THE LIN CONCEPT

FEATURES

LIN is a single-wire serial communications protocol based on the common SCI (UART) byte-word interface. UART interfaces are available as low cost silicon module on almost all micro-controller and can also be implemented as equivalent in software or pure state machine for ASICs. The medium access in a LIN network is controlled by a master node so that no arbitration or collision management in the slave nodes is required, thus giving a guarantee of the worst-case latency times for signal transmission.

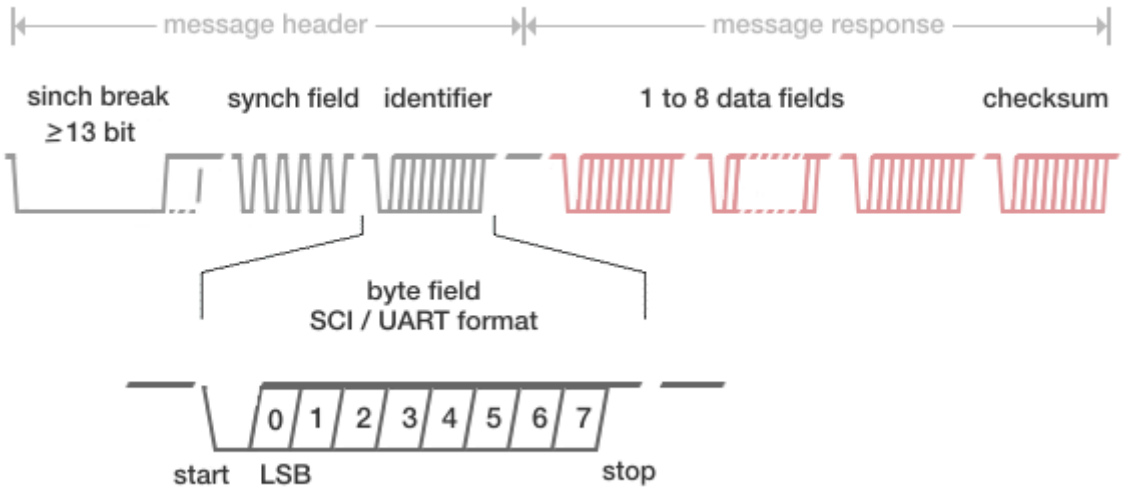
A particular feature of LIN is the synchronization mechanism that allows the clock recovery by slave nodes without quartz or ceramics resonator. The specification of the line driver and receiver is following the ISO 9141 single-wire standard with some enhancements. The maximum transmission speed is 20 kbit/s, resulting from the requirements by electromagnetic compatibility (EMC) and clock synchronization.

A node in LIN networks does not make use of any information about the system configuration, except for the denomination of the master node. Nodes can be added to the LIN network without requiring hardware or software changes in other slave nodes. The size of a LIN network is typically under 12 nodes (though not restricted to this), resulting from the small number of 64 identifiers and the relatively low transmission speed. The clock synchronization, the simplicity of UART communication, and the single-wire medium are the major factors for the cost efficiency of LIN.

COMMUNICATION CONCEPT

A LIN network comprises one master node and one or more slave nodes. All nodes include a slave communication task that is split in a transmit and a receive task, while the master node includes an additional master transmit task. The communication in an active LIN network is always initiated by the master task as illustrated in the figure below: the master sends out a message header which comprises the synchronization break, the synchronization byte, and the message identifier.

Exactly one slave task is activated upon reception and filtering of the identifier and starts the transmission of the message response. The response comprises one to eight data bytes and one checksum byte. The header and the response part form one message frame.



The identifier of a message denotes the content of a message but not the destination. This communication concept enables the exchange of data in various ways: from the master node (using its slave task) to one or more slave nodes, and from one slave node to the master node and/or other slave nodes. It is possible to communicate signals directly from slave to slave without the need for routing through the master node, or broadcasting messages from the master to all nodes in a network. The sequence of message frames is controlled by the master and may form cycles including branches.

TECHNICAL OVERVIEW

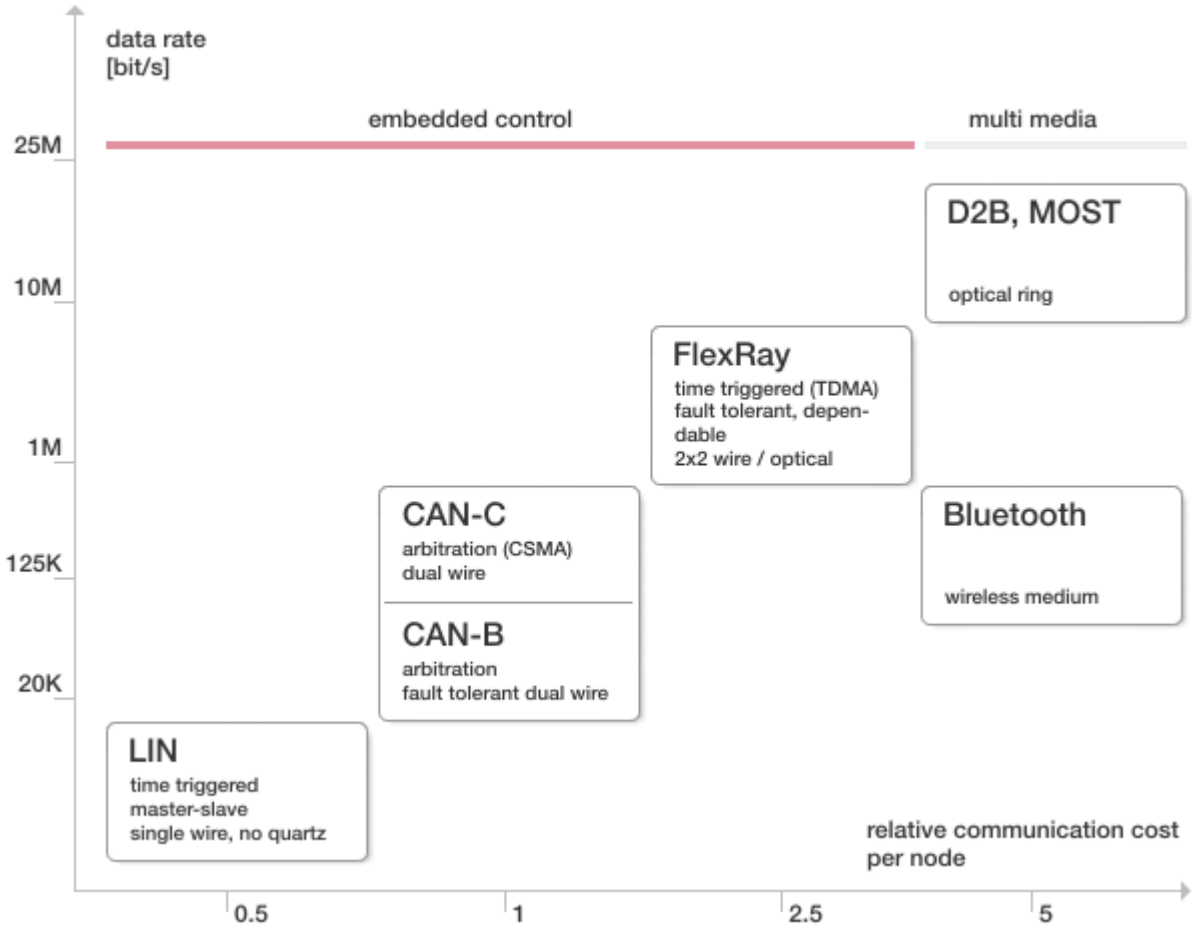
LIN (Local Interconnect Network) is a low cost serial communication system intended to be used for distributed electronic systems in vehicles, which complements the existing portfolio of automotive multiplex networks (see figure below).

LIN is a holistic communication concept for local interconnect networks in vehicles. The specification covers in addition to the definition of the protocol and the physical layer also the definition of interfaces for development tools and application software.

LIN enables a cost-effective communication for smart sensors and actuators where the bandwidth and versatility of CAN is not required. The communication is based on the SCI (UART) data format, a single-master/multiple-slave concept, a single-wire 12V bus, and a clock synchronization for nodes without stabilized time base.

The LIN consortium has been developed to standardize a concept of a serial low cost communication concept in conjunction with a development environment, that enables the car manufacturers and their suppliers to create, implement, and handle complex hierarchical multiplex systems in a very cost competitive way.

The LIN standard reduces the manifold of existing low-end SCI based multiplex solutions and cuts the cost of development, production, service, and logistics in vehicle electronics.



THE KEY FEATURES OF LIN ARE:

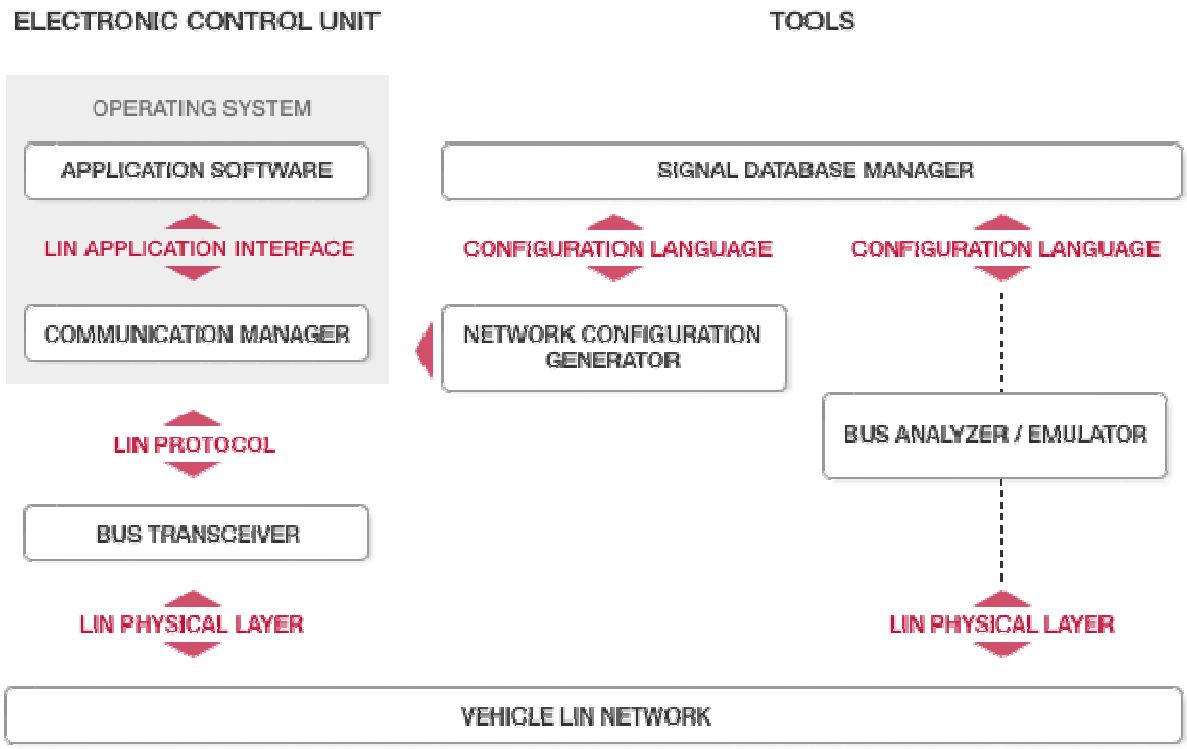
- Low cost single-wire implementation
- Enhanced ISO 9141, VBAT-based speed up to 20Kbit/s
- Acceptable speed for many applications (limited for EMI-reasons)
- Single Master / Multiple Slave concept
- No arbitration necessary
- Low cost silicon implementation based on common UART/SCI interface hardware
- Almost any microcontroller has necessary hardware on chip
- Self synchronization in the slave nodes without crystal or ceramics resonator
- Significant cost reduction of hardware platform
- Off-the-shelf slaves
- Flexibility because of configuraton features
- Guaranteed latency times for signal transmission
- Predictable systems possible

LIN SPECIFICATION

The LIN Standard encompasses the specification of the transmission protocol, the transmission medium, the interface between development tools, and the interfaces for software programming. LIN guarantees the interoperability of network nodes from the viewpoint of hardware and software, and a predictable EMC behavior.

The Specification package consists of the following documents:

- The **Protocol Specification** describes the data link layer of LIN.
- The **Transport Layer Specification** describes how to transport data that can be up to 4095 bytes. Normally the transport layer is used for node configuration, identification and diagnostics.
- The **Node configuration and Identification Specification** defines how to configure a slave node and how to identify a slave node.
- The **Diagnostic specification** describes types of diagnostic services a slave node will support. All diagnostic services are using the transport layer.
- The **Physical Layer Specification** describes the physical layer, including bit rate, bit rate tolerances, etc.
- The **Application Program Interface Specification** describes the interface between the network and the application program, including the node configuration, identification and transport layer interfaces.
- The **Configuration Language Specification** describes the format of the LIN description file, which is used to configure the complete network and serve as a common interface between the OEM and the suppliers of the different nodes, as well as an input to development and analysis tools.
- The **Node Capability Language Specification** describes a format used to describe properties of slave nodes. A node capability file may be used with a LIN cluster design tool to automatically create LIN description files.



