

Estudio del bus de comunicaciones CAN

Francisco Javier Blanco Curieses

Máster Universitario en Ingeniería de Telecomunicación
TFM Electrónica

Nombre Consultor/a: Aleix López Antón

Nombre Profesor/a responsable de la asignatura: Carlos Monzo Sánchez

Enero 2019



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Estudio del bus de comunicaciones CAN</i>
Nombre del autor:	<i>Francisco Javier Blanco Curieses</i>
Nombre del consultor/a:	<i>Aleix López Antón</i>
Nombre del PRA:	Carlos Monzo Sánchez
Fecha de entrega (mm/aaaa):	01/2019
Titulación:	<i>Máster Universitario Ingeniería de Telecomunicación</i>
Área del Trabajo Final:	<i>TFM Electrónica</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>CAN, Comunicaciones, Estándar</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>Debido a los avances producidos en los procesos industriales los diferentes tipos de redes se han visto obligadas a gestionar una gran cantidad de información. Estas redes han permitido administrar esta información en los diferentes campos de la industria. Esto ha permitido el monitoreo, supervisión y control de los parámetros y variables utilizados. La evolución tecnológica producida a lo largo de los últimos años ha puesto a nuestra disposición herramientas software y hardware que han permitido centralizar los datos obtenidos de diferentes sensores que se utilizan actualmente en la industria como: temperatura, nivel, presión, pH, movimiento, distancia, inclinación, aceleración, fuerza, intensidad lumínica, entre otros. Esta información es transportada a los centros de control mediante buses de comunicaciones. Esto ha permitido simplificar enormemente la instalación, operación de máquinas y equipamientos industriales que se utilizan hoy en día den los procesos de producción permitiendo evitar el desperdicio de los recursos y de esta manera hacerlos más eficientes y rentables.</p> <p>En el presente trabajo fin de máster realizaremos un estudio pormenorizado de uno de esos buses de campo utilizados en la actualidad. Concretamente estudiaremos el bus de comunicaciones CAN. Este bus fue diseñado inicialmente para las comunicaciones entre los diferentes dispositivos integrados en los automóviles, pero con el tiempo debido a su eficiencia, bajo coste y a la estandarización que se produjo del mismo, esto sumado a otros muchos factores ha permitido que el uso de este bus se halla generalizado en multitud de sectores.</p>	

Abstract (in English, 250 words or less):

Due to the advances produced in the industrial processes, The different types of networks have been forced to manage a large amount of information. These networks have allowed fields of the industry. This has allowed the monitoring, supervisión and control of the parameters and variables used. The technological evolution produced in recent years has made available software and hardware tools that have allowed us to centralize the data obtained from different sensors currently used in the industry such as: temperature, level, pressure, pH, movement, distance, inclination, acceleration, force, light intensity, among others. This information is transported to the control centers by means of communication buses. This has made it possible to greatly simplify the installation, operation of industrial machines and equipment that are used today in the production processes, thus avoiding the waste of resources and thus making them more efficient and profitable.

In the present master's thesis work we will carry out a detailed study of one of those field buses used today. Specifically, we will study the CAN communication bus. This bus was initially designed for communications between the different devices integrated in cars, but over time due to its efficiency, low cost and the standardization that was produced, this added to many other factors has allowed the use of this bus is widespread in many sectors.

Índice

1. Introducción.....	1
1.1. Contexto y justificación del Trabajo.....	1
1.2. Objetivos del Trabajo.....	1
1.3. Enfoque y método seguido.....	1
1.4. Planificación del Trabajo.....	2
1.4.1. Definición TFM.....	3
1.4.2. Estado del Arte.....	3
1.4.3. Diseño e implementación.....	3
1.4.4. Memoria.....	3
1.4.5. Presentación.....	3
1.4.6. Diagrama de Gantt.....	3
1.5. Breve resumen de productos obtenidos.....	4
1.6. Breve descripción de los otros capítulos de la memoria.....	4
2. Estado del arte.....	6
2.1. Modelo OSI de 7 capas.....	6
2.1.1. Capa Física.....	7
2.1.2. Capa de Enlace de Datos.....	7
2.1.3. Capa de Red.....	8
2.1.4. Capa de Transporte.....	8
2.1.5. Capa de Sesión.....	8
2.1.6. Capa de Presentación.....	8
2.1.7. Capa de Aplicación.....	8
2.2. Modos de transmisión.....	9
2.2.1. Simplex.....	9
2.2.2. Semidúplex (half-duplex).....	9
2.3. Sistemas distribuidos de comunicaciones de tiempo real.....	9
2.4. Topologías.....	10
2.4.1. Anillo.....	10
2.4.2. Topología de anillo doble.....	11
2.4.3. Árbol.....	12
2.4.4. Bus.....	12
2.2.5. Estrella.....	13
2.4.6. Malla.....	14
2.4.7. Híbrida.....	14
2.5. Buses actuadores y sensores.....	14
2.6.1. Arquitectura de nodo.....	15
2.6.2. Características y ámbitos de aplicación.....	15
2.6.3. Arquitecturas de comunicación.....	17
2.6.4. Buses de campo estandarizados.....	18
2.7. El bus de campo PROFIBUS.....	20
2.8. El bus de campo WorldFIP.....	21
2.9. El bus de comunicaciones CAN.....	24
3. Introducción al bus CAN.....	26
3.1. Historia.....	26
3.2. Estándar CAN 2.0.....	28
3.4. Características básicas del bus CAN.....	30

4. Niveles OSI de ISO bus CAN	33
4.1 Capa física	33
4.1.1 Codificación de bit	33
4.1.2. Estándar ISO 11519	35
4.1.3. Estándar ISO 11898	36
4.1.4. Características del cable y conectores CAN	37
4.2. Tiempo de Bit y Sincronización	40
4.2.1. Tipos de sincronización	41
4.2.1.1 Hard sincronización	41
4.2.1.2. Resynchronization	41
4.3. Capa de Enlace	43
4.3.1. Subcapa MAC	43
4.3.2. Subcapa LLC	45
4.3.3. Tramas	46
4.3.3.1. Trama de datos	47
4.3.3.2. Trama remota	48
4.3.3.3. Trama de error	49
4.3.3.4. Trama de sobrecarga	51
4.3.3.5. Espacio entre tramas	52
5. Ejemplo de nodo CAN	53
5.1. Raspberry Pi	53
5.1.1. Hardware	54
5.1.1.1. Puertos USB	57
5.1.1.2. Conector GPIO	58
5.1.1.3. Conector DSI	60
5.1.1.4. Conector CSI	61
5.1.1.5. Ranura para tarjeta microSD	61
5.1.1.6. Puerto Ethernet	61
5.1.1.7. Alimentación	62
5.1.1.8. LEDs indicadores	62
5.1.1.9. Salidas de audio	64
5.1.1.10. Conector HDMI	64
5.1.1.11 SoC	64
5.1.2. Software	64
5.1.2.1. Instalación del Sistema Operativo	67
5.2. Nodo CAN	68
5.2.1. Software utilizado	68
5.2.2. Esquemático	69
5.2.3. Listado de componentes	70
5.2.3.1. MCP2515-I/P	71
5.2.3.2. MCP2551-I/P	73
5.2.4. Diseño PCB	73
5.2.5. Proceso de fabricación	76
5.2.6. Configuración del Software	78
5.2.7. Prueba de funcionamiento	79
6. Presupuesto	81
7. Conclusiones	83
8. Glosario	84
9. Bibliografía	86

