.

4.2.4. Protocolo de diagnóstico

Las plataformas de diagnóstico como anteriormente lo comentamos no solo nos permite leer y borrar la memoria de averías de las unidades de control si no también funciones especiales, pero muchas de estas se encuentran montadas en sistemas informáticos de cada marca lo cuales están restringidos para uso general, ya que muchas funciones hoy en día se venden por tiempos determinados. Otros casos por el cual están restringidas es debido a los inmovilizadores de seguridad que usan los automóviles, estos son casados bajo sistemas informáticos y así poder realizar las adaptaciones de llaves para la apertura y encendido de los autos.

Para tener documentada la evidencia de la memoria de incidentes se realiza un guardado del protocolo, de la siguiente manera. Seleccionamos la opción imprimir protocolo:



Figura 4.12: Softkey Imprimir.

Y esto nos genera un archivo en formato .xml con toda la información de las unidades de control:

Gatewayverbauliste:

0001 - Electrónica del motor	i.O.
0002 - Electrónica del cambio	Fehler
0003 - Electrónica de los frenos	Fehler
0008 - Electrónica del aire acondicionado/calefacción	Fehler
0009 - Centralita eléctrica electrónica	Fehler
0013 - ACC	Fehler
0015 - Airbag	i.O.
0017 - Cuadro de Instrumentos	Fehler
0019 - Interfaz de diagnosis para bus de datos	Fehler
002B - Bloqueo columna direc.	i.O.
0036 - Regulación del asiento del lado del conductor	i.O.
003C - Asistente de cambio de carril	Fehler
0042 - Electrónica puerta conductor	Fehler
0044 - Dirección asistida	Fehler
004B - Módulo multifunción	Fehler
0052 - Electrónica puerta acompañante	Fehler
005F - Electrónica de información 1	Fehler
006D - Electrónica tapa trasera	i.O.
0075 - Módulo de llamada de emergencia y unidad de comunicación	Fehler
0076 - Ayuda de aparcamiento	Fehler
00A5 - Sensores delanteros para sistemas de asistencia al conductor	i.O.
00B7 - Interfaz para sist. acceso y arranque	Fehler
00BB - Electrónica de la puerta trasera del lado del conductor	nicht verbaut / codiert
00BC - Electrónica de la puerta trasera del lado del acompañante	nicht verbaut / codiert
00CA - Unidad de control del techo corredizo	i.O.
00D6 - Mando de la luz izquierda 2	i.O.
00D7 - Mando de la luz derecha 2	i.O.
BF09 - Conexión adicional de bus 9	i.O.
BF09 - Conexión adicional de bus 9	i.O.

Figura 4.13: Protocolo de diagnóstico.

Capítulo 5

Análisis de plan de estudios de la carrera Ingeniería en Sistemas Automotrices (ISA)

En este capítulo se presenta el análisis del plan de estudios de la carrera Ingeniería en sistemas automotrices

5.1. Movilidad

En base a lo estudiado en el plan de estudios de la carrera Ingeniería en Sistemas Automotrices se observa que esta misma maneja 3 fases importantes en el futuro de la movilidad los cuales son:



Figura 5.1: Fases de ISA.

Si bien es cierto la electrificación en los automóviles hoy en día es el punto más importante en el ámbito automotriz, empresas como Tesla llegaron a con un nuevo concepto de tren motriz el cual se ha vuelto la prioridad número uno para muchas de las industrias automotrices a nivel mundial, pero no todo está girando en base a la electrificación de los automóviles ya que empresas orientales como Toyota, Nissan entre otras están apostando por combustibles alternos como lo son el hidrógeno, donde por medio de celdas de combustible o reservorios de hidrógeno llegamos a obtener energía para la movilidad de los automóviles. Mi recomendación al respecto es que no solo se enfoque al auto eléctrico, si no también exista la posibilidad que el plan de estudios este enfocado a cualquier tipo de movilidad, pude observar lo cual me pareció muy interesante que la opción terminal uno habla sobre esto, pero creo podría ser un punto importante que no se manejara como opción si no como un debe.

5.2. Herramientas computacionales

Las herramientas de software utilizadas hoy en día dentro dentro de la industria automotriz son de desarrollo propio por el fabricante o realizada en trabajo conjunto con proveedores desarrolladores, pero en ambos casos se tratan de desarrollos soportados por firmas grandes como lo son:

- National Instrument
- dSpace
- Micronova
- Göpel
- IAV

Al tener una diversidad de proveedores y herramientas de diferentes fabricantes debe existir un trabajo conjunto para embeber todas y poder realizar los diferentes fines tanto de automatización como de simulación para las diferentes pruebas. La recomendación es una plataforma usada por varias firmas automotrices alemanas la cual es una gestora de programas y sirve para las automatizaciones de pruebas. Su función es llamar a los diferentes programas por medio de rutinas de pruebas y ejecutarlos en tiempo informa durante un ciclo de prueba. Esta herramienta se llama **EXAM** y es de la firma alemana **micronova MicroNova - Software und Systeme** ó en su defecto realizar una materia en la cual se pueda realizar alguna herramienta similar la cual sirva de instrumento didáctico para comprender la importancia de trabajar con diferentes proveedores tecnológicos.

5.3. Inmovilizador

Los sistemas antirrobo en los automóviles deben ser modificados continuamente debido a las nuevas tecnologías y vulnerabilidades que van existiendo con el paso del tiempo es por eso que las regulaciones del mercado de igual manera van exigiendo a la industria nuevas variantes o versiones las cuales deben ser liberadas por los diferentes países para que los autos puedan ser comercializados.

5.4. Conectividad y encriptación

La conectividad hoy en día esta de la mano de la movilidad en el transporte, funciones como:

- Apertura de automóvil
- Activación/desactivación de alarmas
- Encendido/apagado de luces
- Manejo autónomo
- Posicionamiento GPS

Se han incluido en el Pull de funciones remotas realizables por medio de dispositivos móviles para la comodidad del cliente y han llevado a la industria automotriz a ser regulados por leyes para la protección de datos en los vehículos. A partir de 2024 Europa reglamenta la encriptación de unidades de control dentro de los automóviles para proteger a los usuarios de posibles ataques cibernéticos.

Mi propuesta de punto 5.2 y 5.3 es la posible introducción de una materia que hable de esto y el desarrollo de algún sistema de inmovilización o telemático encriptado que ayude a los estudiantes a entender este concepto de una mejor manera, basándose en eventos del pasado los cuales pueden ser de gran utilidad para el desarrollo de un proyecto final.

5.5. Buses de comunicación

Actualmente el CAN-Bus es el protocolo más usado en la industria automotriz pero también existen otros tipos de buses de comunicación que se han introducido a las arquitecturas electrónicas unos predominan y otros han sido sacados del mercado por diversos factores.

- **CAN-bus.** Predominante en sus diferentes versiones.
- **LIN-Bus.** Predominante en sus diferentes versiones.
- **Flexray.** Descontinuado por algunas marcas por altos costos
- **MOST.** Descontinuado por algunas marcas por altos costos
- **Ethernet automotriz.** Empieza a predominar en las redes automotrices debido a su escalabilidad y flexibilidad y posiblemente en algunos años sea el protocolo más usado en la industria automotriz.

5.6. Asignatura de Interfaces y Redes Vehiculares

En la unidad de aprendizaje 3 (Tipos de redes automotrices) se habla de 4 protocolos de comunicación:

- 3.1 LIN
- 3.2 CAN
- 3.3 FlexRay
- 3.4 MOST
- 3.5 RF

Como lo comenté en el punto anterior el protocolo ethernet automotriz tiende a ser el protocolo más usado de la industria automotriz desplazando al CAN-Bus debido a su escalabilidad y flexibilidad que ofrece.