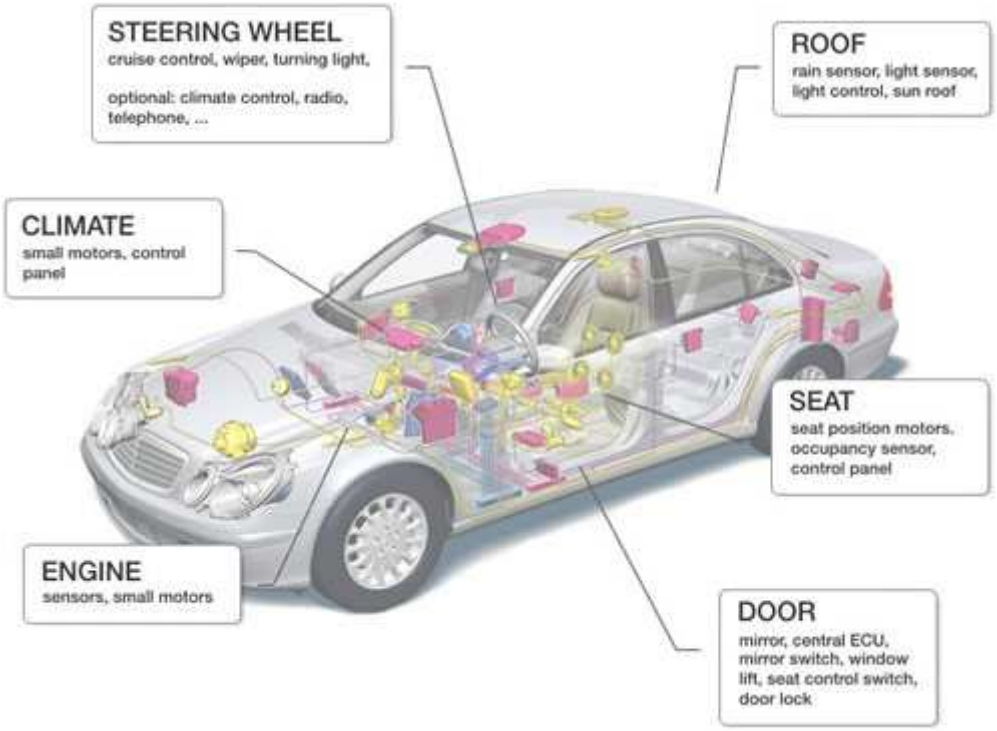
Sistema de multiplexado de datos "LIN-Bus"

Introducción
LIN es la abreviatura de **L**ocal **I**nterconnect **N**etwork.
El LIN-Bus es una extensión del bus de datos CAN. A un máximo de 20 Kbit/s, su velocidad de transferencia de datos es muy inferior a la del sistema de bus CAN. El bus LIN conecta actuadores o sensores con las correspondientes unidades de control. Las órdenes se transmiten siempre en una sola dirección, desde la denominada unidad de control maestra al sensor o actuador conectados en sentido descendente, el "esclavo". El maestro puede transmitir órdenes hasta a 16 esclavos conectados en sentido descendente. Un ejemplo aplicación de LIN bus es el techo solar de cristal eléctrico, cuyo servomotor recibe sus órdenes desde la unidad de control de confort a través del bus LIN.

LOCAL INTERCONNECT NETWORK

Local Interconnect significa en este caso, que todas las unidades de control están localizadas en una zona limitada (p. ej. en el techo). También se le da el nombre de «subsistema local».
El intercambio de datos entre los diferentes sistemas de LIN-Bus en un vehículo se realiza respectivamente por medio de una unidad de control a través del CAN-Bus de datos.
En el caso del LIN-Bus se trata de un bus monoalámbrico. El cable tiene el color básico violeta y un color de identificación. La sección del conductor es de 0,35 mm. No requiere apantallado.
El sistema permite el intercambio de datos entre una unidad de control LIN maestra y hasta 16 unidades de control LIN esclavas.

Elementos del vehiculo que se pueden controlar a traves del LIN-Bus.



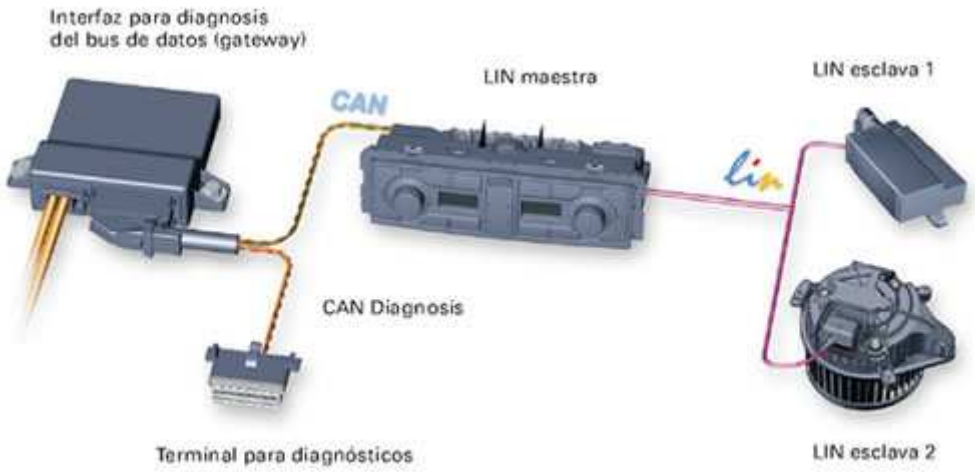
Ejemplo parcial de una instalacion de un bus LIN y su integracion en el bus CAN.

Unidad de control LIN "maestra"

La unidad de control que va conectada al CAN-Bus es la que ejecuta las funciones de maestra en el LIN-Bus.

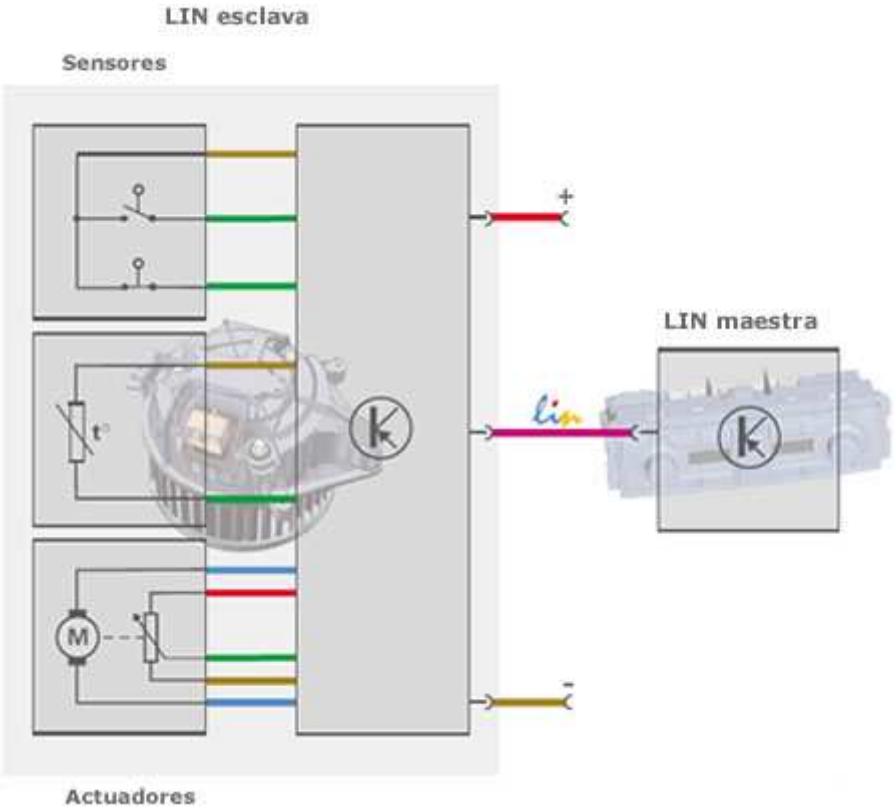
Funciones asignadas:

- Controla la transmisión de datos y su velocidad. La unidad de control LIN maestra transmite el encabezamiento del mensaje.
- En el software se define un ciclo, según el cual se han de transmitir mensajes al LIN-Bus y se especifica cuáles.
- Asume la función de traducción entre las unidades de control LIN abonadas al sistema del LIN-Bus local y el CAN-Bus de datos. De esa forma es la única unidad de control del LIN-Bus que va conectada a su vez al CAN-Bus.
- La diagnosis de las unidades de control LIN esclavas que lleva conectadas se realiza a través de la unidad de control LIN maestra.



Unidades de control LIN "esclavas"

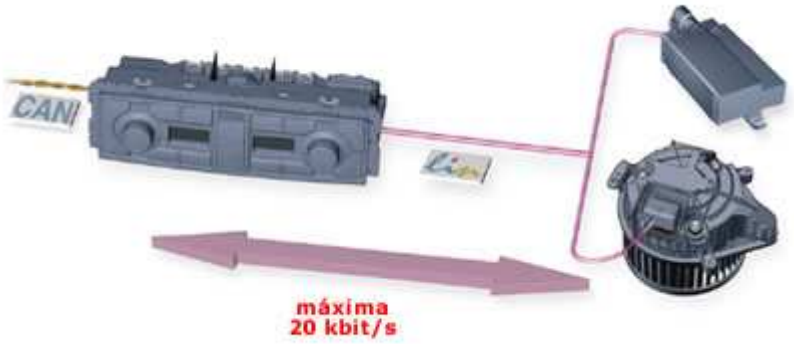
En un sistema de bus de datos LIN pueden funcionar como unidades de control LIN esclavas las unidades de control específicas, p. ej. la de la turbina de aire fresco o también pueden funcionar como tales los sensores y actuadores, p. ej. el sensor inclinométrico o bien el sonorizador DWA.
Los sensores llevan integrada una parte electrónica que analiza los valores medidos.
La transmisión de estos valores se realiza entonces a través del LIN-Bus en forma de señales digitalizadas.
Para varios sensores y actuadores se necesita un solo pin en la hembrilla de la LIN maestra.



Los actuadores en el LIN-Bus son grupos componentes electrónicos o electromecánicos inteligentes, a los que se les pasan sus instrucciones en forma de las señales de datos LIN procedentes de la unidad de control LIN maestra. A través de sensores integrados se puede consultar el estado operativo efectivo de los actuadores a través de la UCE LIN maestra, de modo que sea posible efectuar la comparación de los estados teórico y efectivo.

Transmisión de datos

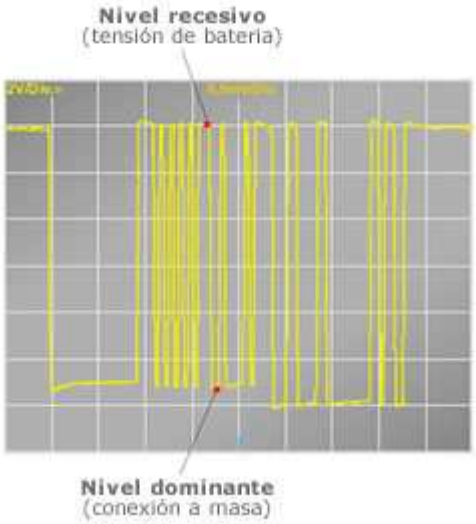
La velocidad de transmisión es de 1 - 20 kbit/s y viene determinada en el software de las unidades de control LIN. Equivale como máximo a una quinta parte de la velocidad de transmisión de los datos en el CAN Confort.



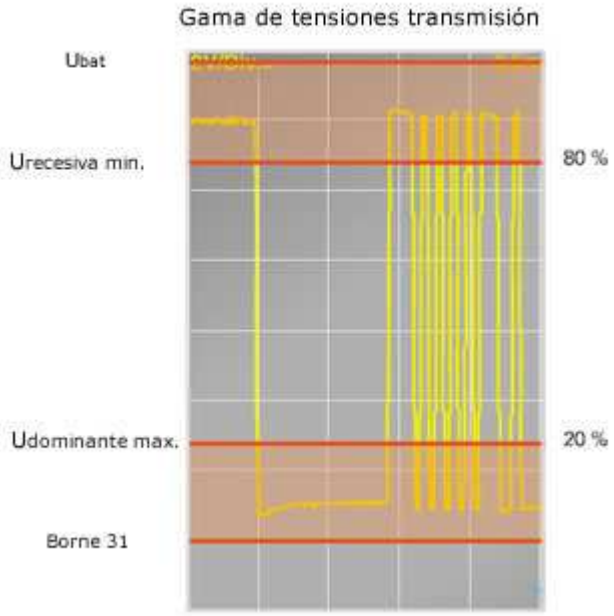
Señal

- Nivel recesivo
  Si a través del LIN-Bus no se transmite ningún mensaje o se transmite un bit recesivo, el cable del bus tiene aplicada una tensión equivalente prácticamente a la de batería.