Lista de figuras

Ilustración 1 - Diagrama WBS	2
Ilustración 2 - Diagrama de Gantt.....	4
Ilustración 3 - Modelo OSI de ISO.....	7
Ilustración 4 - Topología en anillo.....	11
Ilustración 5 - Topología anillo doble.....	11
Ilustración 6 - Topología en árbol.....	12
Ilustración 7 - Topología en bus	13
Ilustración 8 - Topología en estrella	13
Ilustración 9 - Topología en estrella	14
Ilustración 10 - Arquitectura de un nodo.....	15
Ilustración 11 - Ejemplo bus WorldFIP [10]	23
Ilustración 12 - Ejemplo bus CAN SEAT altea [11].....	27
Ilustración 13 - Formato base.....	28
Ilustración 14 - Formato extendido	29
Ilustración 15 - Codificación NRZ.....	33
Ilustración 16 - Codificación RZ.....	34
Ilustración 17 - Método de violación digital.....	34
Ilustración 18 - Niveles voltaje estándar ISO 11519	35
Ilustración 19 - Conexión de nodos a bus CAN.....	36
Ilustración 20 - Niveles de voltaje estándar 11898	36
Ilustración 21 - Conexión de nodos a bus CAN.....	37
Ilustración 22 - D-sub de 9 pines.....	38
Ilustración 23 - Elementos conector M-12	38
Ilustración 24 - Proceso de montaje conector M12	39
Ilustración 25 - Visión frontal de conector M12.....	40
Ilustración 26 - Tiempo de bit y sincronización.....	40
Ilustración 27 - Resincronización.....	42
Ilustración 28 - Resincronización.....	42
Ilustración 29 - Resincronización.....	42
Ilustración 30 - Resolución de colisión	44
Ilustración 31 - Trama de datos estandar	47
Ilustración 32 - Trama de datos extendida	47
Ilustración 33 - Trama remota	49
Ilustración 34 - Trama de error	49
Ilustración 35 - Trama de sobrecarga.....	51
Ilustración 36 - Imagen Raspberry Pi	55
Ilustración 37 - Descripción componentes principales Raspberry Pi.....	56
Ilustración 38 - Cara de pistas Raspberry Pi	57
Ilustración 39 - LAN7500/LAN7500i [23]	58
Ilustración 40 - LAN9514/LAN9514i [24]	58
Ilustración 41 - Pines conector GPIO	59
Ilustración 42 - Configuración pines GPIO	59
Ilustración 43- Conector de alimentación Raspberry Pi.....	62
Ilustración 44 - Ranura SD y Tarjeta microSD con adaptador.....	67
Ilustración 45 - Imagen Win32 Disk Imager.....	68
Ilustración 46 - Captura herramienta Eeschema	69
Ilustración 47 - Esquema utilizado para obtener PCB [37]	70

Ilustración 48 - Captura componentes esquema	71
Ilustración 49 - Chip MCP2515.....	72
Ilustración 50 - Diagrama de bloques chip MCP2515	72
Ilustración 51 - Chip MCP2551	73
Ilustración 52 - Chip MCP2551	73
Ilustración 53 - Captura Pcbnew	74
Ilustración 54 - Placa de circuito impreso	74
Ilustración 55 - Lado de pistas de PCB	75
Ilustración 56 - Lado de componentes de PCB	76
Ilustración 57 - Circuito impreso uno ya montado	77
Ilustración 58 - Circuito impreso dos ya montado.....	77
Ilustración 59 - Esquema de conexiones de placas CAN.....	79
Ilustración 60 - Resultado de la conexión de todas las placas	80
Ilustración 61 - Envío de tramas en una Raspberry Pi	80
Ilustración 62 - Recepción de las tramas en la Raspberry Pi	80

Lista de Tablas

Tabla 1 - Arquitecturas de comunicación	17
Tabla 2 - Lista de buses de campo	19
Tabla 3 - Características buses de campo	20
Tabla 4 - Capas definidas bus CAN	29
Tabla 5 - Relación velocidad, distancia y tiempo de bit.....	32
Tabla 6 - Tipos de cableado disponibles	37
Tabla 7 - Asignación de pines conector M12.....	40
Tabla 8 - Tabla de modelos de Raspberry Pi [22]	55
Tabla 9 - Tabla de componentes de Raspberry Pi	57
Tabla 10 - Descripción puerto DSI	60
Tabla 11 - Conector CSI.....	61
Tabla 12 - Pines del conector de alimentación.....	62
Tabla 13 - Software Raspberry Pi	66
Tabla 14 - Lista de componentes PCBs	71
Tabla 15 - Presupuesto	81
Tabla 16 – Precio de las Raspberrys Pi y los cargadores	82

1. Introducción

1.1. Contexto y justificación del Trabajo

El propósito principal del desarrollo del presente trabajo es la de mostrar los principios básicos de funcionamiento del protocolo de comunicaciones CAN. Para ello realizaremos un estudio en detalle del protocolo y las diferentes capas que lo componen. Durante el desarrollo de este se tratará de mostrar como un bus que inicialmente fue diseñado para comunicar los diferentes sistemas electrónicos integrados en los vehículos de transporte, ha pasado a ser utilizado para las comunicaciones en multitud de aplicaciones de sectores muy diferentes [1][2].

Debido a todo lo expuesto anteriormente es totalmente recomendable realizar un estudio sobre un bus de comunicación serial distribuida, en nuestro caso el elegido es el bus CAN, esto permitirá conocer las bases del diseño e implementación de uno de los protocolos más utilizados actualmente en la industria [2]. Además, cabe decir que mi interés por este bus aumento debido a la realización de un proyecto anterior en el que tuve la oportunidad de tener un primer contacto con este bus de comunicaciones.

1.2. Objetivos del Trabajo

Los objetivos marcados para la realización del proyecto son los siguientes:

- Se tratará de recopilar toda la información posible del bus de comunicaciones CAN y ponerla a disposición de personas que tengan la intención de desarrollar proyectos con esta tecnología.
- El segundo objetivo es promocionar una tecnología que es desconocida para muchos profesionales, de forma que se pueda fomentar su uso en proyectos en los que no se ha utilizado con anterioridad.

1.3. Enfoque y método seguido

El TFM se ha enfocado como un estudio teórico que permita mejorar los conocimientos sobre el bus de comunicaciones CAN. Para ello se ha realizado una planificación del trabajo marcando los hitos más importantes, para ello se a utilizado una estructura WBS, en donde se realiza la descomposición de forma jerárquica de los entregables del proyecto.