Mi aportación a este punto es que se incluya un punto más al punto 3 quedando de la siguiente manera:

- 3.1 LIN
- 3.2 CAN
- 3.3 FlexRay
- 3.4 MOST
- 3.5 RF
- Ethernet automotriz.

5.7. Propuesta de práctica de análisis red automotriz

A continuación se presenta la propuesta de una práctica diseñada para la carrera ISA.

Nombre de la asignatura	Laboratorio de Sistemas Automotrices
Clave de la asignatura	ISSA-202
Plan de estudio	Semestral
Licenciatura	LISA
Período	Otoño 2023

Objetivos Educativos

Formación disciplinar técnico-científica: El egresado de la Ingeniería en Sistemas Automotrices posee un acervo profesional de habilidades y conocimientos técnico-científicos de Ingeniería que lo hacen competente, para desarrollar soluciones en un amplio campo de la industria automotriz.

Competencias Específicas

Resuelve problemas relacionados con ámbito de la industria automotriz mediante la aplicación de conocimientos de matemáticas, ciencias y tecnologías de la ingeniería con la finalidad de mejorar los procesos, productos y servicios. Investiga problemas relacionados con la ingeniería en sistemas automotrices usando conocimientos y métodos basados en la investigación para producir resultados concluyentes en el área. Lidera equipos de trabajo mediante la toma de decisiones para la generación de oportunidades y mejorar la competitividad en su disciplina. Conoce y aplica la normativa y estándares propios de su disciplina con la finalidad de realizar procesos y procedimientos que garanticen la seguridad de los usuarios.

Práctica No.	1
Nombre de la práctica	Determinación del número buses de comunicación CAN en auto real
Duración (horas)	8 Horas

1	Objetivo de la práctica
<p>Realizar un levantamiento de la red de comunicación CAN-Bus en el auto Volkswagen Jetta ó Volkswagen New Beetle para determinar los buses de comunicación que conforman la arquitectura electrónica en estos autos.</p>	
2	Competencia de la práctica
<ul style="list-style-type: none"> ■ Identificar las unidades de control que trabajan con CAN-Bus. ■ Identificar en que bus de comunicación están montadas las unidades de control. ■ Identifica el número de buses de CAN en el automóvil. 	
3	Fundamento teórico
<p>Red CAN La red CAN, Controler Area Network o red de controladores de área fue desarrollada a mediados de los ochenta por el fabricante de componentes electrónicos Bosch, como una solución viable a la transmisión de gran cantidad de información en tiempo real entre unidades de control, sensores y actuadores en un tiempo muy reducido. La información circula por dos cables trenzados que unen todas las unidades de control que forman el sistema. Esta información se trasmite por diferencia de tensión entre los dos cables, de forma que un valor alto de tensión representa un 1 y un valor bajo de tensión representa un 0. La combinación adecuada de unos y ceros conforman el mensaje a transmitir. Los colores para alta y baja velocidad son de colores definidos por normas de cada marca, un ejemplo se muestra a continuación</p> <ul style="list-style-type: none"> ■ Low Speed - Naranja Verde ■ High Speed - Naranja Negro <p>Arquitectura electrónica</p> <p>El término arquitectura eléctrica/electrónica se refiere a la convergencia de hardware electrónico, comunicaciones de red, aplicaciones de software y cableado en un sistema integrado que controla un número cada vez mayor de funciones del vehículo en las áreas de control del vehículo, carrocería y seguridad, infotainment, seguridad activa y otras funcionalidades de comodidad, conveniencia y conectividad. La primera integración de sistemas eléctricos y mecánicos ocurrió a fines de la década de 1950 con la llegada del control de cruceo básico. En los años 60, hubo mejoras de audio e iluminación; en los años 70, los nuevos controles de emisiones impulsaron los avances en la arquitectura Eléctrico/Electrónicas y en la década de 1990, gestionar la complejidad de la arquitectura eléctrica/-electrónica se estaba convirtiendo en un problema para las grandes organizaciones automotrices.</p>	

A principios de la década de 2000, los protocolos de datos y comunicación impulsaron los requisitos de nuevos productos y, en la última década, las organizaciones automotrices se han centrado en las características y regulaciones que rigen la seguridad de los ocupantes, la distracción del conductor y el ahorro de combustible, lo que ha llevado a la adopción de sistemas y sistemas de propulsión de alto voltaje.

Mirando hacia el futuro, las arquitecturas Eléctrico/Electrónicas deberán adaptarse a los avances en conducción automatizada, información y entretenimiento ampliada, conectividad 5G y, lo que es más importante, mayor electrificación de vehículos. El enfoque tradicional de agregar gradualmente nuevas unidades de control electrónico (ECU) con su propia potencia, procesamiento, datos y conectividad para cada nueva función ya no funciona; no escalará para adaptarse a todos los nuevos requisitos de potencia informática, procesamiento de datos y distribución de energía. Al mismo tiempo, la industria automotriz intenta automatizar el ensamblaje de los arneses de cableado que sirven como columna vertebral de la arquitectura Eléctrico/Electrónicas y ahora se han convertido en uno de los componentes más complejos y difíciles de manejar del vehículo. El auge de los vehículos totalmente eléctricos es el punto de inflexión que brinda a la industria automotriz la oportunidad de comenzar de nuevo y crear una nueva arquitectura eléctrica/electrónica desde cero, una arquitectura que considera las necesidades de energía y datos de cada dispositivo eléctrico en el vehículo y cumple esas necesidades de la manera más ágil e integrada posible. Los beneficios y oportunidades potenciales son impresionantes. La firma que lo haga bien tendrán una ventaja competitiva al ofrecer una mejor experiencia de usuario digital.

4	Material necesario
	<ul style="list-style-type: none"> ■ Desarmadores punta plana. ■ Pica cables. ■ Pinzas para desmontaje de tapas plásticas y vestiduras plásticas.

5	Equipo necesario
	<ul style="list-style-type: none"> ■ Multímetro. ■ Linterna. ■ Computadora con Software de ruteo de circuitos para realizar el levantamiento.

6	Desarrollo experimental
	<ol style="list-style-type: none"> 1. Como actividad de investigación previa a la realización de la práctica, se debe investigar las características del par trenzado de CAN-Bus y los números de parte de las unidades de control que sospechamos que cuentan con comunicación CAN. 2. Desconectamos la fuente de alimentación del auto o batería montada en el automóvil. 3. Ubicar los conectores de las unidades que cuenten con CANBus y desconectarlos de las unidades de control. 4. Seleccionamos un conector de cualquiera de las unidades de control desconectadas. 5. Identificar los cables correspondientes a CAN High y CAN Low. 6. Colocar nuestro multímetro en modo continuidad. 7. Conectar una punta del multímetro directamente en la terminal del CAN High ó en su defecto con un pica cables, picar el cable de CAN High. 8. Colocamos la otra punta de nuestro multímetro en el CAN High de otra unidad de control. <i>Nota: Si existe continuidad en esos dos puntos, quiere decir que ambas unidades se encuentran en el mismo bus de comunicación.</i> 9. Conectar una punta del multímetro directamente en la terminal del CAN Low ó en su defecto con un pica cables, picar el cable de CAN Low. <i>Nota: Si existe continuidad en esos dos puntos, quiere decir que ambas unidades se encuentran en el mismo bus de comunicación.</i> <p>Es posible que exista una puerta de enlace o Gateway en el cual estén concentrados todos los buses de comunicación, esto nos facilitaría la obtención de nuestra arquitectura.</p>

7	Preguntas de la actividad
	<ol style="list-style-type: none"> 1. ¿Existe una puerta de enlace en estos autos? 2. ¿Cuántos buses de comunicación CAN encontraron? 3. ¿Cuántas unidades de control CAN están montadas en cada bus de comunicación? 4. ¿Cuáles son los códigos de colores que encontraron en el par trenzado de CAN Bus en los diferentes buses de comunicación? 5. ¿Existe algún sub bus de CAN Bus que esté definido para algún esclavo de alguna unidad de CAN Bus?