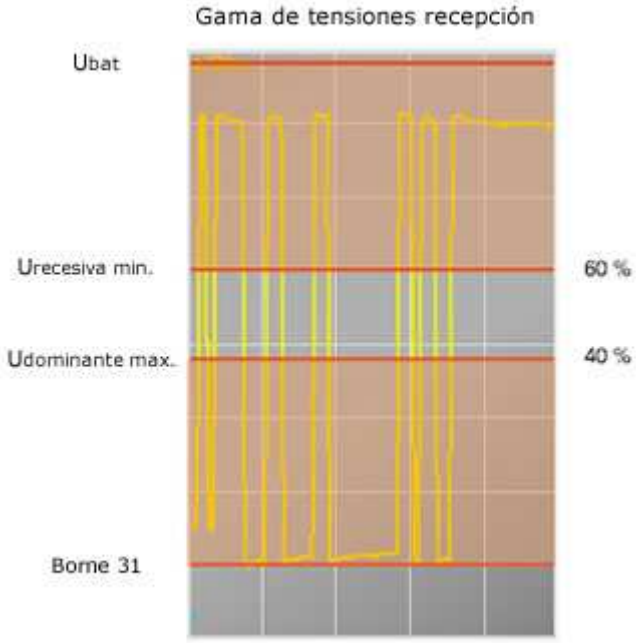
- Nivel dominante
  Para transmitir un bit dominante sobre el LIN-Bus, un transceptor en la unidad de control que efectúa la transmisión conecta el cable del bus de datos a masa.



Seguridad de transmisión
Con la determinación de las tolerancias para la transmisión y recepción en la gama de los niveles recesivo y dominante se tiene dada una transmisión estable.



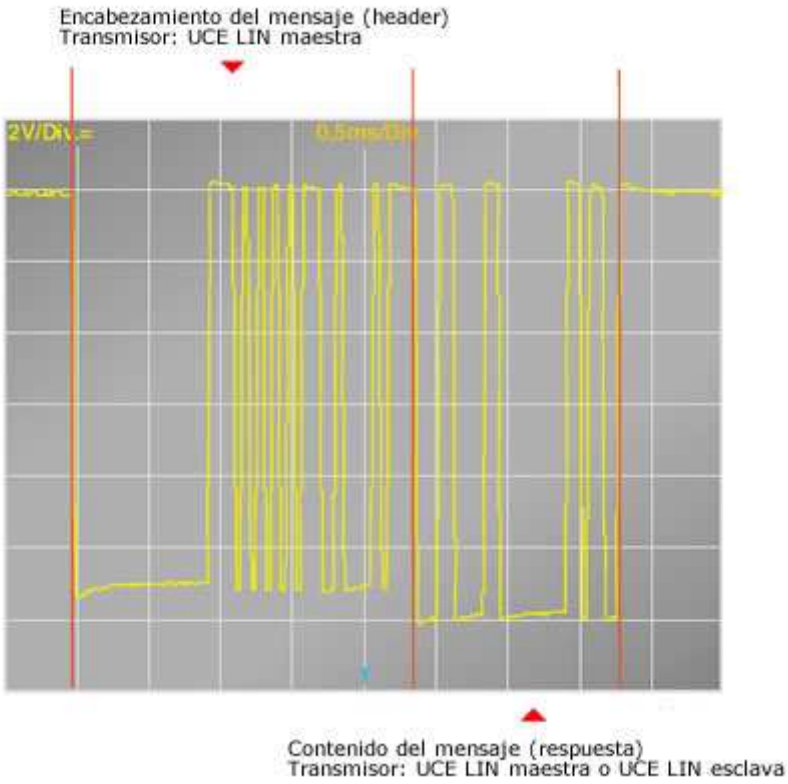
Para poder recibir señales válidas a pesar de existir interferencias parásitas se han configurado más extensas las gamas de tensiones admisibles por el lado de la recepción.



Mensajes

Mensaje con respuesta de esclava
La unidad de control LIN maestra exhorta en el encabezamiento a una unidad de control LIN esclava a que transmita información, p. ej. sobre condiciones de conmutadores o valores de medición.
La unidad de control LIN esclava transmite la respuesta correspondiente.

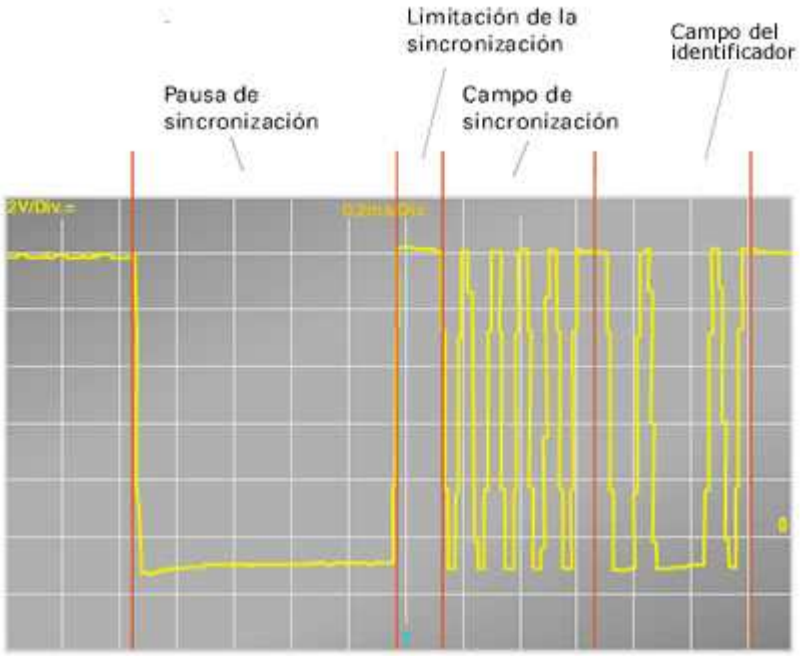
Mensaje con mandato de maestro
Por medio del identificador en el encabezamiento, la unidad de control LIN maestra exhorta a las correspondientes unidades de control LIN esclavas a que utilicen los datos contenidos en la respuesta.
La respuesta es transmitida en este caso por la unidad de control LIN maestra.



Encabezamiento del mensaje (header)
El encabezamiento es transmitido de forma cíclica por la unidad de control LIN maestra.

Se divide en cuatro campos:

- Pausa de sincronización
- Limitación de la sincronización
- Campo de sincronización
- Campo del identificador



- La pausa de sincronización («synch break») tiene una longitud mínima de 13 tiempos por bit. Se transmite con nivel dominante.
  Resulta necesaria la longitud de 13 bit para indicar de forma inequívoca el comienzo de un mensaje a todas las unidades de control LIN esclavas.
  En las demás partes del mensaje se transiten como máximo 9 bits dominantes consecutivos.

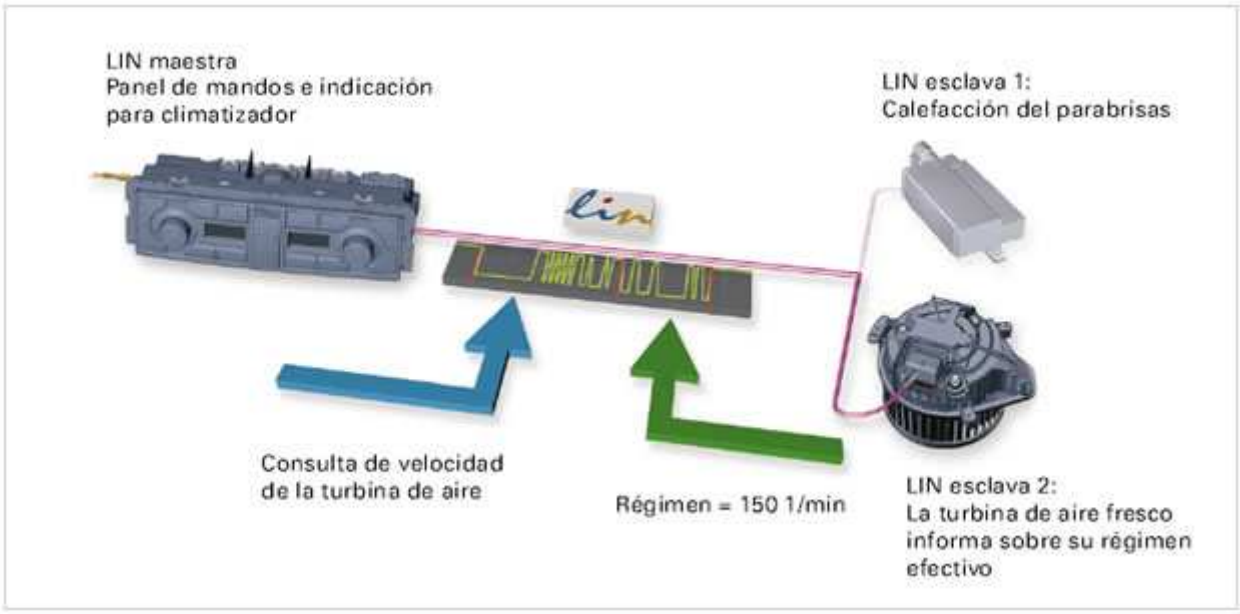
- La limitación de la sincronización («synch delimiter») tiene una longitud mínima de 1 bit y es recesiva (˜Ubat.).

- El campo de sincronización («synch field») está compuesto por la cadena binaria 0 1 0 1 0 1 0 1 0 1. Con esta secuencia de bits se pueden ajustar (sincronizar) todas las unidades
  de control LIN esclavas al ritmo del sistema de la unidad de control LIN maestra.
  La sincronización de todas las unidades de control resulta necesaria para disponer de un intercambio de datos exento de errores. Si se pierde la sincronización, los valores de los
  bits serían implantados en un sitio incorrecto del mensaje en el receptor, produciéndose errores en la transmisión de los datos.

- El campo del identificador tiene una longitud de 8 tiempos por bit. En los primeros 6 bits está contenida la identificación del mensaje y el número de campos de datos (ver página 14)
  que componen la respuesta.
  El número de campos de datos en la respuesta puede ser de entre 0 y 8.
  Los dos últimos bits reciben la suma de verificación de los 6 primeros bits, para la detección de errores de transmisión. La suma de verificación se necesita para evitar que se produzcan asignaciones a mensajes equivocados al haber errores de transmisión del identificador.

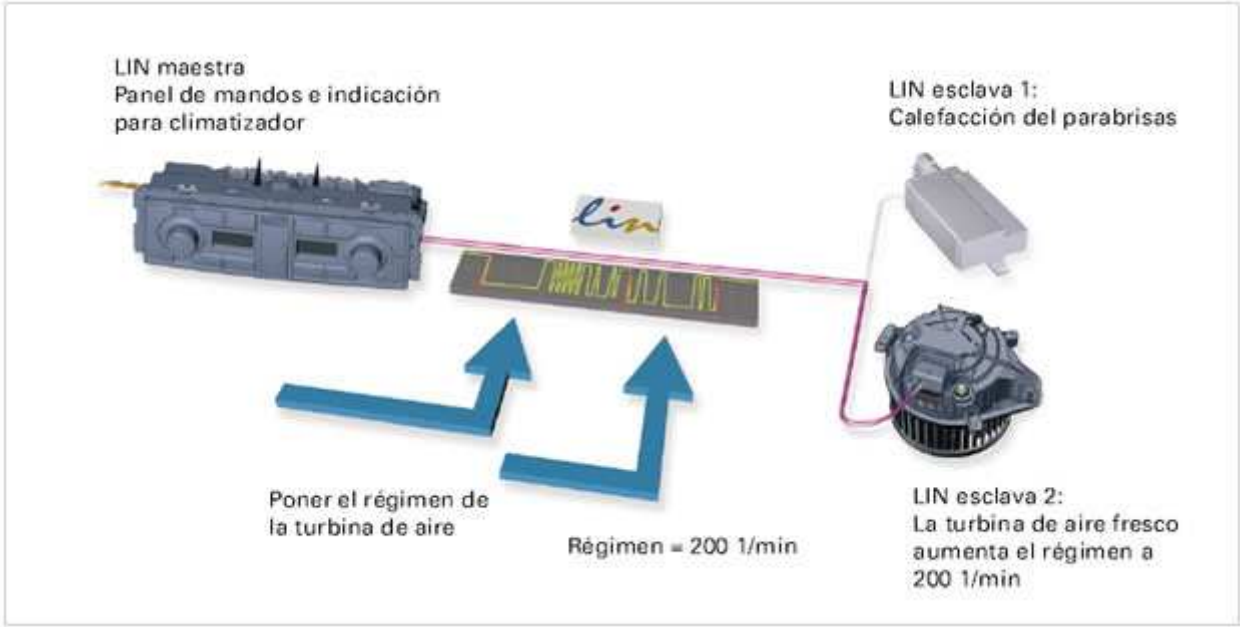
Contenido del mensaje (respuesta)
En el caso de un mensaje con respuesta de esclava, una unidad de control LIN esclava agrega información a la respuesta obedeciendo a lo especificado en el identificador.