Aunque inicialmente no se tenía pensado la implementación de ningún hardware, según se iba desarrollando el proyecto se veía la necesidad de realizar algún tipo de hardware. El objetivo de la implementación de ese hardware es el de ayudar a entender mejor el funcionamiento del bus de comunicaciones sobre el que se desarrolla el trabajo, esto quiere decir que el hardware no es una finalidad en sí mismo, sino que tiene que verse como una herramienta que permite llegar a los objetivos marcados en el punto anterior.

El Hardware utilizado como ejemplo se ha obtenido de documentación encontrada en diferentes foros y permite la comunicación entre dos Raspberrys Pi mediante bus CAN. Este ejemplo ha permitido comprobar lo fácil y barato que resulta el uso de este bus de comunicaciones, en menos de 15 días se ha puesto en funcionamiento el sistema con un coste de 32€.

El software que permite la comunicación está a disposición de todos los usuarios y solo tiene que descargarse e instalarse en las dos Raspberrys utilizadas para la comunicación. En concreto se denomina can-utils y se trata de un SocketCAN que dispone de controladores CAN de código abierto y que puede ser incorporado al kernel de Linux.

1.4. Planificación del Trabajo

A continuación, se muestra el diagrama WBS que muestra los entregables del proyecto:



Ilustración 1 - Diagrama WBS

En el diagrama de pueden ver las partes más importantes que forman parte del proyecto:

- Hardware: está formado por dos nodos que tienen que ser fabricados, estos permiten la comunicación de las dos Raspberrys.
- Software: con respecto al software no se realiza un desarrollo en sí mismo, sino que se aprovecha el software disponible para establecer la comunicación mediante el bus CAN.
- Documentación: es donde se plasma toda la información obtenida durante la realización del TFM. Se divide en la memoria, presentación y un video

donde se puede visualizar el funcionamiento de las PCB fabricadas. La planificación se muestra en los apartados incorporados a continuación.

1.4.1. Definición TFM

Esta es una de las partes más relevantes del trabajo ya que es donde se realiza la elección del tema sobre el que se va a realizar el mismo. Para ello con ayuda del consultor se realiza la elección del tema.

Una vez elegido el tema se realiza un primer estudio del tema del trabajo, para ello se realiza un resumen, se eligen algunas palabras claves, se marcan los objetivos, la motivación que nos lleva a realizar este trabajo y una primera versión del índice. Además, hay que realizar una primera búsqueda de documentación que nos podría ayudar a la realización del trabajo.

1.4.2. Estado del Arte

Este bloque nos permite plasmar lo encontrado sobre otros proyectos de estudio similares al que estamos desarrollando. Como el trabajo se centra principalmente es un estudio seguimos buscando recursos que nos serán de utilidad en apartados posteriores.

1.4.3. Diseño e implementación

En este bloque es donde se desarrolla todo el estudio sobre el funcionamiento del bus de comunicaciones. Además, se ha realizado la implementación de dos circuitos PCB que permiten comunicarse a dos Raspberrys Pi mediante bus CAN. Una vez terminada este bloque se tiene una primera versión de la memoria, aunque esta esta aun incompleta.

1.4.4. Memoria

En este punto del trabajo se realiza la recopilación de toda la información redactada hasta el momento, para aplicarle el formato correcto y terminar de darle forma al TFM.

1.4.5. Presentación

En este bloque es donde se realiza la defensa o presentación de TFM, para ello se realiza un PowerPoint que puede estar acompañado por un video donde se muestra el desarrollo realizado durante todo el trabajo.

1.4.6. Diagrama de Gantt

En el diagrama se puede ver de forma más detallada la planificación descrita anteriormente. En ella se muestran las acciones desarrolladas en cada uno de los bloques y su duración aproximada.

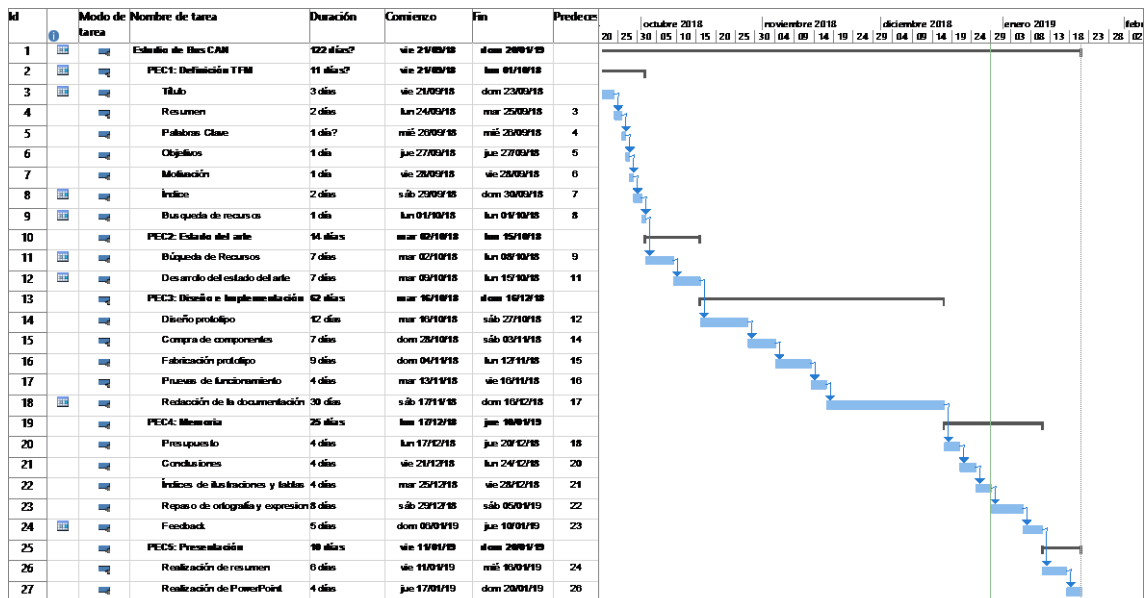


Ilustración 2 - Diagrama de Gantt

1.5. Breve resumen de productos obtenidos

Este TFM queda dividido en dos grandes entregables (documentación, implementación de un ejemplo de nodo CAN). La parte de documentación queda dividida en:

- Memoria
- Presentación

Mientras que la implementación de un ejemplo de nodo CAN se divide en:

- Diseño de PCB. En esta parte se procede a introducir el esquemático que nos permitirá realizar la PCB que se fabricará posteriormente.
- Fabricación. Fabricación de PCB, compra de componentes y montaje de estos.
- Pruebas de funcionamiento. Configuración de Raspberrys Pi, montaje de circuitos y comprobación del funcionamiento correcto de todo el sistema de comunicaciones.

1.6. Breve descripción de los otros capítulos de la memoria

- Capítulo 2. Estudio previo. Descripción general de los conceptos más importantes que ayudaran a entender el resto del trabajo.
- Capítulo 3. Introducción al bus CAN. En este capítulo se describe la historia del bus CAN y sus características principales del mismo.
- Capítulo 4. Niveles OSI de ISO bus CAN. Descripción de las capas del protocolo de comunicaciones CAN a través de los estándares disponibles.