8	Evaluación
<p>Reporte impreso debe incluir: una portada, contenido, introducción, objetivo, desarrollo, resultados, las preguntas de la actividad y conclusiones.</p>	
9	Seguridad en la realización de la práctica
<ul style="list-style-type: none">■ Esta práctica se hace sin tener ningún tipo de suministro de energía en el auto (batería o fuente de alimentación)■ Precaución al desmontar las diferentes piezas del vehículo	
10	Referencias bibliográficas para la realización de la práctica
<ul style="list-style-type: none">■ Tesis manual de pruebas de can-bus para arquitecturas electrónicas de automóviles.■ Control Area Network, Version 1, pag.107., Dipl.Ing. Herbert Nabereit CAN AUFBAU, SYS-TECH+AUTRON.■ ISO-11898-1:2003-Road vehicles-Controller area network (CAN)-Part 1: Data link layer and physical signalling	

Conclusiones

- De todo lo anterior se deduce que la aplicación hoy en día del protocolo de comunicación CAN-Bus es el más usado y normalizado en la industria automotriz.
- De acuerdo con los objetivos planteados, de este trabajo concluimos que este manual es de ayuda para los estudiantes en la licenciatura en ciencias de la electrónica y carreras afines, para así poder entender de una manera más sencilla el uso y características del CAN-Bus en los automóviles y medios de transporte que hoy en día utilizan este protocolo de comunicación.
- Los objetivos planteados en esta tesina se cumplen satisfactoriamente ya que los resultados obtenidos son los requeridos para nuestras pruebas y para el aprender el protocolo de CAN Bus en un automóvil.
- El diseñar un manual de CAN Bus no es tan fácil ya que existen muchos detalles, como por ejemplos los documentos de confidencialidad que no se pueden revelar.
- Se estudiaron los aspectos técnicos del CAN Bus, tanto etapa física, protocolo de comunicación y diagnóstico con las diferentes herramientas que se usan en la industria automotriz, tomando como base ejemplos reales de pruebas.

Apéndice A

Cálculo de la resistencia de cierre

A continuación se describe como se calcula teóricamente la resistencia de cierre del CAN-bus de alta velocidad y corroborar este valor realizando una medición entre CANH y CANL (500 K Baudios). Matemáticamente tenemos que:

$$R_{\text{CANH-CANL}} = \left(\frac{1}{\frac{1}{R_{\text{ECU}_1}} + \frac{1}{R_{\text{ECU}_2}} + \frac{1}{R_{\text{ECU}_3}} + \dots + \frac{1}{R_{\text{ECU}_n}}} \right) \quad (\text{A.1})$$

donde R_{ECU} es la resistencia interna de cierre de cada unidad de control dentro del bus, la cual se obtiene el valor teórico dentro de la especificación de desarrollo de cada unidad de control. Haciendo una medición del Bus de alta velocidad tenemos que:

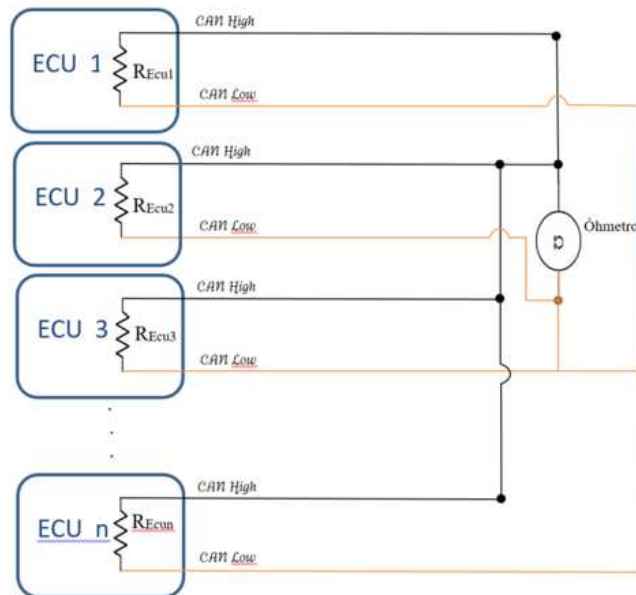


Figura A.1: Circuito de resistencia de cierre en unidades de control.

La tabla que se tiene que llenar es la siguiente:

Resistencia	Debe ser	Es	Correcto/Incorrecto
$R_{CANH-CANL}$	Valor teórico	Valor medido	

Tabla A.1: Medición de Resistencia total del Bus.

Apéndice B

Mediciones

Al realizar la medición del ancho de pulso y los valores de voltaje del recesivo y dominante del Bus de alta velocidad (500 K Baudios) se deben llenar las siguientes tablas.

	Debe ser	Es	Correcto/Incorrecto
Ancho de pulso	1.998 ... 2.002		

Tabla B.1: Ancho de pulso CAN High Speed.

Nivel de señal	UCAN _H		Correcto/Incorrecto
	DEBE [V]	ES[V]	
Nivel de señal recesiva	2.40, ... 2.6		
Nivel de señal dominante	2.75, ... 4.5		

Tabla B.2: Nivel de señal UCAN_H para CAN High Speed

Nivel de señal	UCAN _L		Correcto/Incorrecto
	DEBE [V]	ES[V]	
Nivel de señal recesiva	2.40, ... 2.60		
Nivel de señal dominante	0.50, ... 2.25		

Tabla B.3: Nivel de señal UCAN_L para CAN High Speed

Apéndice C

Pruebas de comunicación

Para realizar la prueba de comunicación se realizan los siguientes pasos:

1. Estabilidad del anillo de comunicación para cada uno de los buses de comunicación con una configuración estándar de CANoe.
 - Realizar un diagnóstico completo del automóvil
 - Ejecutar una configuración estándar de CANoe durante 24 horas.
 - Analizar la ventana de estadísticas en CANoe y llenar los datos de la tabla siguiente:

	Debe ser	Es	Correcto/Incorrecto
Errorframes	0		
Alive Messages	0		

Tabla C.1: Prueba de estabilidad del anillo de comunicación.

- Realizar un diagnóstico completo del automóvil y analizar que no existen errores en las unidades de control después de la prueba (comparando el protocolo de inicio de la prueba con el protocolo final de la prueba).
2. Realizar prueba de tranquilidad del Bus de comunicación para cada uno de los buses de comunicación, si se tiene una la capacidad de analizar más de un Bus al mismo tiempo la prueba nos ahorra tiempo.
 - Realizar un diagnóstico completo del automóvil
 - Apagar el auto, cerrar todas las puertas y ventanas, esperar 10 minutos.
 - Ejecutar una configuración estándar de CANoe durante 24 horas (como mínimo).

- Al finalizar el tiempo analizar la ventana de estadísticas en CANoe y llenar los datos de la tabla siguiente:

	Constante	No constante	Correcto/Incorrecto
Tranquilidad del Bus			

Tabla C.2: Tranquilidad de Bus.