Ejemplo en la figura inferior:



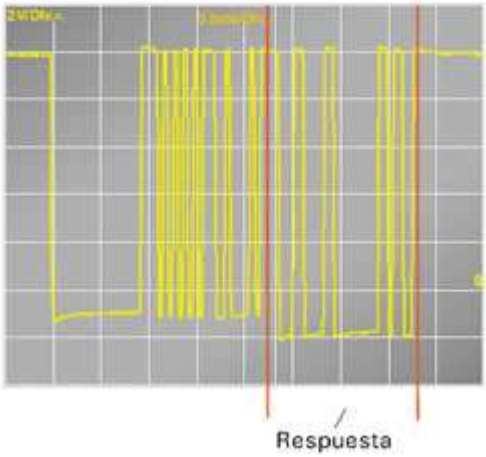
En un mensaje con solicitud de datos por parte de la UCE LIN maestra, ésta última es la que agrega la respuesta.
En función de lo especificado en el identificador, las unidades de control LIN esclavas procesan los datos y los utilizan para la ejecución de funciones.

Ejemplo en la figura inferior:



La respuesta consta de 1 a 8 campos de datos (data fields). Un campo de datos consta de 10 bits. Cada campo de datos está compuesto por un bit de arranque dominante, un byte de

datos que contiene la información y un bit de parada. Los bits de arranque y parada se utilizan para la resincronización y, por tanto, para evitar errores de transmisión.

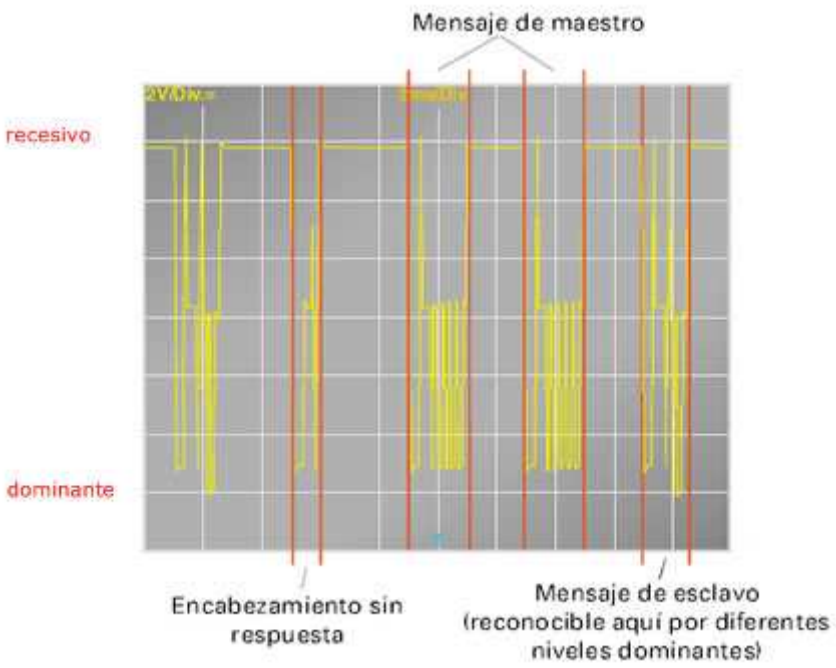


Orden de los mensajes
Siguiendo el orden especificado en su software, la unidad de control LIN maestra transmite cíclicamente sobre el LIN-Bus los encabezamientos y, al tratarse de mensajes maestros, incluye las respuestas.
La información que se necesita con mayor frecuencia se transmite también más frecuentemente.
El orden de los mensajes puede variar en función de las condiciones dadas en el entorno de la unidad de control LIN maestra.

Ejemplos de condiciones del entorno:

- Encendido ON/OFF
- Diagnosis activa/inactiva
- Luz de población ON/OFF

Para reducir la cantidad de versiones de la unidad de control LIN maestra, ésta transmite sobre el LIN-Bus los encabezamientos destinados a las unidades de control de un vehículo con equipamiento completo.
Por la ausencia de unidades de control para equipamientos opcionales aparecen en la imagen del osciloscopio encabezamientos sin respuestas. Esto no influye sobre el funcionamiento del sistema.



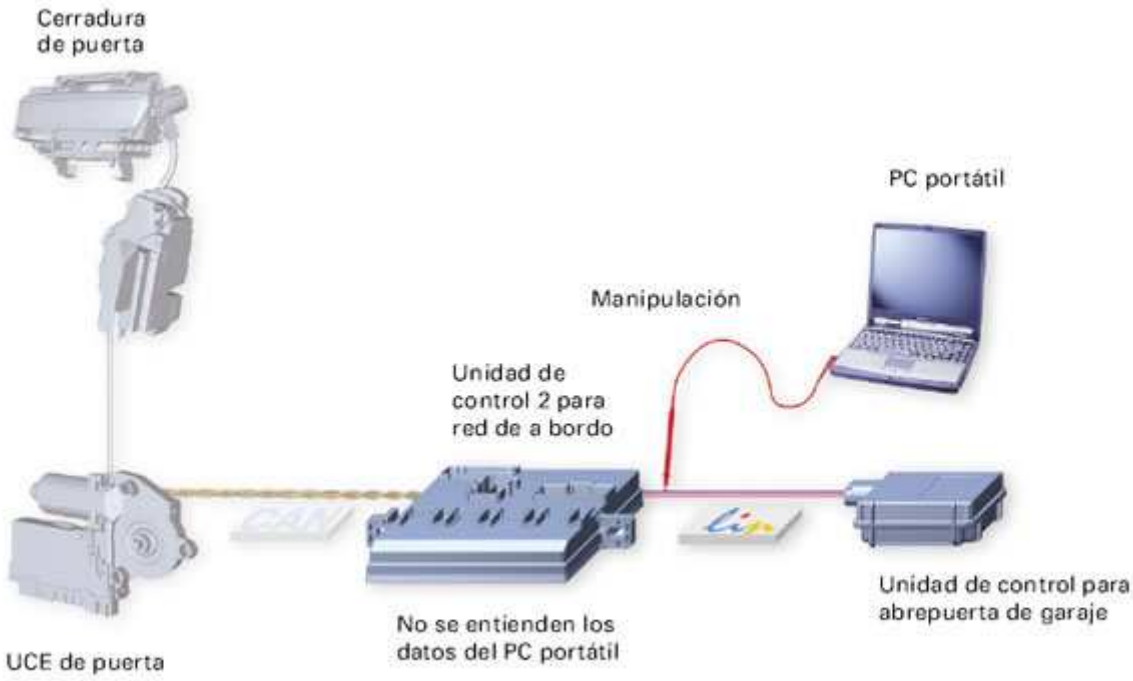
Proteccion antirrobo

La transmisión de datos en el sistema de bus LIN únicamente se lleva a cabo si la unidad de control LIN maestra transmite un encabezamiento con un identificador correspondiente.

Las posibles manipulaciones en un cable LIN situado fuera de la lámina exterior del vehículo se imposibilitan a base de que la unidad de control LIN maestra efectúe una verificación completa de todos los mensajes.

La unidad de control LIN esclava sólo puede responder. De esta forma, por ejemplo, no es posible desbloquear las puertas a través del LIN-Bus.

Estos nexos permiten incorporar unidades de control LIN esclavas en la zona exterior del vehículo (p. ej. la unidad de control para el abrepuerta del garaje, situada en el paragolpes delantero).



Diagnosis

La diagnosis de los sistemas de LIN-Bus se realizan a través del código de dirección correspondiente a la unidad de control LIN maestra.
La transmisión de los datos de diagnosis por parte de las unidades de control LIN esclavas hacia la UCE LIN maestra se efectúa a través del LIN-Bus.

Punto de la avería	Texto de la averia	Causa de la inscripción de avería
Unidad de control LIN esclava, p. ej. regulador de turbina de aire	sin señal/ sin comunicación	Interrupción de la transmisión de datos de la unidad de control LIN esclava sobre un período de tiempo definido en el software de la UCE LIN maestra. • Interrupción de cable o cortocircuito • Alimentación de tensión defectuosa para la unidad de control LIN esclava • Versión incorrecta LIN esclava o LIN maestra • Unidad de control LIN esclava averiada

Unidad de control LIN esclava, p. ej. regulador de la turbina de aire	Señal no plausible	Error en la suma de verificación. Transmisión incompleta de los mensajes. • Interferencias electromagnéticas en el cable LIN • Alteraciones de capacidad • resistencia en el cable LIN (p. ej. por humedad/suciedad en la carcasa de conexión) • Problema de software (versiones incorrectas de las piezas)

Ejemplo practico de la aplicación del LIN-Bus

En el vehículo de la marca Seat modelo Altea se amplia de forma considerable el número de líneas de CAN-Bus y se introducen nuevas lineas de LIN-Bus (ver esquema figura inferior).

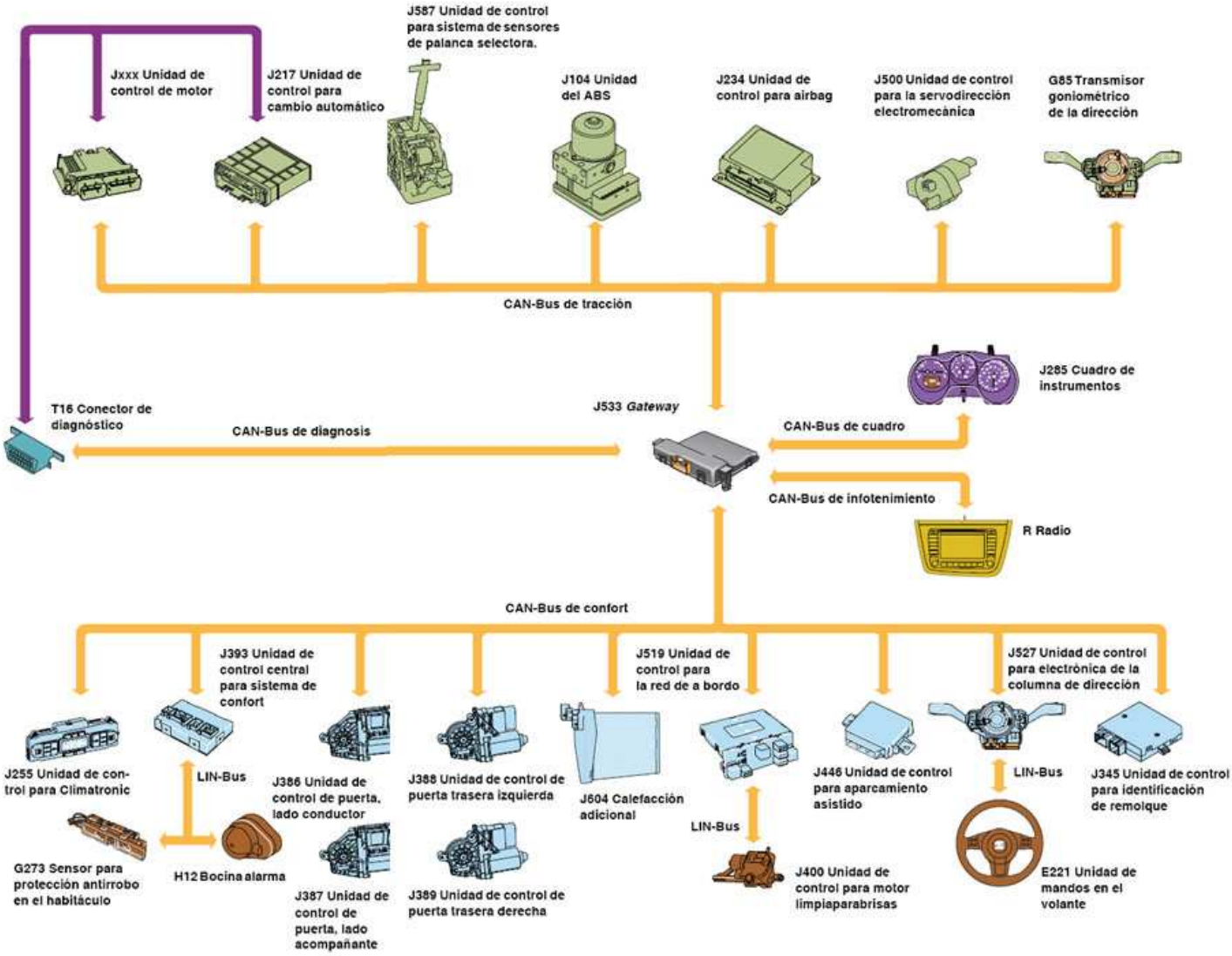
Las actuales líneas de CAN-Bus son las siguientes:

- de tracción,
- de confort.
- de infotenimiento (información y entretenimiento),
- de cuadro
- y de diagnosis.

Así como las diferentes de LIN-Bus:

- mandos del volante,
- limpiaparabrisas
- y alarma.

El diagnóstico se realiza a través del CAN-Bus, lo que aumenta la velocidad de transmisión y la cantidad de datos.