- Capítulo 5. Ejemplo de nodo CAN. Fabricación y montaje de dos placas para llevar a cabo la comunicación de dos Raspberrys Pi mediante bus CAN.
- Capítulo 6. Presupuesto. Realización de un presupuesto que permita conocer el coste de las dos placas montadas.
- Capítulo 7. Conclusiones.

2. Estado del arte

En el presente capítulo se trata de exponer todos los conocimientos adquiridos durante el proceso de búsqueda de bibliografía utilizada para el desarrollo del trabajo final de máster. Centrándome en explicar algunos conceptos que nos permitirán entender el posterior desarrollo que se llevará a cabo en los siguientes puntos.

Se tratará de explicar de forma breve el modelo de capas ISO y como afecta al bus de comunicaciones sobre el que se desarrolla el siguiente estudio. A pesar de mostrar unas breves pinceladas de todas las capas del modelo OSI, tengo que comentar que en nuestro estudio del bus CAN que se llevara a cabo en capítulos posteriores solo se mencionaran 3 de las capas que son las que están definidas para este bus en concreto.

Se darán algunas pinceladas sobre las principales topologías de red existentes o más usadas en la actualidad. Se hará una pequeña introducción a los buses de campo y se enumeraran algunos de ellos para a posteriori dar una explicación más detallada del bus CAN.

2.1. Modelo OSI de 7 capas

El modelo OSI se creó a principios de 1980 por ISO. ISO es una organización para la creación de estándares internacionales que está compuesta por diversas organizaciones nacionales de estandarización. Esta necesidad por la gran proliferación de multitud de tecnologías de comunicaciones diferentes, estas tecnologías en muchas ocasiones eran incompatibles entre sí. El diseño de este modelo sirvió de ayuda para la creación de tecnologías de comunicación que fuesen compatible entre sí.

El modelo creado estaría compuesto por siete capas o niveles por los que tendrían que pasar los datos para pasar de un dispositivo a otro en una red de comunicaciones. En nuestro caso solo se trataría con tres de estos niveles, esto es debido a que los buses de campo solo definen la capa física, la capa de enlace de datos y la capa de aplicación.



Ilustración 3 - Modelo OSI de ISO

Lo que se hace en el modelo OSI es subdividir todo el proceso de comunicaciones en tareas. Estas tareas permiten la gestión del flujo de información (capas 4,3,2, y 1) y otra parte se encarga de dar el soporte necesario a los usuarios (capas 7,6,5) [3].

2.1.1. Capa Física

En esta capa es donde quedan definidas todas las características mecánicas y eléctricas de la comunicación. Se indican el tipo de cableado a utilizar, los conectores y el tratamiento eléctrico de los bits que se envían. En el estudio desarrollado mostraremos los diferentes tipos de cable y conectores disponibles que puede utilizar el bus a estudio. Además, se mostrará de forma detenida el comportamiento eléctrico que utiliza el bus para la transmisión de datos [3][4].

2.1.2. Capa de Enlace de Datos

Esta capa se encargará de recoger los datos del nivel superior y conformar las tramas que se enviarán a través de los elementos definidos en la capa física. La capa de enlace se encargará del direccionamiento, detección y recuperación de errores, además de la gestión del tráfico y el reenvío de las tramas que son erróneas. Esta capa a su vez queda dividida en dos subcapas, llamadas MAC y LLC [3] [4].

2.1.3. Capa de Red

Esta capa se encarga del funcionamiento de la red, su misión principal es la de controlar la ruta que los datos entre los distintos elementos que componen la comunicación. Será la encargada de determinar la ruta que deben de seguir los datos para llegar al destino deseado, es lo que se conoce como encaminamiento (Routing). Además, la capa de red se encarga del control de la congestión y la traducción de los nombres lógicos en direcciones físicas [3] [4].

2.1.4. Capa de Transporte

La capa de transporte es la encargada de realizar el enlace entre los diferentes nodos de forma fiable, también se ocupa de la interpretación de las direcciones, además si los paquetes que se desean enviar son muy largos se responsabiliza de su fraccionamiento a todo esto, hay que añadir que también puede generar varias conexiones que permitiría el envío de los mensajes a mayor velocidad. La capa de transporte se hace cargo del establecimiento del enlace, la posterior transmisión de los datos y por último de la disolución del enlace. El usuario puede determinar la calidad del servicio (QoS). Algunos de los parámetros que se pueden variar para obtener la calidad del servicio pretendida son, la velocidad de transferencia y la tasa de errores [3] [4].

2.1.5. Capa de Sesión

Como su propio nombre indica en esta capa se establecerá la sesión de comunicación entre dos capas de aplicación. Se encargaría de configurar como se establece la comunicación, algunos de estos modos de comunicación serían Simplex, Semiduplex (half-duplex), Dúplex (full-duplex) [3] [4].

2.1.6. Capa de Presentación

Esta capa se ocupa de la conexión entre sistemas que utilizan diferentes lenguajes de comunicación. Se puede decir que se encargaría de traducir los datos que se desean enviar a un lenguaje que sería el acordado para la comunicación. Para entendernos mejor los datos vendrían de la capa superior (Capa de Aplicación) y estos serían transformados a un lenguaje que pueda ser enviado sin generar problemas de compatibilidad con los diferentes sistemas a los que se quiere enviar la información. También se responsabilizaría de realizar el proceso inverso, cogería los datos recibidos y los convertiría al lenguaje que utiliza la aplicación que está solicitando la transmisión y recepción [3] [4].

2.1.7. Capa de Aplicación

Esta capa es la encargada de proporcionar la interfaz entre las aplicaciones que se utilizan para la comunicación y la propia red por la cual se transmiten los datos. Podemos decir que es la encargada del intercambio de los datos de los programas que son ejecutados en los nodos de origen y destino. Como funciones principales podemos enumerar la sincronización de las aplicaciones, marcado de los procedimientos que permiten recuperar los datos producidos por errores en la comunicación [3] [4].

2.2. Modos de transmisión

Me parece de interés la incorporación de una breve descripción de los modos de transmisión, debido a que fueron mencionados en el punto 2.1.5. Capa de Sesión [5].

2.2.1. Simplex

En el modo simplex la comunicación solo se realiza en un sentido, desde el terminal transmisor al receptor [5].

2.2.2. Semidúplex (half-duplex)

Semidúplex permite la transmisión en los dos sentidos, pero esta transmisión solo puede producirse de forma alterna. En este caso solo uno de los dos terminales puede transmitir en cada instante [5].

2.2.3. Dúplex (full-duplex)

Dúplex en cambio permite la transmisión de los terminales en ambos sentidos de forma simultánea. Evidentemente esta es la mejor forma para llevarse a cabo las comunicaciones entre dos terminales [5].

2.3. Sistemas distribuidos de comunicaciones de tiempo real

Entre los sistemas de tiempo real no solo se da importancia al resultado lógico que se obtiene de una medición, sino que también se da mucha importancia al momento en el que se obtienen esos resultados para poder asegurar su correcto funcionamiento. Esto da como resultado que no solo se debe tener en cuenta la salida obtenida a partir de la entrada, sino que además se deben tener en cuenta ciertas restricciones temporales que el sistema debe cumplir.