- Realizar un diagnóstico completo del automóvil y analizar que no existen errores en las unidades de control después de la prueba (comparando el protocolo de inicio de la prueba con el protocolo final de la prueba).
3. Realizar prueba de formación del anillo de comunicación para todos los buses de comunicación utilizando la configuración llamada análisis de anillo y sacar un diagnóstico antes y después de la prueba.
- Realizar diagnóstico completo del auto
 - Instrumentar el auto con CANExtender para simulación del switch de encendido
 - Ejecutar la configuración RingAnalyse.cfg con la cual activaremos de manera cíclica el switch de encendido del auto durante un mínimo de 24 horas.
 - Al finalizar realizamos un diagnóstico completo del auto y comparamos con el diagnóstico inicial.
 - Llenamos la siguiente tabla con la información obtenida del diagnóstico completo después de la prueba.

Los resultados de la prueba se muestran en la siguiente tabla:

	Debe se	Es	Correcto/Incorrecto
Fallas antes de la prueba	ninguna		
Fallas después de la prueba	ninguna		
Respondieron las unidades de control	Todas/Siempre		

Tabla C.3: Formación del anillo de comunicación.

Apéndice D

Instructivo Vector-CANoe

Son herramientas de software, para el desarrollo y análisis de la comunicación de las unidades de control que se encuentran conectadas a las diferentes redes del auto. **CANoe** sirve para visualizar los mensajes que envían las unidades de control, para poder realizar un análisis de la comunicación en las distintas redes. Los mensajes se pueden guardar en un archivo denominado **Trace**.

D.1. Interfaces (Hardware)

Se utilizan diversos tipos de hardware para realizar mediciones de FlexRay, CAN, Most y LIN:

D.1.1. Piggies

Son tarjetas electrónicas intercambiables con el Transceiver de CAN, LIN o FlexRay, utilizadas para la adquisición de datos.

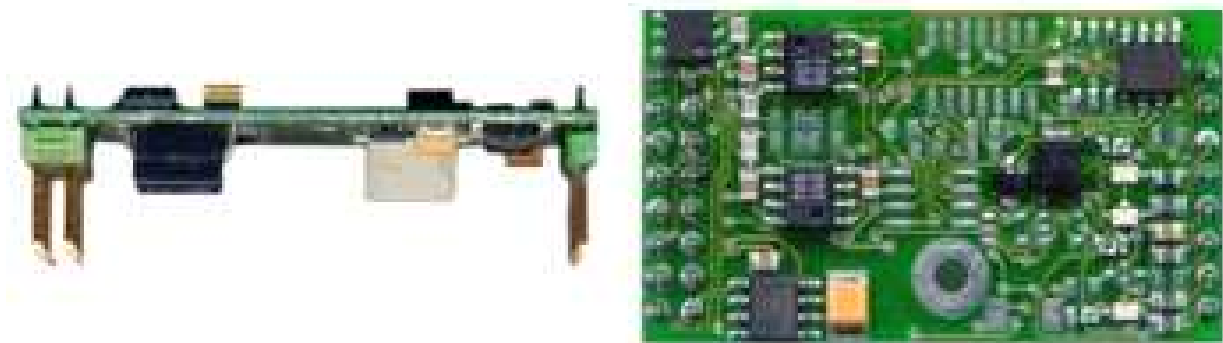


Figura D.1: Piggies.

D.1.2. VN8900

Se utiliza para FlexRay, CAN y LIN. La lectura se realiza por medio de tarjetas electrónicas llamadas **Piggy** y se debe de utilizar dependiendo del tipo de red que se va a medir. La configuración de Piggies son:

- CH1: FlexRay
- CH2: CAN1
- CH3: CAN2
- CH4: LIN
- CH5: No se conecta
- CH6: CAN3
- CH7: CAN4
- CH8: No se conecta

Para los canales de CAN se utilizan **Cables 2Y** de Vector:

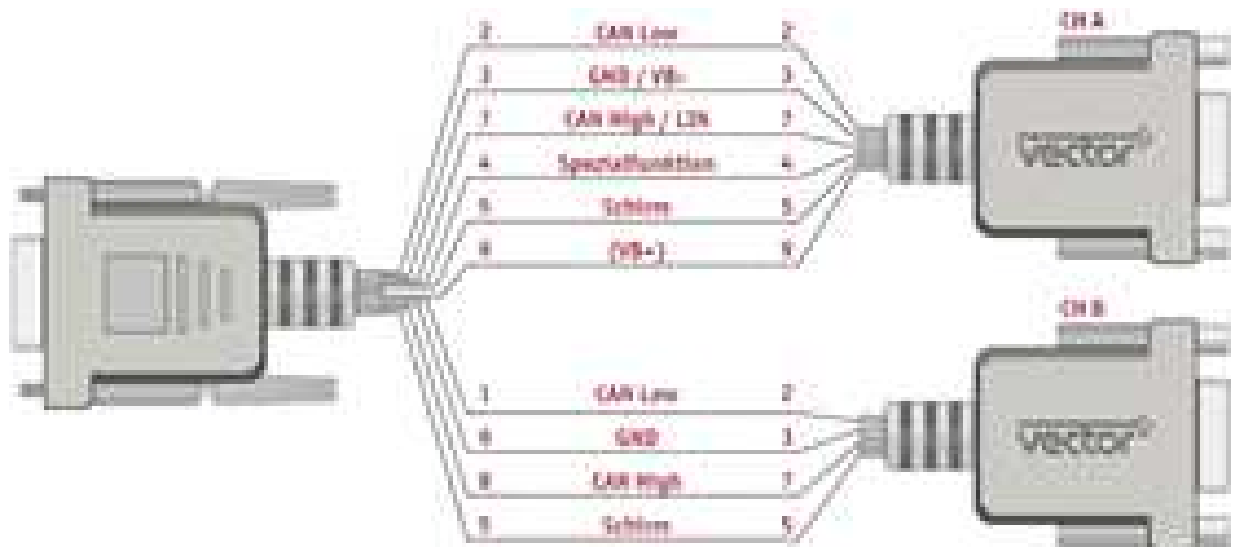


Figura D.2: Piggies.

La interface se compone de dos módulos VN8910A que es el case y VN8970 el cual corresponde a la tarjeta electrónica para leer FlexRay, CAN y LIN.

D.1.3. VN1600

Se utiliza para realizar mediciones de CAN y LIN. También se tienen que insertar **Piggies** dependiendo del tipo de red que se va medir y se utiliza el cable 2Y de Vector para realizar la conexión. Actualmente cuenta con:

- CH1: LIN
- CH2: CAN1
- CH3: No utilizado
- CH4: CAN2



Figura D.3: VN1600.

D.1.4. VN2640

Es utilizado para las mediciones de MOST



Figura D.4: VN2640.

D.2. Uso del CANoe

La pantalla principal de **CANoe** puede ser configurada por el usuario, de acuerdo a la información que se desea visualizar. En la imagen que se muestra a continuación se puede observar que está dividida en distintas ventanas, las cuales corresponden a los mensajes de cada una de las redes, configuración de canales y la configuración de la medición y visualización.