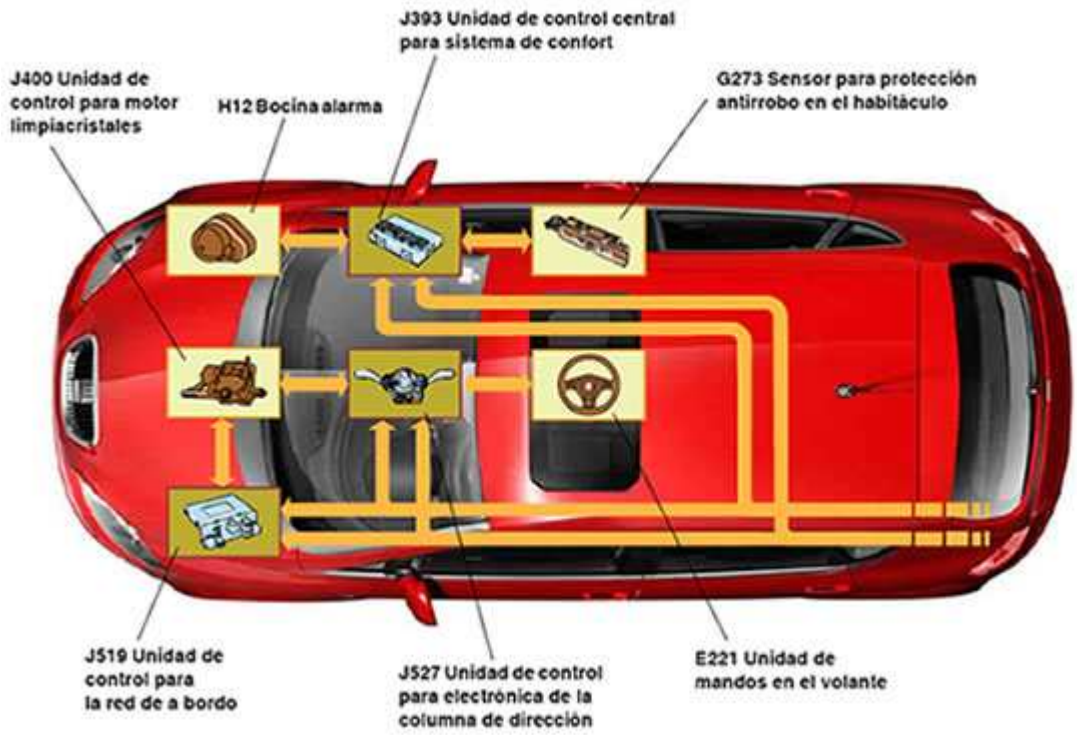
La velocidad de transmisión del CAN-Bus de tracción, cuadro y diagnosis es de 500 kbit/s mientras que el de infotenimiento y confort trabajan a una velocidad de 100 kbit/s.
En todos los casos, a excepción del CAN-Bus del cuadro, el cable "1ow" es de color naranja/marrón, y el cable "high" varía según la línea de CAN-Bus que se trate. Para tracción es naranja/negro, para confort naranja/verde, en infotenirniento naranja/lila, y para diagnosis naranja/negro.
En el cableado de CAN-Bus del cuadro el "high" es amarillo y el "low" es marrón.
Todas las líneas de CAN-Bus quedan comunicadas a través del gateway.

En el modelo Altea existen tres lineas de LIN-Bus:

- la primera, entre la unidad de control central de confort y la unidad de control para el motor lirnpiaparabrisas.
- la segunda, entre la unidad de control para la electrónica de la columna de dirección y la unidad de mandos en el volante
- y la tercera, entre la red de a bordo y la bocina de alarma y el sensor para protección antirrobo en el habitáculo.

Todas las lineas de LIN-Bus en altea, quedan comunicadas al CAN-Bus con las respectivas unidades maestras. La diagnosis de los sistemas LIN-Bus se realiza a través de la unidad de

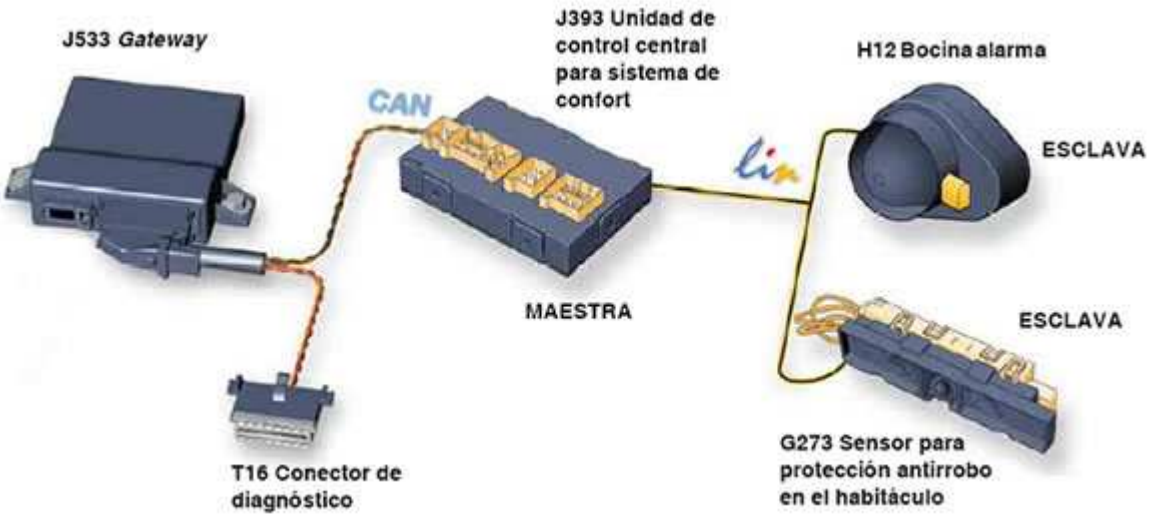
control maestra. La transmisión de los datos de diagnosis por parte de las unidades esclavas hacia la maestra se realiza a través del LIN-Bus.



Unidad maestra
La unidad de control que van conectada al CAN-Bus es la que realiza las funciones de maestra en el LIN-Bus.

Las funciones que tiene asignadas son:

- el control de la transmisión de datos y su velocidad,
- en el programa de la unidad se define un ciclo, según el cual se han de transmitir mensajes al LIN-Bus y se especifica cuales,
- asume la función de traducción entre las unidades de control abonadas a LIN y el CAN-Bus de datos. De esta forma es la única que esta conectada a su vez al CAN-Bus,
- y la diagnosis de las unidades de control LIN.

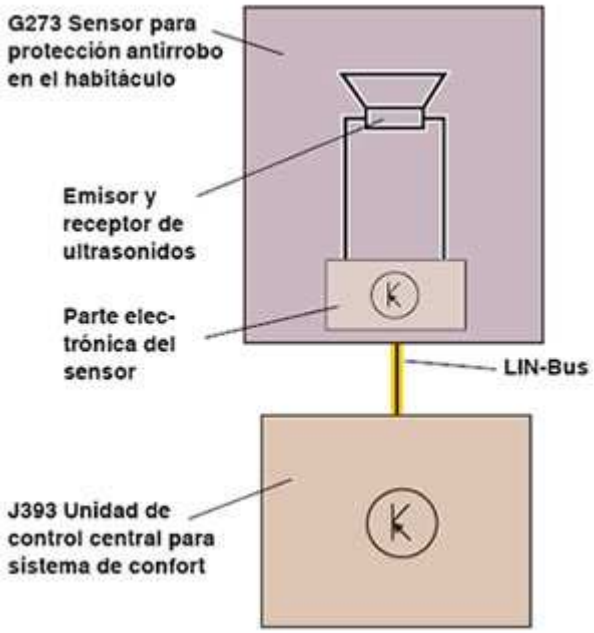


Unidad esclava

En un sistema de bus de datos LIN, la función de esclava la pueden realizar, tanto una unida de control como diferentes sensores o actuadores, por ejemplo el sensor volumétrico de la alarma antirrobo.

Los sensores llevan integrada una parte electrónica que analiza los valores medidos por el propio sensor. La transmisión de estos valores se realiza entonces a través del LIN-Bus en forma de señales digitalizadas.

Varias unidades esclavas se pueden conectar a un solo terminal de la unidad de control maestra del LIN-Bus.



The protocol

LIN is a serial single wire bus system. Each LIN network contains one master and a maximum of 16 slaves. Because LIN is a single wire protocol, it must operate asynchronously. Each member is responsible for its internal clock, which must be easy to generate. Expensive crystals are not feasible a RC- clock must do the job. This is also the reason for master - slave principle. The master has the task to perform all the synchronization of the slaves. The master must also provide the slaves with information, and catch the information of the slave. Only when the master gives an order to communicate, slaves may (talk to each other). The highest defined speed is approximately 20 kbits/s. however in practice usual speeds of serial communication are applied, namely 2400, 9600 and 19200 bits/s. Higher speeds is of course possible but in that case the system does not comply to the LIN 2.0 protocol anymore.

Differences with regard to LIN 1.3

Below an overview is given of the most important differences between LIN 1.3 and 2.0

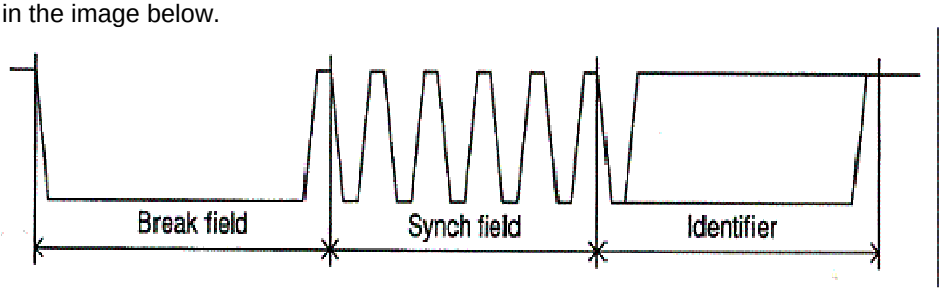
Byte array indicators is now supported, thus messages are now larger than 8 bits in one time are sent. The checksum calculation has been extended with the possibility taking indentifier into the calculation.

An automatic synchronization has been added in this specification.

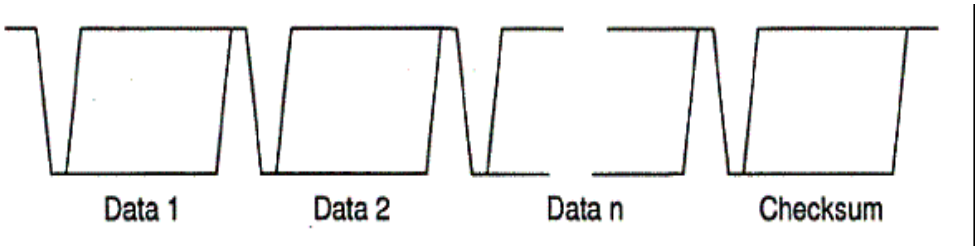
Signal groups have been removed.

LIN Frames

The communication between the units is done via sending LIN frames. The master sends a header (the first part of a frame) whereupon a response follows (Second part of the frame); the header is seen in the image below.



On the next image the response is showed.



The headers are generated by the master only. However the response can be generated by both the master and the slave. The slave can only respond if the master addresses the slave's header ID. The header exists of three parts: the synch break, a synchronization byte and an identifier. The break must warn the participant about a new frame is sent on the bus. This frame has 13 dominant bits (at

LIN is this logical "0") including start bit followed by a break delimiter. The break delimiter must be in any case 1 bit time high after the synch break follows a synchronization byte, which must synchronize the slaves. The contents of this byte are $55 (alternating 01 pattern) including start - and stop bit. The last part of the header is 8 bits identifiers Bit 0 till 5 are the ID bits, bit 6 is and 7 are the parity bits. These 6 ID bits result in a range of 0 up to 63 message ID's.

These have been as follows divided:

0 up to 59: user available
60 and 61: Reserved for diagnostic data
62: reserved for user additions 63: reserved for future improvements of the protocol.

The function of the two parity bits is ensuring that the identifier exist never entirely from 0-en or 1-en. Also each node can now check if the identifier is correctly received. In the original specification (rev. 1.1) bit 5 and 6 were used to indicate the length of the message. Now this happens by means of a configuration file, where all messages are specified. The parity bits in the identifier are calculated as follows:

P0 = ID0 ⊕ ID1 ⊕ ID2 ⊕ ID4
P1 = inv(ID1 ⊕ ID3 ⊕ ID4 ⊕ ID5)

After the identifier a response will be transmitted, which size is up to 8 bytes followed by a checksum How many bytes a response contains is specified in a configuration file. In this way every user on network knows the length of messages send over the network. The first byte that is sent is the LSB and the last is MSB. The last byte of response is the checksum. To calculate the checksum all the bytes must be added. To get the checksum the calculated sum has to be inverted. There is also an "enhanced "checksum. The indentifier is also taken along in calculation.

Example:

The following databytes will be sent; 0x4A, 0x55, 0x93 0xE5. The bytes are added up to carry arise. If this is the case the sum is increased by one and the carry will be cleared. After the last date byte is processed the outcome is inverted. The checksum is 0xE6. Members can calculate the checksum to make sure all the bytes are correctly received. In case of an enhanced checksum the identifier also will be taken along in calculation (in this example this would result in 0x4A).