Este tipo de sistemas se compone de un conjunto de nodos que se conectan entre si a través de una red de comunicaciones de tiempo real. Estos deben cumplir con diferentes requerimientos que son demandados por otros sistemas. Estos requerimientos quedan clasificados de la siguiente forma: fiabilidad, exactitud, flexibilidad y estructural.

La fiabilidad es una característica que define el grado de seguridad que se produce durante el proceso de comunicación. La misión principal es la detección de errores, además del control de las posibles averías y fallos que se pueden producir durante la transmisión de la información. Lo que se busca son sistemas de comunicación lo más fiables posibles, de esta forma se reducen al máximo o totalmente los errores durante la comunicación.

La exactitud trata de obtener un jitter mínimo además de una latencia lo más corta posible, esta es precisamente la característica más relevante de los sistemas en tiempo real con respecto al resto. Para poder tener controlados los parámetros anteriores se tiene que disponer de una base de tiempos común para

todos los elementos que componen el sistema. Como ejemplo de esto se propone hacer la lectura de forma dinámica de 100 células de carga, con los pesos obtenidos se pretende realizar una gráfica con todos los pesos obtenidos en instantes concretos. Para conseguir que esto sea así se deben de realizar las lecturas de los pesos de las 100 células de carga en instantes concretos, porque si no se realizara así la gráfica obtenida no sería realmente la deseada.

Un protocolo de comunicaciones que funciona en tiempo real debe de ser bastante flexible, debe de ser capaz de soportar cambios en la red sin que esto suponga tener que hacer modificaciones en el software. Esto quiere decir que no debería de haber problemas para incorporar o eliminar nodos de la red, solo deberían de estar limitados por parámetros como el ancho de banda o las limitaciones que se generarán debido al aumento del tráfico. En el bus CAN se pueden incorporar los nodos que se deseen siempre que no se supere el máximo que nos permite el protocolo, al aumentar el número de nodos lo que ocurre es que el ancho de banda se reparte entre todos los nodos y por tanto aparecerán restricciones temporales. Siguiendo el mismo ejemplo visto anteriormente, si disponemos de un bus CAN de 100 canales que dispone de una velocidad de muestreo máxima, si ampliamos el número de canales a 120 cada canal verá reducido en número de muestras por segundo que podrá transmitir.

En lo que se refiere a los requerimientos estructurales no solo se tendrá en cuenta las limitaciones técnicas sino también evidentemente se deben tener en cuenta las limitaciones económicas. Para resumir, se deben de asumir los sistemas más robustos posibles, pero con el menor costo posible. La elección del sistema cobra mucha relevancia porque de ello depende los resultados obtenidos [6].

2.4. Topologías

Los tipos de topología de red muestra los diferentes tipos de interconexiones existentes entre los diferentes nodos de red. Hace referencia de una forma geométrica o lógica el cómo se distribuyen las estaciones de trabajo y cada uno de los medios que las conectan [7].

2.4.1. Anillo

Las redes con topología de anillo conectan todos los nodos entre sí mediante un cable formando un círculo entre los mismos. En este tipo de topología la información viaja por la red en un solo sentido del círculo, tiene como desventaja que si un nodo de la red deja de funcionar la red dejará de funcionar, esto la convierte en una topología poco eficaz [7].

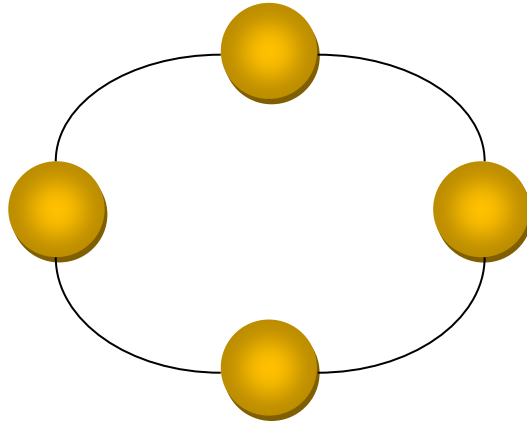


Ilustración 4 - Topología en anillo

2.4.2. Topología de anillo doble

Es una topología similar a la anterior, pero en lugar de disponer de un solo anillo dispone de dos, de forma que cada uno de los equipos estará conectado con el resto a través de los dos anillos. La diferencia con la topología anterior es que dispondrá de un anillo redundante para conectar los mismos dispositivos [7].

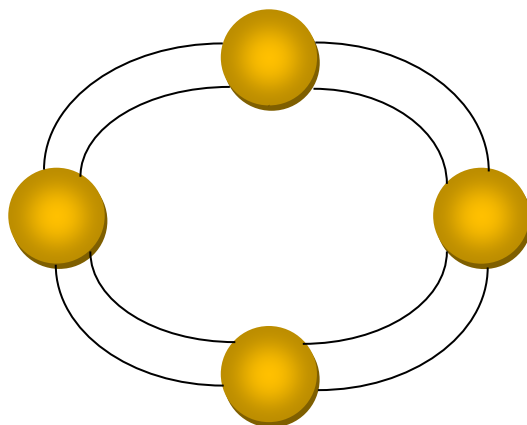


Ilustración 5 - Topología anillo doble

2.4.3. Árbol

La topología en árbol como su propio nombre indica conecta los diferentes nodos en forma de árbol. Guardaría un parecido con una serie de redes en estrella interconectadas, pero con la diferencia de que no tiene un nodo central. La principal ventaja se encuentra en que la falla de un nodo no implicaría la interrupción de las comunicaciones. Este tipo de topología toma características de las topologías bus y estrella. Es muy utilizada porque permite realizar ampliaciones en la red de forma sencilla [7].

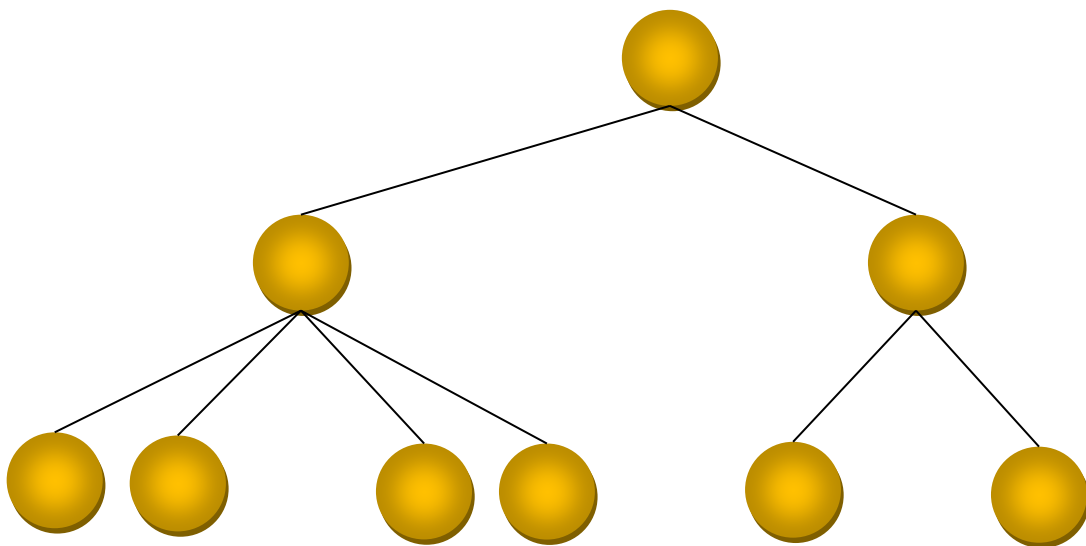


Ilustración 6 - Topología en árbol

2.4.4. Bus

Todos los componentes que conforman la red estarán conectados en serie a un cable central, este será el encargado de distribuir toda la información a los nodos conectados. Esta topología en concreto es la utilizada normalmente por el bus de comunicaciones que se encuentra en estudio en el presente trabajo final de Máster que se está desarrollando [7].