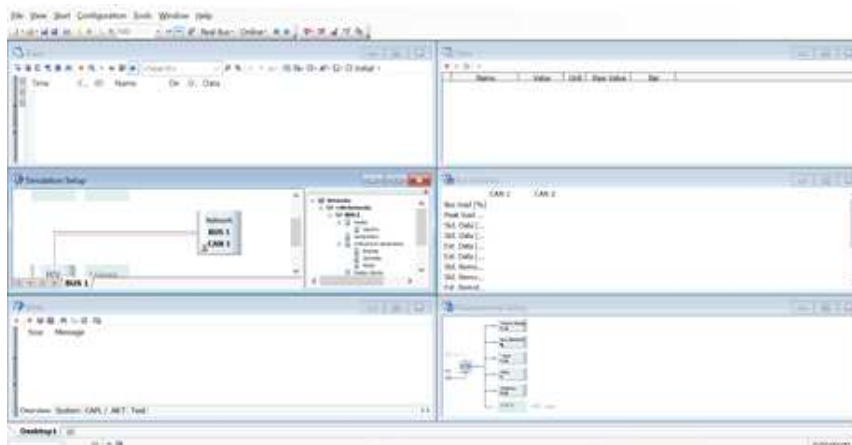


Figura D.5: pantalla principal de **CANoe**.

D.2.1. Configuración del Hardware

En el menú de configuración se selecciona **Options** y en la opción de **General** se escoge la cantidad de canales de CAN, LIN, MOST o FlexRay que se desea tener.

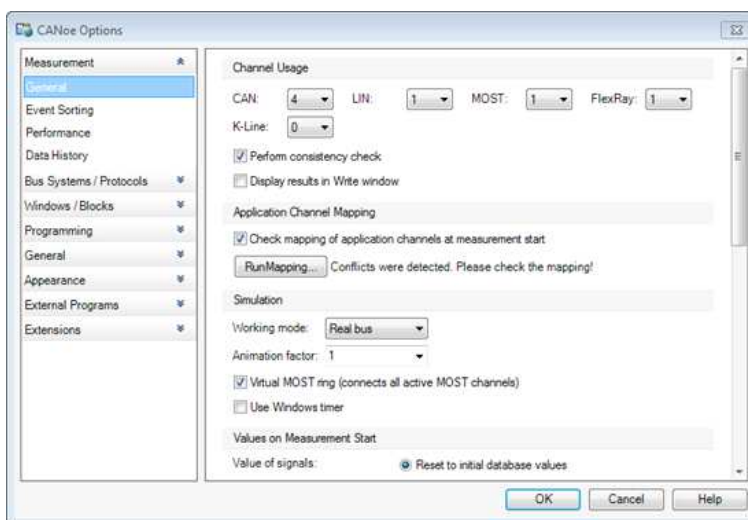


Figura D.6: Configuración del Hardware.

En el menú de **Configuration** se selecciona la opción de **Network Hardware** y se despliega la siguiente ventana:

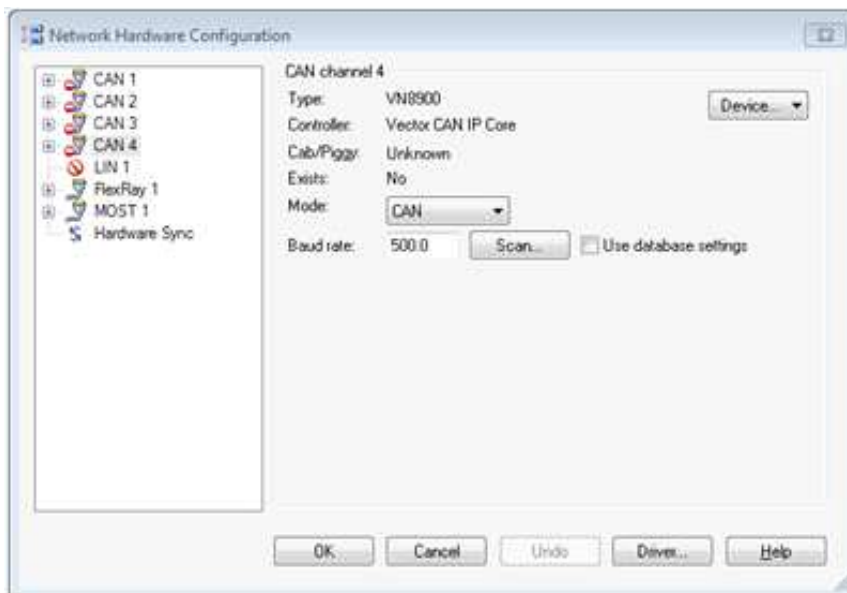


Figura D.7: menú de **Configuration**.

En esta ventana se configura la velocidad de la transmisión de datos (Baud Rate) en los canales de CAN, la cual debe de ser de 500 Mbit/s. En el caso del LIN la velocidad de 19200. Para el MOST se selecciona la opción de MOST 150. Para realizar la asignación de los canales en la interfaz, se presiona en la opción de **Driver** y se muestra la siguiente ventana.

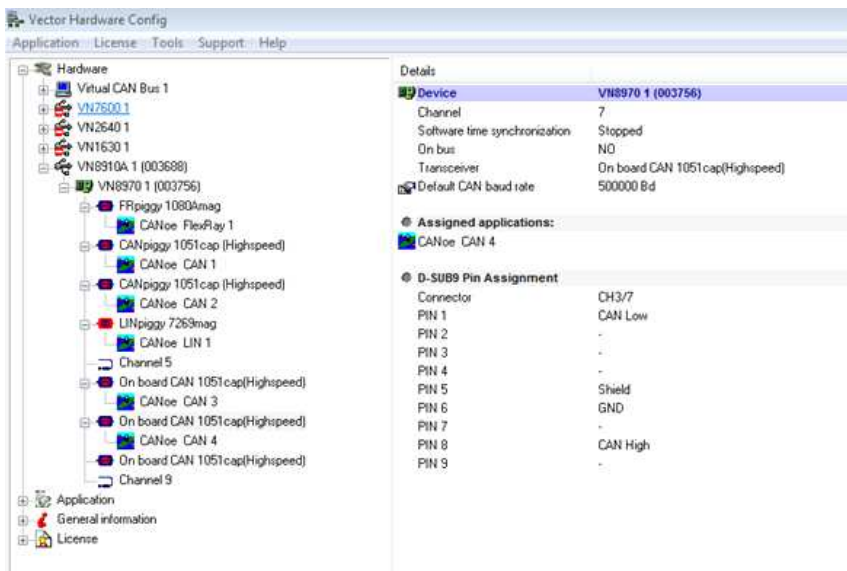


Figura D.8: Velocidad de la transmisión.

En el listado se muestran los canales y el tipo de **Piggy** que se encuentra conectado a dicho canal. Con el botón derecho del Mouse se presiona sobre el canal y se muestran las opciones para asignar la red deseada. Aunque no se tenga instalado el CANalyzer o el CANape, es posible que se muestren estas opciones, las cuales no tienen influencia en la medición.

D.2.2. Configuración de la Simulación

En el panel de la derecha se muestran las distintas redes (CAN, LIN, FlexRay o MOST). Con el botón derecho del Mouse sobre el tipo de red se muestra la opción de agregar una red, la cual se le puede cambiar el nombre para que sea más fácil su identificación.