Wake up en sleep mode

If there no bus activity in a 4 sec. time span, the node must go to sleep. Only if a node is in sleep he can receive a wake up call. To wake up a node the LIN bus must send a dominant signal during 250 ms to 5 ms, everything outside these limits is not a valid wake up. The recipient must a recognize this signal between 150 ms and 5 ms and after maximum 100 ms he must be ready for reception of the break. A second option to get a node in sleep is sending diagnostic message $3C, the first byte must be $00.

If the master does not react within 150 ms after the break, then the node will retry the wake up sequence three times. After the third time the node must wait for 1.5 sec. and it tries one more time. What hereafter happens is not stated in the specification. Usually the decision is made to put the mode in sleep.

Hardware: LIN transceiver.

The LIN communication goes by means of LIN transceiver, this component interface between the LIN and the RxD and TxD pins. Although their main function of these devices is the same yet there are small differences.

Manufacturer		Datasheet
Motorola		MC33399
Melexis		TH8082_006
Philips		TJA1020
On Semiconductor		NCV7361A

The most important characteristics of the Motorola MC33399:

Enable	:Driving this pin low the transceiver will be enabled.
Wake	:The wake up pin is one of the ways to detect a wake up signal. If the transceiver is in sleep mode and a edge has been detected the wake up pin will drive de inhibit pin low.
TxD and RxD	:These pins are respectively the signal to send data and receive data.
Inhibit	:This output becomes low when a wake up has been detected.
LIN	:This pin is wired to the LIN bus. If the transceiver is in sleep mode and a dominant signal is received, the inhibit pin will be driven high.

Hardware: LIN Microcontrollers:

Here also a number of variations are possible. At this moment MC68HC908GZ60 of Motorola and the uPD78F9852 of NEC are good choices.

Both controllers have an 8 bits databus. The major difference is that the Motorola comes with an ESCI (Enhanced Serial Communications interface) and the NEC has an ordinary UART interface. The advantage of an ESCI are that specific LIN options directly can be addressed. THE ESCI can for example generate and detect a break, properties which ordinary UART does not support.

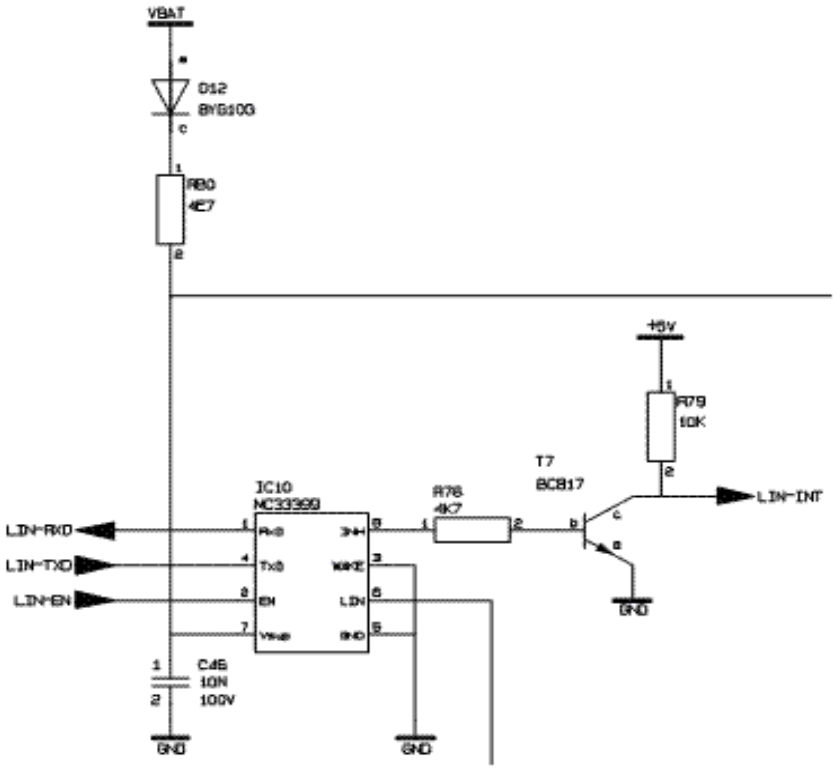
Hardware: LIN Microcontrollers:

INFINEON	

Toshiba	TOSHIBA
Hitachi	HITACHI Inspire the Next

Hardware "master" design:

Such as said there is only one master in the LIN system. This means in terms or hardware that the LIN transceiver, besides the internal capacitance, must be connected to an extra resistor of 1 kOhm. This resistor must be placed between the LIN bus (pin6) and the Vbat (pin7). The transceiver is linked directly to the RxD and the TxD pins UART pins of the microcontroller. In the figure below this configuration is shown. The inhibit pin is switched to the micro by means or a transistor. The pin is fixed to ground. This only endures in that the LIN transceiver by means or a dominantly indicator on the LIN bus sleep can come. As the transceiver sleep must be obtained there a dominant vigil up indicator on the LIN bus will appear, afterwards the inhibit pin will become high. This edge generates an interrupt in the microcontroller whereupon the software makes transceiver high. The remaining component around the transceiver serves for protection and reducing of noise in signals Hardware "slave" design: With regard to hardware the hardware or the master differs from slave on the following points: The inhibit pin is switched to the power supply by means or a transistor. The slave configuration is not terminated with a resistor. The wake up pin of the slave is wired to the connector.

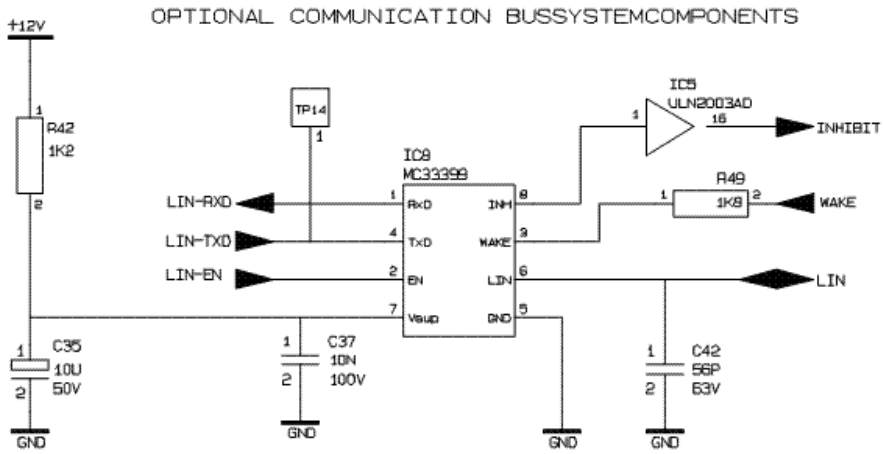


configuration is shown

The inhibit pin is connected by means of a transistor on the microcontroller. The pin has been laid to ground. This only endures in that the LIN transceiver by means of a dominant indicator on the LIN bus sleep can come. As the transceiver sleep must be obtained there a dominant vigil up indicator on the LIN bus will appear, afterwards the inhibit pin will become high. This edge generates an interrupt in the microcontroller whereupon the software makes transceiver high. The remaining components around the transceiver serve for protection and reduce noise in signals

Hardware “slave” design:

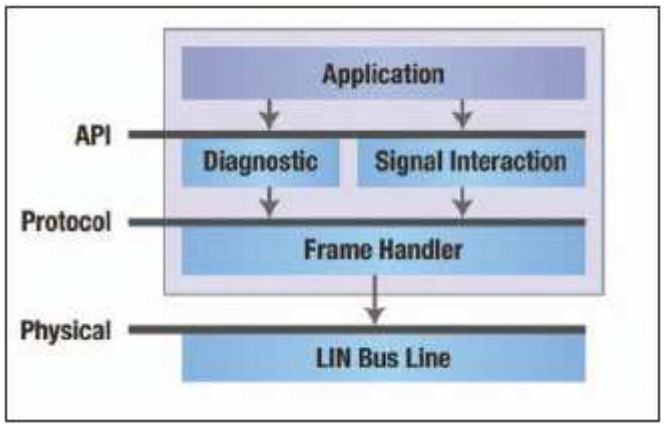
With regard to hardware the hardware of the master differs from slave on the following points:
The inhibit pin is switched to the power supply by means of a transistor.
The slave configuration has no terminated with a resistor.
The wake up pin of the slave is wired to the connector.



LIN Protocol Overview:

Topology:

- **Broadcast serial network**
- **Single Master/multiple Slave Concept**
  - the master node (task) can be assigned to perform both master and slave operation
  - the medium access in a LIN base network is controlled by a master node so that no arbitration or collision management slave node is required.
  - true “plug-and-play” slave nodes implementation due to standardizesyntax for the specification by enhanced LIN features and LIN - node compatibility language.
- **12v Bus with 16 nodes or less**
  - based upon LIN specification and standard driver characteristics
  - LIN is limited to 64 identifiers and relatively low transmission speed.



LIN Protocol Overview:

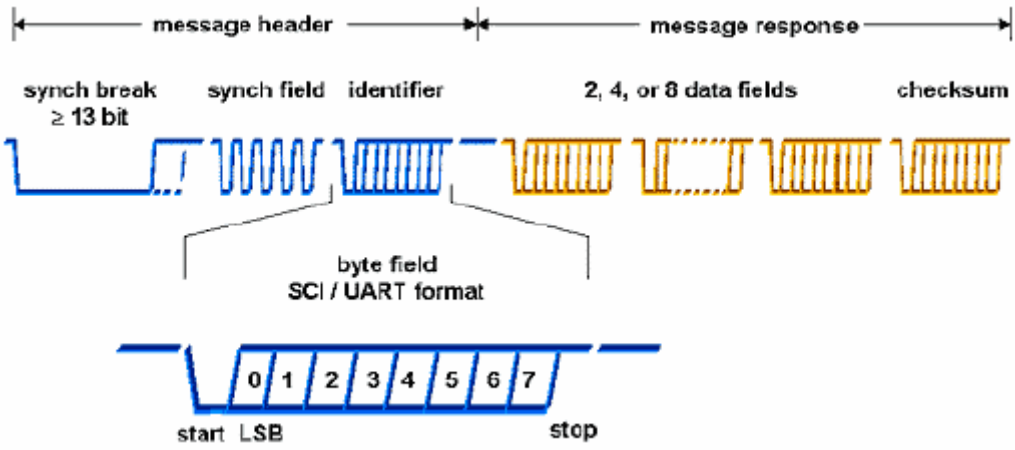
- **Master Task**
  - control over the whole Bus and Protocol: The master controls which message at what time is to be transferred over the bus. It can also do the error handling.

- to accomplish this the master

- sends Sync Break, Sync Byte & ID-Field
- monitors Data Bytes and Check Byte, and evaluates them on consistence
- receives Wakeup Break from slave nodes when the bus is inactive and they request some action
- serves as a reference with it's clock base (stable clock necessary)
- **Slave Task**

  - one of 2-16 members on the bus and receives or transmits data when an appropriate ID is sent by the master.
- Slave waits for Sync Break
- Slave synchronizes on Sync Byte
- Slave snoops for ID.
- According to ID, slave determines what to do: either receive data, or transmit data or do nothing.

- When transmitting, the slave sends 2, 4, or 8 Data Bytes & Check-Byte.
- The node serving as a master can be slave.



LIN Protocol Overview:

- **Message Frame Header**

- **Sync Break:**

Marks the Beginning of a Message Frame (less than 13 bit)

- Synch Field:

Specific Pattern for Determination of Time Base(Determination of the time between two rising edges)

- ID-Field:

- Message Identifier: Incorporates Information about the sender, thereceiver(s), the purpose, and the Data field length.
- Length 6 Bit.
- 3 classes of 2/4/8 Data Bytes. The length coding is in the
- 2 MSB of the ID-Field. A total of 64 Message Identifiers ispossible.
- 2 linked Parity Bits protect this highly sensitive ID-Field.
- **Message Frame Header**

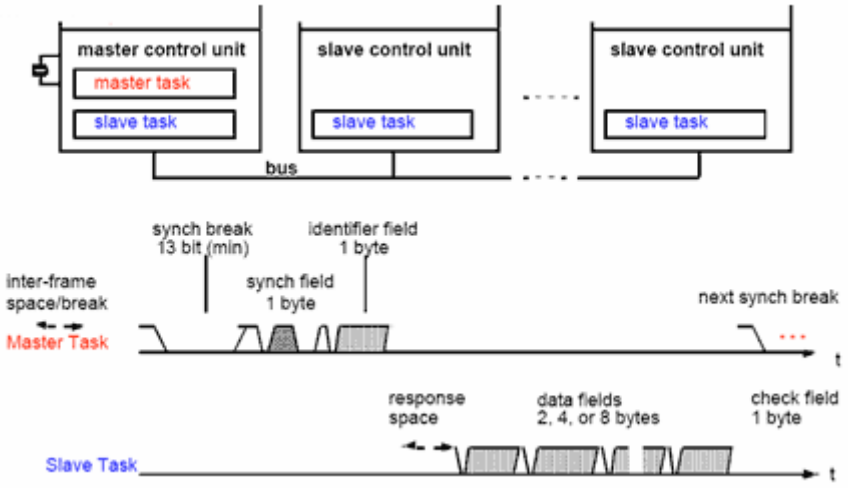
- Data Field:

- 1-8 bytes in length, as determined during system configuration of the message frame ID's.
- may consist of one or more 'signals' appended together.