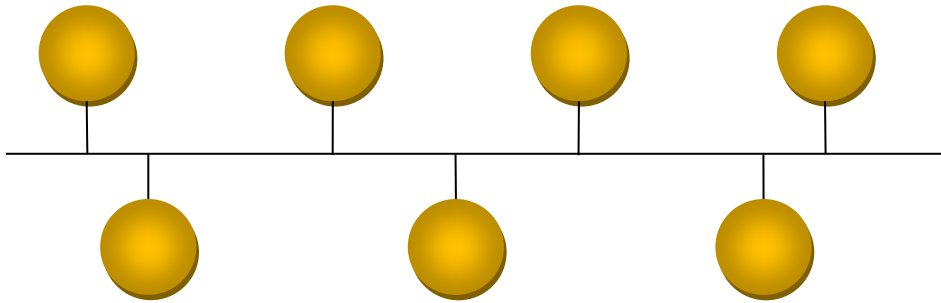


Ilustración 7 - Topología en bus

2.2.5. Estrella

La topología en estrella dispone de un nodo central desde el que se distribuye toda la información hacia el resto de los nodos que forman la red o los puntos de destino de la información. El uso de esta topología se encuentra muy extendida y su principal ventaja es que si un nodo falla la red continuaría trabajando, aunque esto último siempre dependerá del modo de funcionamiento del host central. La desventaja principal es que si el nodo central se estropea el resto de la red dejará de funcionar [7].

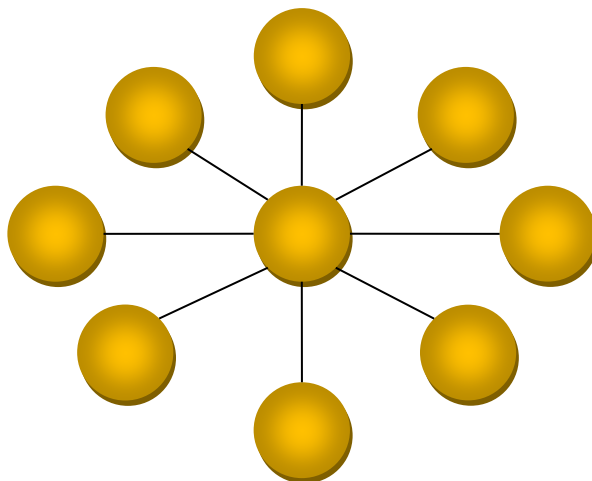


Ilustración 8 - Topología en estrella

2.4.6. Malla

Los nodos en la topología de malla se conectan todos con todos por medio de conexiones independientes. De esta manera se obtienen multitud de caminos redundantes de modo que si se produce un fallo en una de las conexiones el tráfico se puede desviar por otro de los caminos disponibles. No dispone de un nodo central s diferencia de otras tipologías. Debido a la gran cantidad de caminos disponibles, podemos decir que la perdida de datos durante las comunicaciones se hace tremendamente difícil. En cambio, su mayor desventaja viene dada por el altísimo coste de instalación [7].

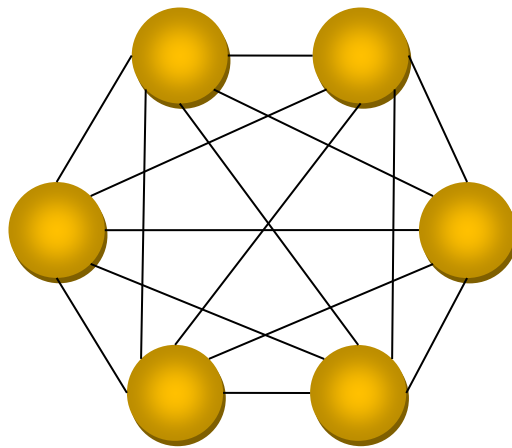


Ilustración 9 - Topología en estrella

2.4.7. Híbrida

Las redes híbridas pueden estar compuestas por una mezcla de las diferentes topologías que se han visto con anterioridad. De esta manera podemos obtener una red que se adapte perfectamente a las necesidades de despliegue. Como recordatorio podemos decir que dependiendo de las necesidades de la red de comunicaciones se podrán realizar una mezcla de las diferentes tipologías, para obtener una solución de comunicaciones idónea [7].

2.5. Buses actuadores y sensores

Dependiendo de la bibliografía a la que se tenga acceso, el bus can puede quedar clasificado o como un bus de actuadores y sensores o dentro de los buses de campo y dispositivos, esto se puede deber a que hay buses de campo como CANopen y DeviceNET, estos buses de campo están basados en la tecnología CAN para su funcionamiento. Los siguientes apartados se desarrollaron para facilitar la comprensión del funcionamiento del bus CAN, que se llevara a cabo de forma más pormenorizada en capítulos posteriores [8].

2.6 Buses de campo y dispositivos

Los buses de campo se pueden definir como un enlace de comunicaciones digital, bidireccional y multipunto entre dispositivos inteligentes de control y medida. Los buses de campo se utilizan hoy en día en multitud de aplicaciones, enumeraremos algunas de ellas: automatización industrial, construcción de maquinaria, máquinas de rayos x, ascensores, aplicaciones en sectores como la automoción, ferrocarril y aeronáutica entre otros. Durante los siguientes puntos se podrá ver de forma más detenida su funcionamiento [8].

2.6.1. Arquitectura de nodo

En este punto se explicará cómo se compone de forma generalizada la arquitectura de un nodo que se podría encontrar en un bus de campo, más centradas en el bus CAN, aunque se podría extender para la mayoría de los buses que se encuentran en el mercado.