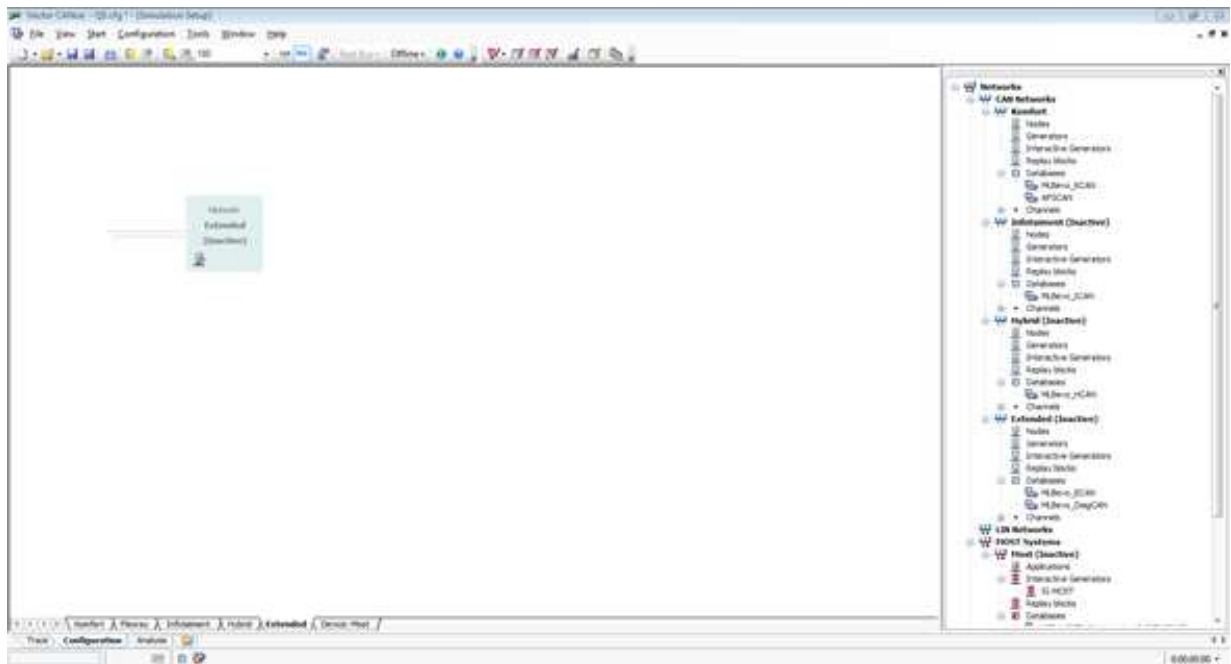


Figura D.9: Panel.

En la Opción de **Databases** se agrega la base de datos que se le quiera asignar a esa red. La base de datos se utiliza para traducir los datos en binario que se leen de la red, en mensajes con valores y rangos pre-establecidos. En el caso de redes CAN la base de datos es un archivo `.dbc`. Para el FlexRay y Most se utilizan archivos `.xml` y en el caso del LIN es archivo LDF. Las bases de datos son confidenciales y cada firma automotriz desarrolla sus propias bases de datos. En la opción de **Channel** se asigna el canal que se va a utilizar para cada una de las redes. Cuando no se utiliza un canal o tipo de red es recomendable desactivarlo, con el fin de que no se generen errores en la ejecución de la medición. Para desactivar un canal se presiona con el botón derecho sobre el nombre del canal a desactivar y se presiona en la opción de **Network active**. Cuando los canales se encuentran desactivados se muestra la indicación junto al nombre del canal. Para desactivar un tipo de red es necesario presionar con el botón derecho sobre el tipo de red y seleccionar la opción de **Deactivate All**. Este procedimiento es necesario cuando no se encuentra conectado el Hardware específico para tipo de red, por ejemplo **MOST**, **LIN** o algún otro.

D.2.3. Configuración de una medición de CAN-Bus

En la ventana anterior de **CANoe** se configuran los filtros para visualizar cada una de las redes en una ventana diferente. También se configura el bloque de Logging para guardar los **Traces**. Para agregar una ventana de visualización de CAN o cualquier otro protocolo de comunicación se presiona el botón derecho del Mouse sobre la línea vertical del corchete y se selecciona la opción de **Insert Trace Windows** y después la opción deseada.

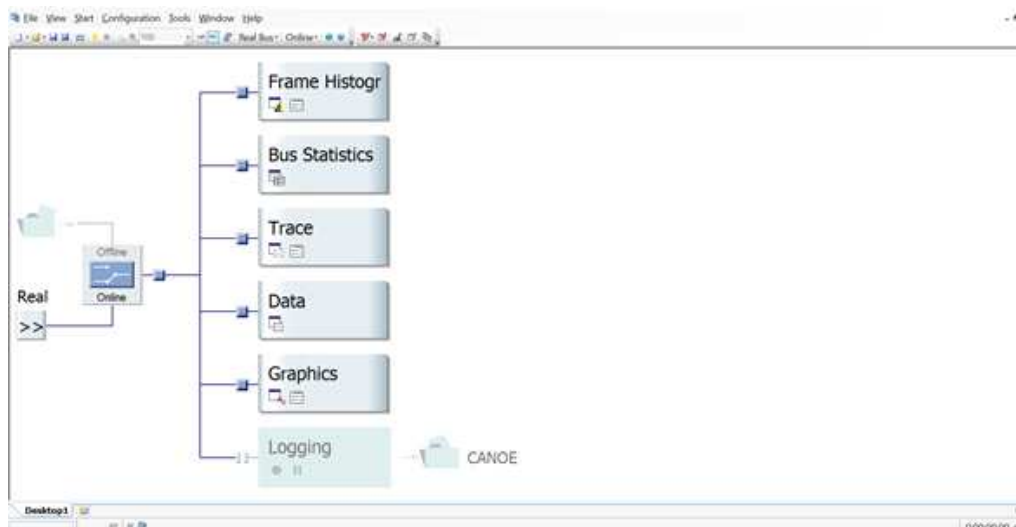


Figura D.10: ventana de **CANoe**.

El filtro se agrega dando click con el botón derecho sobre el cuadro que se encuentra entre la línea vertical y la ventana de Trace. Después se escoge la opción de **Channel Filter**. Al dar doble click sobre el filtro se abre la ventana de configuración del filtro, en el que se selecciona que canales se van a bloquear. El nombre del bloque se cambia con el botón derecho sobre el bloque y seleccionado la opción de **Rename**. En el bloque que se encuentra en lado izquierdo de la ventana, se utiliza para configurar si es una medición **On Line** que es cuando el equipo está conectado al auto o la opción de **Off Line** la cual se utiliza para analizar la información de un archivo que se había obtenido previamente. En el caso de la función **Off Line**, se selecciona el archivo fuente dando click con el botón derecho y seleccionando la opción de **Configuration**.

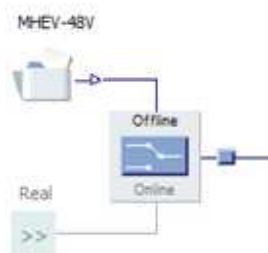


Figura D.11: Rename.

D.2.5. Finalizar Medición

La medición se finaliza presionando el ícono que se muestra en la siguiente imagen:

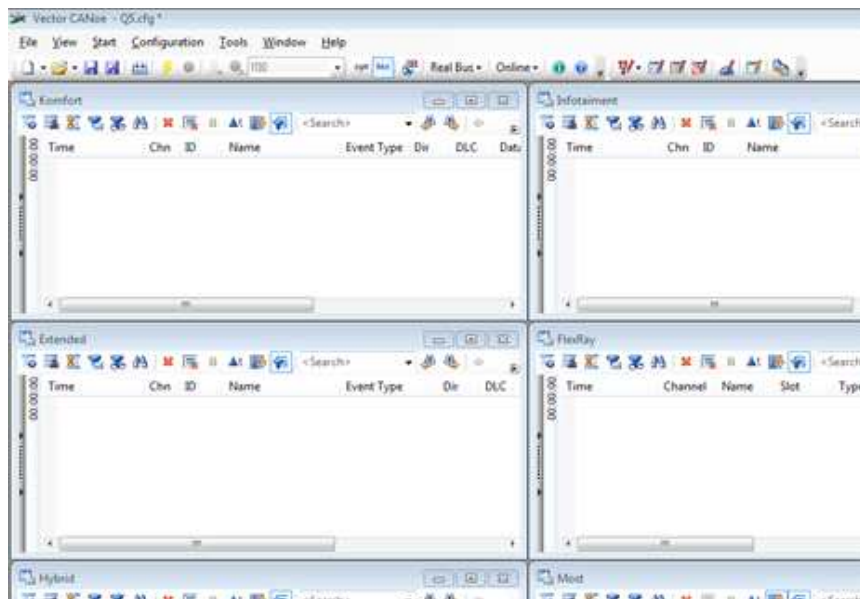


Figura D.14: Ventana de Gráficas.