- **Checksum Field:**

- consist of single byte.

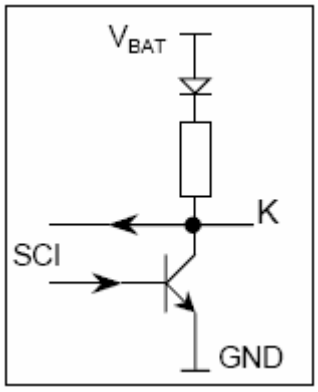
- defined as the 1’s complement sum of all data bytes.
- LIN 1.3 and Diagnostic message frames: classic checksum.
- LIN 2.0: Extended checksum-all data bytes plus the protected ID byte.

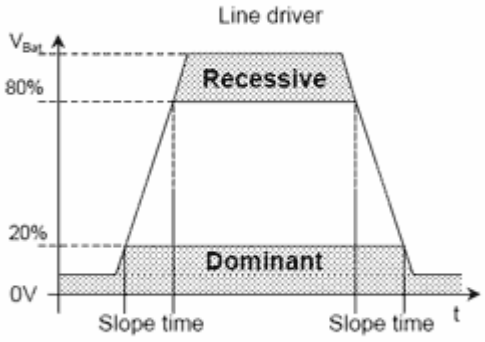


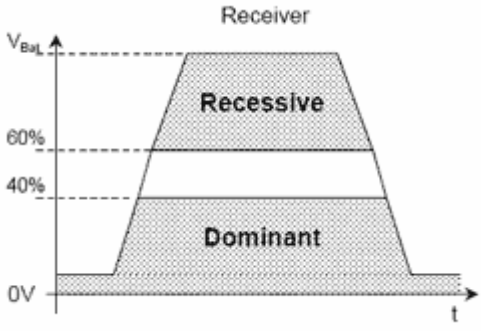
LIN Protocol Overview:

Physical Layer:

- **single-wire: ISO 9141 compliant**
  - max 20% / min 80% VBAT Low/High transmit level
  - min 40% / max 60 % VBAT Low/High receive threshold
  - controlled slew rate (1-2 V/µs)







LIN Protocol Overview:

LIN Transfer Speed:

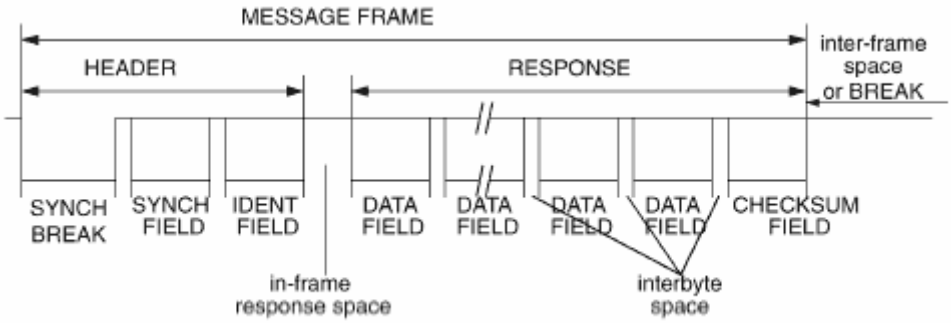
- **Speed up to 20K bit/sec:**
  - Acceptable speed for many application (limited for EMI-reasons)
  - The recommended data rates are 2400bps, 9600bps and 19200 bps.
  - controlled slew rate (1-2 V/µs)
- **Data Format:**
  UART/SCI base
  - Hardware or software (with standard I/O port)
  - Low cost silicon implementation based upon common UART/SCI interface

LIN Protocol Overview:

Guaranteed signal-transmission latency times:

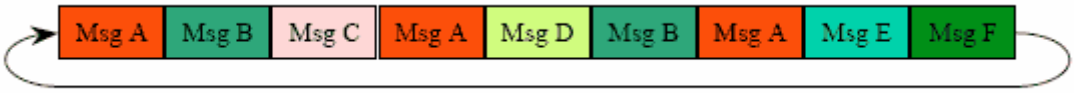
- **Time Triggered Approach**

- Message Length is known
  - Number of transmitted data bytes is known minimum length can be calculated
  - Each Message has length budget of 140% of it's minimum length
    - maximum allowed length is known
    - distance between beginning of two messages

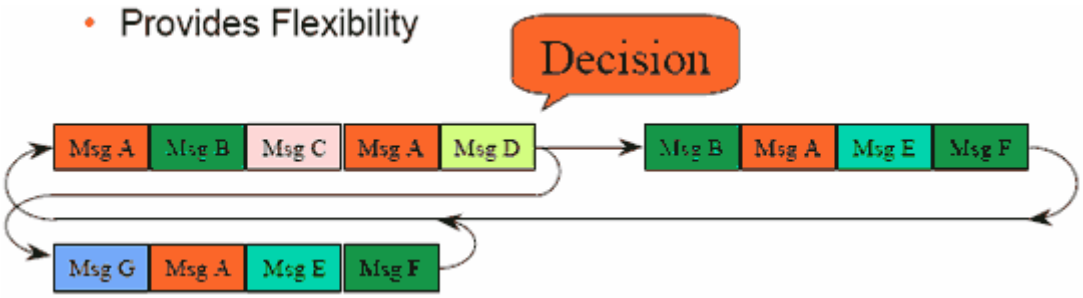


- Message sequence is known

- **Master uses scheduling table**

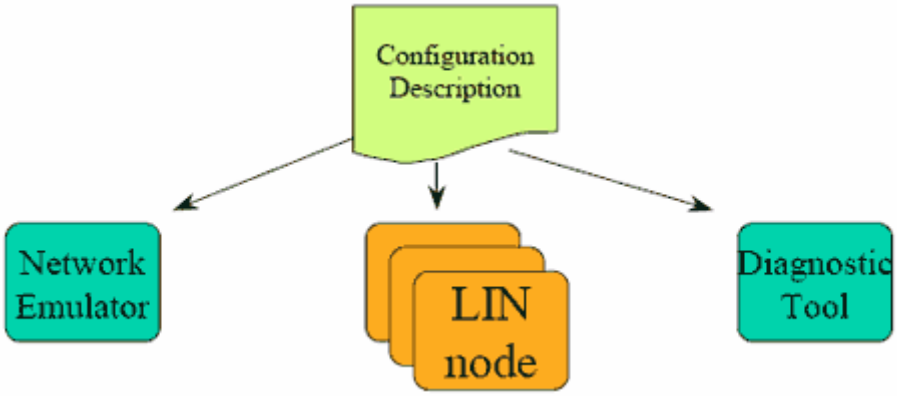


- Use of different scheduling tables is possible

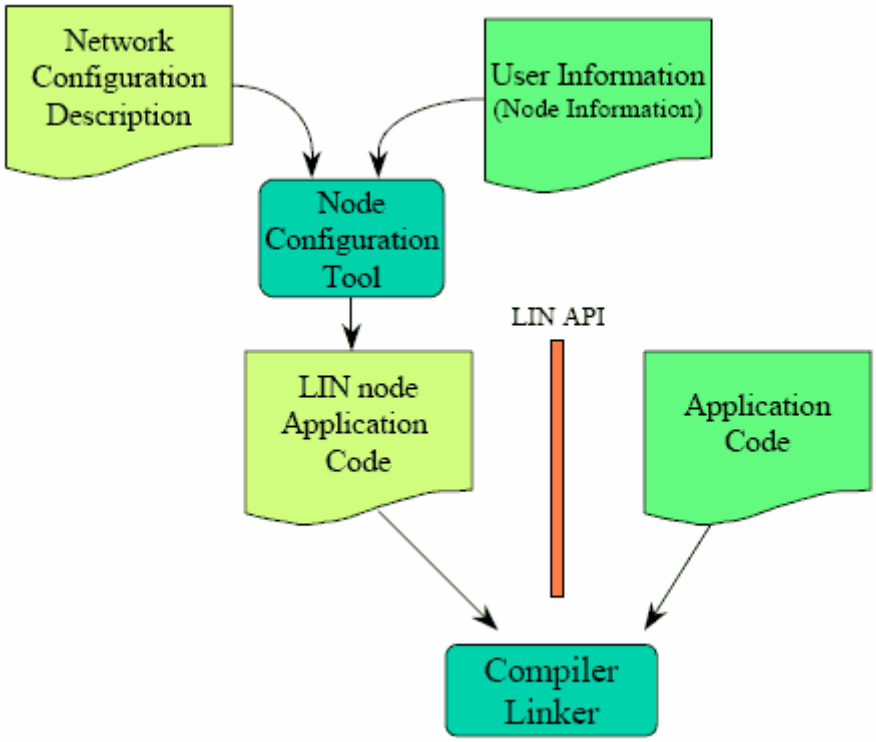


Network configuration:

- **Standard LIN Description File**
  - describes complete LIN network and also contains all information necessary to monitor the network.



- Standard LIN Configuration Language
  - allows nodes to be used for various purposes without jeopardizing the LIN system functionality by message incompatibility or network overload.
  - is useful for debugging LIN clusters or diagnose the traffic. - provides “plug-and-play” capability with any off-the-shelf slave node MCUs.
- Standard LIN API
  - simplifies Application code design.
  - enables to select any controllers that support LIN peripheral

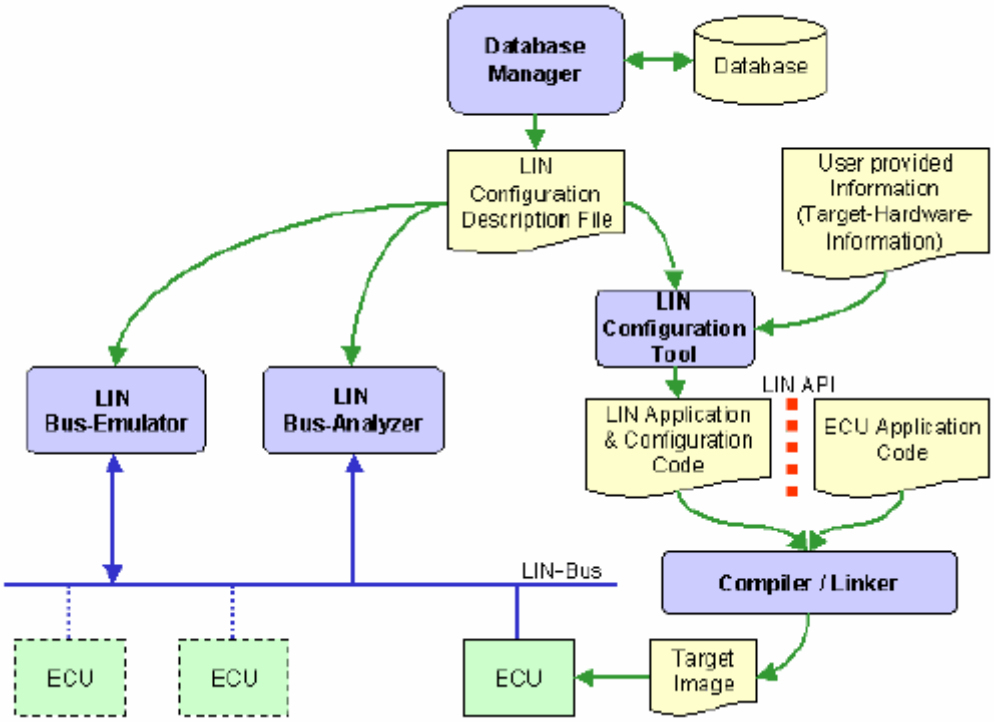


Dynamic configuration:

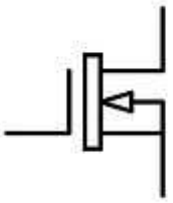
- Standard LIN Tool
  - **LIN Database Manager (LDM):**
  The LDM is a standalone offline tool, providing a user-friendly Windows interface for logically describing and configuring LIN systems at a high abstraction level.

- LIN Configuration Tool (lcfg):
The LIN API, together with the LIN Configuration Tool and an optimized embedded SW package enables the user to get correctness and quality together with efficiency and reconfiguration flexibility.

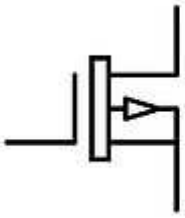
- LINspector:
a highly flexible tool for testing and verifying communication for compliance with the LIN standard.



MOSFET



Transistor MOSFET de empobrecimiento canal N.



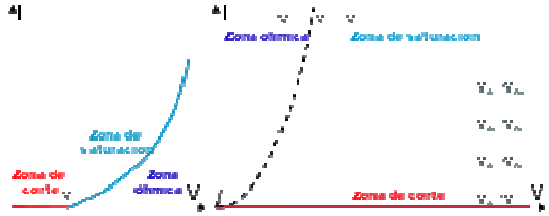
Transistor MOSFET de empobrecimiento canal P.

MOSFET son las siglas de **Metal Oxide Semiconductor Field Effect Transistor**. Consiste en un transistor de efecto de campo basado en la estructura MOS. Es el transistor más utilizado en la industria microelectrónica. Prácticamente la totalidad de los circuitos integrados de uso comercial están basados en transistores MOSFET.