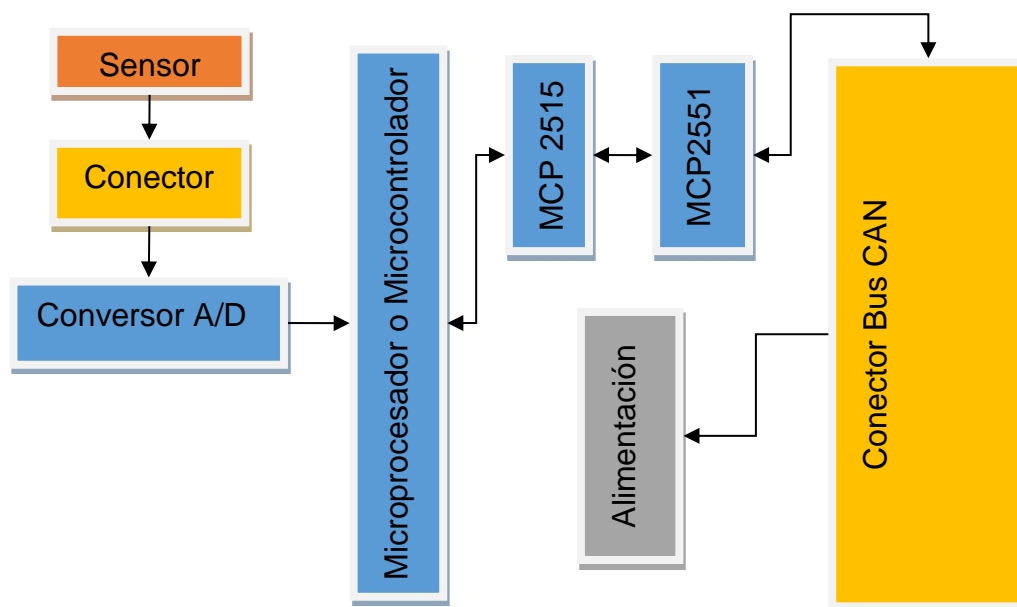


Ilustración 10 - Arquitectura de un nodo

Gracias a la ilustración se puede tener una idea de los elementos que se podrían disponer en un nodo de comunicaciones de un bus de campo, evidentemente se podrían tener algunas diferencias de configuración. Por ejemplo, en este caso la alimentación se traslada a través del propio cableado del bus, pero hay también configuraciones en que la alimentación no se realiza por el mismo bus si no que se obtiene de forma independiente.

2.6.2. Características y ámbitos de aplicación

Una de las características que tienen muchos de estos buses es la reducción o simplificación del cableado que necesitarían las aplicaciones. Antes de la

aparición de estos buses la información que proporcionaban los sensores se trasladaba con un estándar analógico de corriente conocido como 4-20 mA de hecho, si tomamos el ejemplo expuesto con anterioridad que hace referencia a un sistema de adquisición para 100 células de carga, una de las soluciones propuesta incorporaba este método como uno de los posibles a utilizar. Evidentemente se desestimó debido a que cablear 100 células de carga con este método requería de una cantidad inmensa de cableado, además en algunos casos requería de un tratamiento intermedio de la señal para que el dato pudiera soportar sin pérdidas las grandes distancias a cubrir.

La introducción de los buses de campo simplifico en gran medida las comunicaciones, además de reducir el tiempo de diseño, instalación configuración y mantenimiento de los sistemas. Esto ha permitido una reducción enorme de los costes y un gran aumento de la flexibilidad, debido a que estos buses facilitan la eliminación y incorporación de nuevos dispositivos a la red. Otra de las mejoras obtenidas es la de facilitar el mantenimiento de los equipos, puesto que las redes conformadas permiten conocer el estado de los componentes que forman la red. Volviendo al ejemplo anterior cada vez que se arranca el sistema este realiza una prueba sobre todos los nodos conectados al sistema y sobre cada una de las células de carga conectadas al mismo, si alguna célula esta estropeada o un nodo está en mal estado el sistema lo detecta de forma inmediata, esta detección permite aislar los nodos o las células en mal estado y seguir funcionando con normalidad hasta su reparación. Estas redes permiten comunicaciones bidireccionales, además del acceso a los datos de forma remota si se desea. Por otro lado, la fiabilidad es muy elevada y permite llevar a cabo estrategias de control más avanzadas, que no serían posibles con los sistemas que se disponían hasta la aparición de los buses de campo.

Los buses de campo tienen la capacidad de trabajar en condiciones muy adversas, son capaces de soportar radiaciones electromagnéticas, ruido eléctrico, fuertes vibraciones y atmósferas de trabajo muy agresivas entre otros. Si tenemos en cuenta el ejemplo de bus CAN con las 100 células de carga, podemos decir que las condiciones de trabajo en las que se lleva a cabo el pesaje son extremas, debido a las fuertes vibraciones por el trabajo de la maquinaria pesada y con unas condiciones atmosféricas tremendamente agresivas producidas por los abonos que se han de pesar.

El campo de aplicación de este tipo de buses en la industria es inmenso, ya que permite la comunicación entre máquinas o entre dispositivos que pueden encontrarse en el interior de las propias máquinas. Son muy utilizados en aplicaciones en las que se requieran de mucha fiabilidad y seguridad en la transmisión de los datos. También se utilizan en muchas aplicaciones en las que se requiere llevar la alimentación a los nodos mediante el cableado del propio bus de comunicaciones. Como ejemplo podemos decir que este tipo de buses se pueden utilizar para aplicaciones domóticas, en la industria química, en el control de servicios públicos, como el control del tráfico y monitorización de autopistas, es muy utilizados en los sistemas de control de los automóviles. Estos sistemas se encargarían de controlar todos los elementos que incorporaría un automóvil, luces, motor, frenos, etc. En estos últimos se podrían disponer de varios nodos encargados de comunicar diferentes elementos del coche [9].

2.6.3. Arquitecturas de comunicación

En la industria se pueden diferenciar tres modelos distintos de comunicación:

- a) Modelo de publicación-suscripción
- b) Modelo de producto-consumidor.
- c) Modelo de publicación-subscripción.

	Modelo de publicación-suscripción	Modelo productor-consumidor	Modelo cliente-servidor
Tipo de comunicación	Multicast	Broadcast	Peer-To-Peer
Estilo de comunicación	Sin conexión explícita	Sin conexión explícita	Orientado a la conexión
Relación maestro-esclavo	Varios maestros	Varios maestros	Uno o varios maestros
Servicios de comunicaciones	Sin confirmar, con confirmación	Sin confirmar, con confirmación	Confirmados, sin confirmar, con confirmación
Clases de aplicaciones	Cambios de estado y notificación de eventos	Notificación de eventos, alarmas, eventos, sincronización	Transferencia de parámetros, comunicación cíclica

Tabla 1 - Arquitecturas de comunicación

El modelo de comunicación más extendido es el de cliente-servidor. En este modelo disponemos de un cliente que solicita o pide un servicio y un servidor que es el encargado de responder a las peticiones realizadas por los clientes. Este es por lo tanto un modelo punto a punto. Este tiene más utilidad para la transmisión de información de estado. El servidor solo responderá a peticiones del cliente. La interacción entre el cliente y el servidor se realiza generalmente mediante cuatro pasos. Se produce una indication esto es un evento que se produce en el cliente que permite indicar la recepción de una petición request. Esto genera una respuesta mediante response que se comunicará en la parte del cliente mediante confirmation. Este es una de las arquitecturas de comunicación más utilizadas en la actualidad, debido a esto es lo que nos ha llevado a realizar una breve descripción sobre el mismo. El modelo cliente-servidor tiene grandes problemas de consistencia en sistemas distribuidos. Es decir, si se desea enviar el mismo mensaje a dos o más dispositivos, se tendrá que generar una relación de cliente servidor con cada uno de los clientes a los que se desea enviar. Producto de esta forma de funcionamiento el sistema enviara los mensajes por separado, primero se envía el mensaje u uno de los clientes y posteriormente al otro. Esto hace que el valor que se les envía no sea el mismo ni se realice en el mismo instante de tiempo.