En esta ventana se puede visualizar el comportamiento de las señales deseadas. Para agregar señales a esta ventana se puede realizar de tres diferentes maneras:

- En el área de la gráfica o en la lista de señales se presiona con el botón derecho y se selecciona la opción de **Add Signals** o **Signale hinzufügen** y se despliega la siguiente ventana:



Figura D.15: opción de **Add Signals**.

En la cual se encuentran organizadas las señales dependiendo las bases de datos que se hayan agregado a la configuración. Se busca la señal en el mensaje deseado y se le da doble clic.

- También es posible copiar la señal de la ventana de Trace. En la ventana de Trace se busca la señal que se quiera graficar y se presiona con el botón derecho, se selecciona la opción de **Copy to Clipboard** y después en la ventana de gráfica se presiona con el botón derecho en la zona de listado de señales y se selecciona la opción de **Paste**.
- La tercera opción es arrastrar la señal desde la ventana de Trace. En la ventana de Trace se busca la señal deseada y se arrastra a la ventana de Trace. Para realizar esta función es necesario tener las ventanas en miniatura.

Para tener una mejor visualización de las señales, es recomendable que se elija la opción de vistas separadas. Se presiona con el botón derecho y se selecciona la opción de **Configuration** en el menú de Y-Axis se selecciona la opción de **Separate Views**.

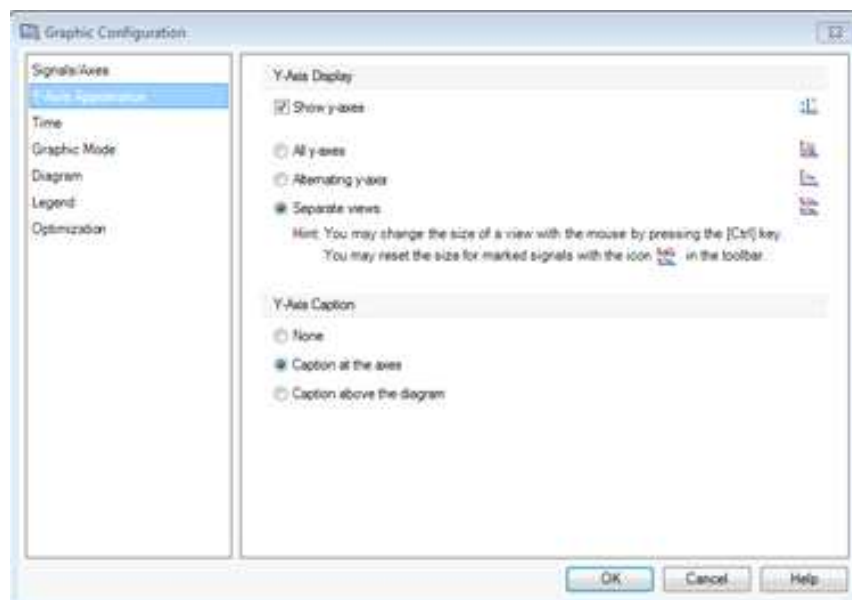


Figura D.16: opción de **Separate Views**.

Las opciones de visualización como: Zoom, Fit, cursores, configuración de los Ejes, etc.; se encuentran en la barra de herramientas.

D.2.6. Traces

Para realizar un Trace se utiliza el bloque de **Logging** en la ventana de **Measurement Setup**.

Para acceder a la ventana de configuración, se da doble clic sobre el bloque de **Logging**. En esta ventana se puede seleccionar el modo del Trace.

- Entire Measurement: Es el modo más usual para almacenar un Trace completo. Los datos se almacenan al iniciar la medición y se termina cuando se presiona el Stop.
- Single Trigger y Toggle Trigger: Se utilizan para iniciar la grabación de los datos al ocurrir un suceso determinado por el usuario, puede ser mensajes la misma red. Al presionar el botón de inicio de la medición no se guarda información en el archivo hasta que las condiciones establecidas se alcancen.

También es posible configurar el tiempo de Pre-Trigger y un Post-Trigger. En el caso del Pre-Trigger se establece el tiempo que va a almacenar de información antes de que ocurra el Trigger. En el momento en el que se da el Trigger la información almacenada durante el tiempo establecido se guarda en el archivo y después se almacena el tiempo establecido en el Post-Trigger.

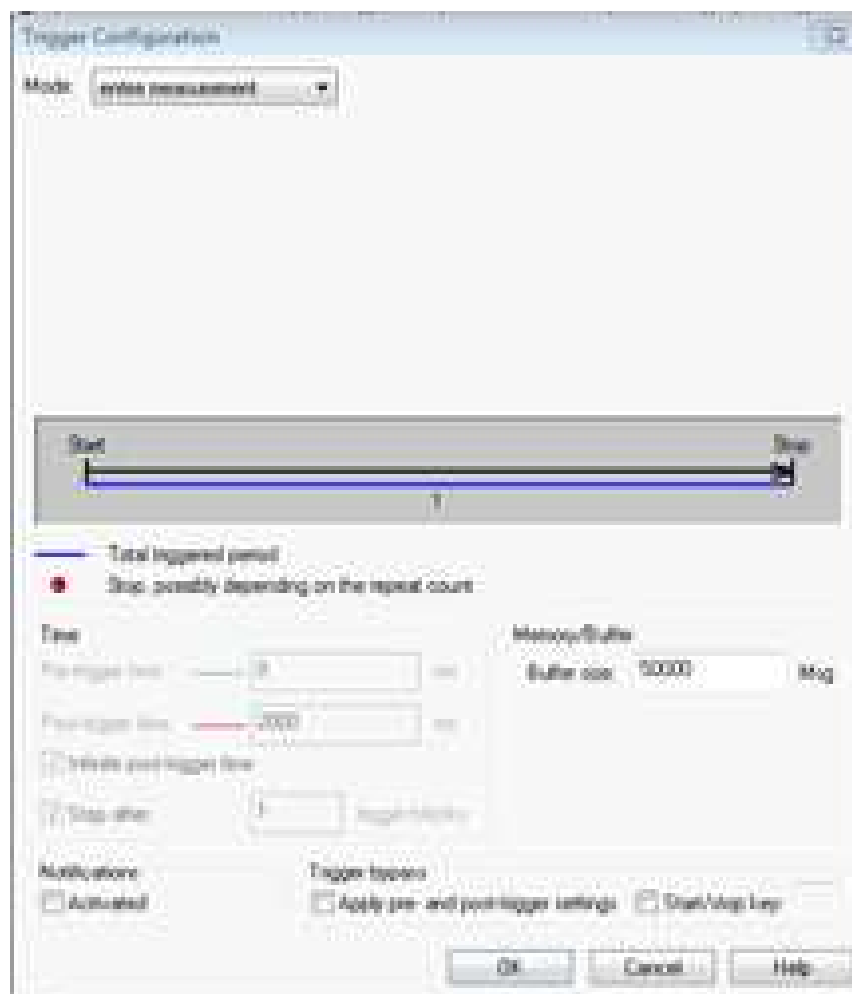


Figura D.17: configurar el tiempo de Pre-Trigger.