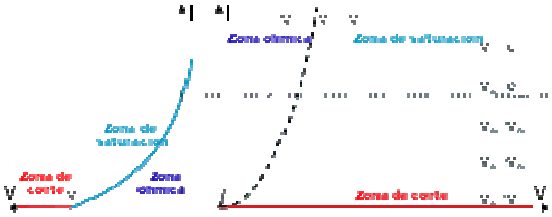
Historia

Fue ideado teóricamente por el alemán Julius von Edgar Lilienfeld en 1930, aunque debido a problemas de carácter tecnológico y el desconocimiento acerca de cómo se comportan los electrones sobre la superficie del semiconductor no se pudieron fabricar hasta décadas más tarde. En concreto, para que este tipo de dispositivos pueda funcionar correctamente, la intercara entre el sustrato dopado y el aislante debe ser perfectamente lisa y lo más libre de defectos posible. Esto es algo que sólo se pudo conseguir más tarde, con el desarrollo de la tecnología del silicio.

Funcionamiento



Curvas característica y de salida de un transistor MOSFET de acumulación canal n.



Curvas característica y de salida de un transistor MOSFET de deplexión canal n.

Un transistor MOSFET consiste en un sustrato de material semiconductor dopado en el que, mediante técnicas de difusión de dopantes, se crean dos islas de tipo opuesto separadas por un área sobre la cual se hace crecer una capa de dieléctrico culminada por una capa de conductor. Los transistores MOSFET se dividen en dos tipos fundamentales dependiendo de cómo se haya realizado el dopaje:

- Tipo **nMOS**: Sustrato de tipo **p** y difusiones de tipo **n**.
- Tipo **pMOS**: Sustrato de tipo **n** y difusiones de tipo **p**.

Las áreas de difusión se denominan **fuente**(source) y **drenador**(drain), y el conductor entre ellos es la **puerta**(gate).

El transistor MOSFET tiene tres estados de funcionamiento:

Estado de corte

Cuando la tensión de la puerta es idéntica a la del sustrato, el MOSFET está en estado de no conducción: ninguna corriente fluye entre fuente y drenador. También se llama mosfet a los aislados por juntura de dos componentes.

Conducción lineal

Al polarizarse la puerta con una tensión negativa (pMOS) o positiva (nMOS), se crea una región de deplexión en la región que separa la fuente y el drenador. Si esta tensión crece lo suficiente, aparecerán portadores minoritarios (electrones en pMOS, huecos en nMOS) en la región de deplexión que darán lugar a un canal de conducción. El transistor pasa entonces a estado de conducción, de modo que una diferencia de potencial entre fuente y drenador dará lugar a una corriente. El transistor se comporta como una resistencia controlada por la tensión de puerta.

Saturación

Cuando la tensión entre drenador y fuente supera cierto límite, el canal de conducción bajo la puerta sufre un estrangulamiento en las cercanías del drenador y desaparece. La corriente entre fuente y drenador no se interrumpe, ya que es debido al campo eléctrico entre ambos, pero se hace independiente de la diferencia de potencial entre ambos terminales.

Modelos matemáticos

- Para un MOSFET de canal inducido tipo n en su región lineal:

$$I_{D(Act)} = K[(V_{GS} - V_T)V_{DS} - \frac{V_{DS}^2}{2}]$$

donde $K = \frac{b\mu_n\epsilon}{LW}$ en la que b es el ancho del canal, μ_n la movilidad de los electrones, ε es la permitividad eléctrica de la capa de óxido, L la longitud del canal y W el espesor de capa de óxido.

- Cuando el transistor opera en la región de saturación, la fórmula pasa a ser la siguiente:

$$I_{D(Sat)} = \frac{K+1}{K_0}(V_{GS} - V_T)^2$$

Estas fórmulas son un modelo sencillo de funcionamiento de los transistores MOSFET, pero no tienen en cuenta un buen número de efectos de segundo orden, como por ejemplo:

- Saturación de velocidad: La relación entre la tensión de puerta y la corriente de drenador no crece cuadráticamente en transistores de canal corto.
- Efecto cuerpo o efecto sustrato: La tensión entre fuente y sustrato modifica la tensión umbral que da lugar al canal de conducción
- Modulación de longitud de canal.

Aplicaciones

La forma más habitual de emplear transistores MOSFET es en circuitos de tipo CMOS, consistentes en el uso de transistores pMOS y nMOS complementarios.

Las aplicaciones de MOSFET discretos más comunes son:

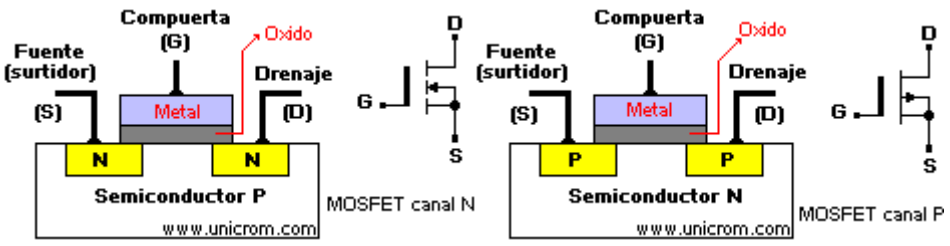
- Resistencia controlada por tensión.
- Circuitos de conmutación de potencia (HEXFET, FREDFET, etc).
- Mezcladores de frecuencia, con MOSFET de doble puerta.

Ventajas

La principal aplicación de los MOSFET está en los circuitos integrados, p-mos, n-mos y c-mos, debido a varias ventajas sobre los transistores bipolares:

- Consumo en modo estático muy bajo.
- Tamaño muy inferior al transistor bipolar (actualmente del orden de media micra).
- Gran capacidad de integración debido a su reducido tamaño.
- Funcionamiento por tensión, son controlados por voltaje por lo que tienen una impedencia de entrada muy alta. La intensidad que circula por la puerta es del orden de los nanoamperios.
- Un circuito realizado con MOSFET no necesita resistencias, con el ahorro de superficie que conlleva.
- La velocidad de conmutación es muy alta, siendo del orden de los nanosegundos.
- Cada vez se encuentran más en aplicaciones en los convertidores de alta frecuencias y baja potencia.

En el **MOSFET** de canal N la parte "N" está conectado a la fuente (source) y al drenaje (drain)
En el **MOSFET** de canal P la parte "P" está conectado a la fuente (source) y al drenaje (drain)



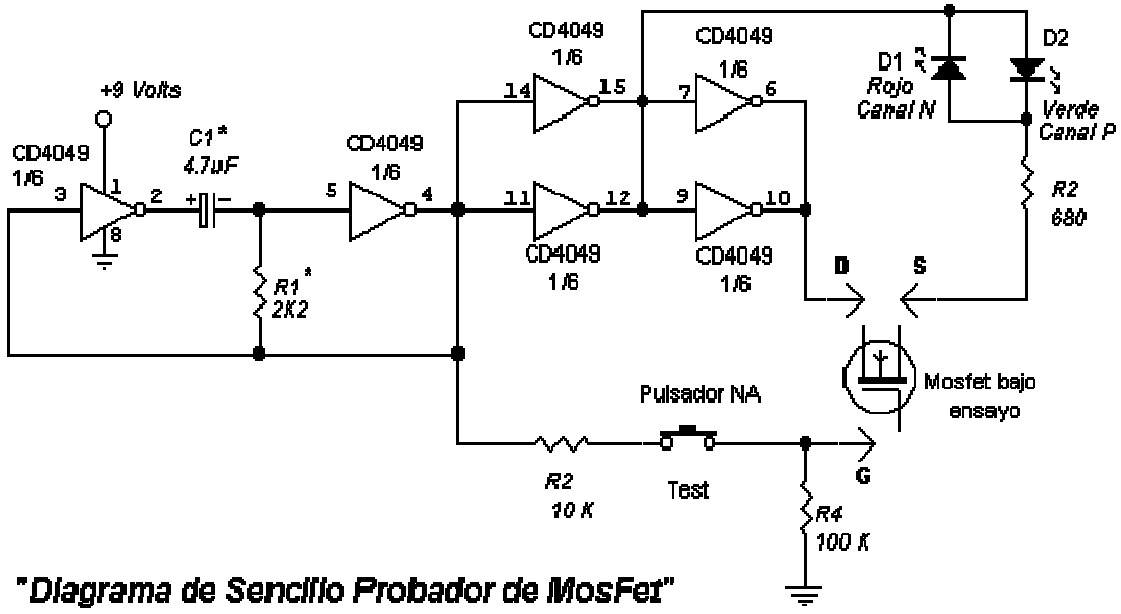
En los transistores bipolares la corriente que circula por el colector es controlada por la corriente que circula por la base. Sin embargo en el caso de los **transistores FET**, la corriente de salida es controlada por una tensión de entrada (un campo eléctrico). En este caso no existe corriente de entrada.

Los **transistores MOSFET** se pueden dañar con facilidad y hay que manipularlos con cuidado. Debido a que la capa de óxido es muy delgada, se puede destruir con facilidad si hay alta tensión o hay electricidad estática.

Este proyecto de "sencilla construcción" permite comprobar el estado de los Mosfet (tipo IRF630; PH6N60; etc), de los cuales es bastante difícil determinar su estado, salvo cuando estos presentan "cortocircuito" entre sus terminales, en ese caso es muy fácil de determinarlo con el multímetro o tester.
El circuito es de tal sencillez que podría ser armado en protoboard en sólo 10 minutos (aprox.); con los componentes a disposición.

Funcionamiento:
Consiste en un oscilador astable formado por las dos compuertas izquierdas en el diagrama y cuya frecuencia de oscilación viene determinada por los valores de **R1** y **C1** (en este caso una frecuencia cercana a 140 Hertz para evitar el clásico y para mí, molesto parpadeo).
Si el colega quiere bajar la frecuencia (para "destello" por ejemplo) puede hacerlo mediante la fórmula de los osciladores astables:
f =1 /(0,7 * R1 *C1) [Hz]
Donde **R1** [ohms] y **C1** [Farads]; y con valores R1=100K y C1= 4,7uF, se obtiene el efecto destello a frecuencia cercana al Hertz.
Nota: C1 conviene que no sea mayor a 10uF por las "elevadas corrientes de fugas" que se presentan, comparables a la corriente inicial de carga de este capacitor en muchos casos. (El capacitor se comportaría como un cortocircuito y nunca se cargaría!).

Los inversores siguientes en pares paralelos (Buffers) aseguran el correcto funcionamiento al entregar la corriente de excitación necesaria a los LED e invirtiendo el sentido de la corriente a través del transistor (drenador-surtidor) en cada semiperiodo de oscilación y solamente cuando la excitación en la compuerta sea la apropiada con "pulsador activado" y el transistor esté en buen estado, se encenderá el LED correspondiente, indicando su polaridad (Canal N ó Canal P).



"Diagrama de Sencillo Probador de MosFet"

Lista de materiales:
C1 - Capacitor 4,7uF * (16Volts mínimo)
R1 - Resistencia 2200ohm 1/4W
R2 - Resistencia 10Kohm 1/4W
R3 - Resistencia 680ohm 1/4W
R4 - Resistencia 100 Kohm 1/4W
IC - CMOS CD4049
D1 - LED Rojo
D2 - LED Verde (o colores y tamaños a elección o disposición)
Pulsador: NA (Normal Abierto)
Bateria de 9Volts; zócalo para transistores, conectores, etc.

Modo de Uso:
Consiste en **conectar correctamente los terminales D, G y S** del transistor MOS-FET en los correspondientes terminales del probador y verificar lo siguiente (de acuerdo al diagrama):

I) TRANSISTOR EN BUEN ESTADO:
a) *"Transistor c/ diodo interno surtidor-drenador".*
Si el "LED verde" enciende (debido a presencia del diodo interno) antes de presionar el pulsador y luego de "presionar" el mismo es acompañado por el "LED Rojo" (Canal N), significa que el transistor de "canal N" y su correspondiente diodo surtidor-drenador se encuentran en BUEN ESTADO.
El caso "inverso" significa que un transistor "canal P" con diodo interno (S-D) está en BUEN ESTADO.
b) Si el transistor carece de diodo entre surtidor y drenador, solo el "LED Rojo" encenderá luego de presionar el pulsador, si éste es de "canal N" y se encuentra en BUEN ESTADO; lo inverso ("LED verde" enciende solamente c/ pulsador activado) se cumpliría para un transistor de "canal P" en las mismas condiciones.

II) TRANSISTOR EN CORTOCIRCUITO (malo):
En caso de estar el transistor en CORTOCIRCUITO, se produce el "encendido" de "ambos" LED sin necesidad de presionar el pulsador. (Esto es más rápido y práctico determinarlo con el buzzer o comprobador de continuidad del tester!).

III) TRANSISTOR ABIERTO (malo):
En caso de transistor ABIERTO tanto con el pulsador activado como sin activarlo, "ambos" diodos permanecen "apagados". (En este caso convendría hacer un ligero corto entre terminales D y S del probador y al producirse el "encendido de ambos LED" nos aseguramos el estado medido del transistor).