El archivo se configura dando click con el botón derecho del Mouse sobre el símbolo de la carpeta y en la ventana que se abre se puede configurar el nombre y la ubicación del archivo.

Los archivos ASCII (.asc) son más sencillos de manejar y resultan compatibles con otros programas capaces de visualizar datos de comunicación de redes, pero son archivos de gran capacidad. Los archivos BLF (.blf) tienen un gran grado de compresión, por lo cual resulta más sencillo el envío de información, pero únicamente se pueden visualizar mediante programas de **CANoe**/CANalyzer.

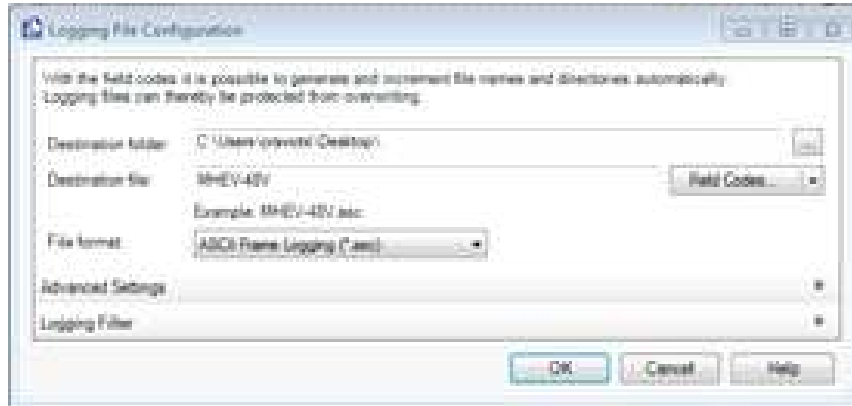


Figura D.18: programas de **CANoe**/CANalyzer.

Apéndice E

Reconocimientos

Miembro de la sexta generación del programa Trainee de Volkswagen de México con el proyecto automatización del banco de pruebas Confort-HiL por medio de Robot Mitsubishi RV-1A.

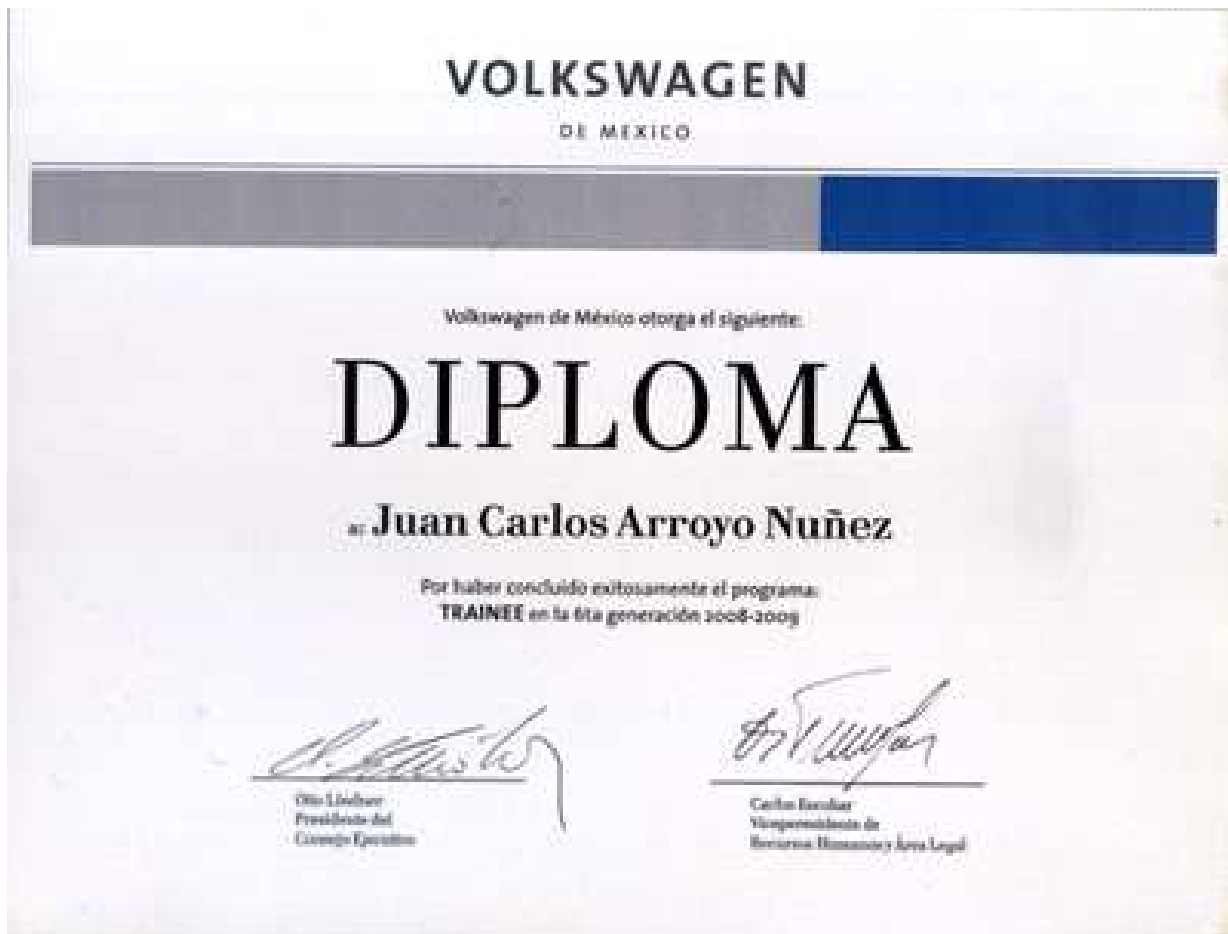


Figura E.1: Reconocimiento

Bibliografía

- [1] Vector training, Version 1, *CANoe SYSTEM DEVELOPMENT*, Vector CANtech inc., pag. 205. suite 550, 39500, Orchard Hill Place, Novi, Michigan.
- [2] Contro Area Network, Version 1, pag.107., *Dipl. Ing. Herbert Nabereit CAN AUFBAU, SYSTECH+AUTRON*.
- [3] XL interface, Version 2.0, 130 Pag Vector Informatik GmbH, Ingershiemer Strasse 24 D-70499 Stuttgart.
- [4] Documentación de experiencia laboral en la industria automotriz
- [5] Especificaciones de CAN Bus High Speed
- [6] <http://canbus.galeon.com/electronica/canbus.ht.m>
- [7] <http://www.can-cia.org/index.php?id=systemdesign-can-physicallayer>
- [8] <http://dSPACE.ups.edu.ec/bitstream/123456789/1121/2/Capitulo>
- [9] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=33422
- [10] <http://www.can-cia.org/index.php?id=systemdesign-can-physicallayer>
- [11] http://www.bosch-semiconductors.de/media/pdf_1/canliteratur/can2spec.pdf
- [12] <http://es.scribd.com/doc/106167326/12/ISO-9141-2>
- [13] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=33423
- [14] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=36055
- [15] http://www.interfacebus.com/Automotive_SAE_J1850_Bus.html