Los sistemas distribuidos que utilizan buses de campo solucionan el problema con el modelo producto/consumidor que podemos ver en la tabla dada anteriormente, este es un modelo multipunto. En este modelo los consumidores se subscriben al productor para poder recibir la información que precisan. En este caso los mensajes serán accesibles por todos los consumidores con el mismo valor y en el mismo instante de tiempo. Buses de campo que utilizan este modelo son WorldFIP, CAN, EIBus, ControlNet, Swiftnet y FF, aunque también

es verdad que muchos de los buses de campo pueden utilizar de forma simultánea diferentes modelos dependiendo del tipo de servicio requerido.

El modelo publicación-subscripción es un modelo en el que los proveedores de información no disponen de un enlace directo con los consumidores de dicha información. En este tipo de modelos el publicador de la información no necesita saber quiénes utilizan la información y el suscriptor tampoco necesita saber quién la pública. Este estilo es completamente diferente al utilizado en los sistemas punto a punto, en la que los servidores que envían los mensajes necesitan conocer el destinatario de estos.

Los buses de campo incorporan mecanismos de acceso al medio en el nivel de enlace que les permiten gestionar el tráfico dentro del bus. Un aspecto relevante de esto es la planificación de los mensajes, que permite satisfacer las exigencias temporales que tiene un sistema de control, así como la flexibilidad necesaria para adaptarse de forma flexible a los cambios [9].

2.6.4. Buses de campo estandarizados

Hoy en día se disponen de infinidad de buses de campo, tanto propietarios como estandarizados. En la siguiente tabla se enumeran algunos de estos buses de campo con la relación de los diferentes estándares IEC y CENELEC, así como el nombre por el que son más conocidos en la industria.

Nombre Comercial	CENELEC	IEC
P-NET	EN 50170-1(07/1996)	IS 61158 type 3
Profibus	EN 50170-2(07/1996)	IS 61158 type 1/3/10
WorldFIP	EN 50170-3(07/1996)	IS 61158 type 1/7
Foundation Fieldbus	EN 50170-A1(04/2000)	IS 61158 type 1/9
Profibus-PA	EN 50170-A2(04/2000)	IS 61158 type 1/3
ControlNet	EN 50170-A3(04/2000)	IS 61158 type 2
Interbus	EN 50254-2(10/1998)	IS 61158 type 8
Profibus-DP	EN 50254-3(10/1998)	IS 61158 type 3
WorldFip (FIPIO)	EN 50254-4(10/1998)	IS 61158 type 7
DeviceNet	EN 50325-2(01/2000)	IS 62026-3(2000)
SDS	EN 50325-3(04/2000)	IS 62026-5(2000)
CANOpen	EN 50325-4(07/2002)	

AS-Interface	EN 50295-2	IS 6226-2(2000)
---------------------	------------	-----------------

Tabla 2 - Lista de buses de campo

En la siguiente tabla se muestran los principales buses de campo con sus características más relevantes:

Bus de campo	Topología	Medio Físico	Velocidad	Distancia segmento	Nodos segmento	Acceso al bus-Estándar
ARCNET	Bus Estrella	Par trenzado Fibra óptica Coaxial	2.5 Mbps	122 m	255	Paso testigo-ANSI 878
AS-I	Bus Árbol Estrella	Cable de dos hilos	167 kbps	100 m	32	Sondeo principal-subordinadas
Bitbus	Bus	Par trenzado Fibra óptica	62.5/375 kbps 1.5 Mbps	1200 m/300 m	29	Sondeo principal-subordinadas-IEEE 1118
CAN	Bus	Par trenzado	50 kbps a 1 Mbps	1000 m a 40 m	127	CSMA/CD con arbitraje de bit-ISO 11898/11519
CC-Link	Bus	Par trenzado	156 bps a 10 Mbps	1200 m a 100 m	64	Sondeo principal-subordinadas
ControlNet	Bus Árbol Estrella	Coaxial Fibra óptica	5 Mbps 5 Mbps	1000 m 3000 m	48	CTDMA-Basado en CAN
DeviceNet	Bus	Par trenzado	125-250-500 kbps	500-250-100 m	64	CSMA/CD-ISO 11898 y 11519 (basado en CAN)
EIB	Libre	Par trenzado Infrarrojos Red eléctrica radio				CSMA/CA-CENELEC TC105
Filbus		Par trenzado	375 kbps	1200 m	32	
Firewire	Bus Árbol	Cable de 6 hilos Fibra óptica	100 Mbps 200-400 Mbps	700 m	63	Ranuras de tiempos-IEEE1394
Foundation Fieldbus		Par trenzado		1900 m	32	Paso testigo-ISA SP50/IEC TC65
HART	Bus	Par trenzado apantallado	1200 bps	3000 m	30	Sondeo principal-subordinadas-IEC TC65
Interbus-S	Anillo	Par trenzado	500 kbps	400 m	256	Paso testigo-DIN E19258
J1939	Bus	Par trenzado apantallado	250 kbps			CSMA/CD con arbitraje de bit-SAE J1939 (Basado en CAN)
LonWorks	Bus Anillo Libre	Par trenzado Fibra óptica Red eléctrica Coaxial Radio Infrarrojos	78 kbps a 1.25 Mbps	130 a 2700 m	64	CSMA/CA EIA-709
M3S	Bus	Cable de 6 hilos				CSMA/CD con arbitraje de bit-Basado en CAN

M-Bus	Bus	Cable de 2 hilos	300 a 9600 bps	1000 m	250	Centralizado-IEC 870; EN1435-3
Measurement Bus	Bus	Cable de 4 hilos	110 bps a 1 Mbps	500 m	31	Sondeo principal-subordinadas-DIN 66348
Modbus	Bus	Par trenzado	300 bps a 19.2 kbps	1000 m	248	Sondeo principal-subordinadas
NBIP	Bus	Cable de 2 hilos				Sondeo principal-subordinadas
P-NET	Anillo	Par trenzado apantallado	76.8 kbps	1200 m	125	Paso testigo Sondeo principal-subordinadas-CENELEC EN50170 Vol.1; DSF 21906; DK 502058; DK 502066
PROFIBUS	Bus	Par trenzado apantallado	9.6 kbps 19.2 kbps 93.75 kbps 187.5 kbps 500 kbps	1200 m 1200 m 1200m 600 m 200 m	32	Paso testigo Sondeo principal-subordinadas-CENELEC EN50170 Vol.2; DIN STD 19245
SDS	Bus	Cable de 4 hilos	125 kbps 250 kbps 500 kbps 1 Mbps	500 m	64	CSMA-ISO 11989 (basado en CAN)
SENSOPLEX	Bus	Coaxial			120	Sondeo principal-subordinadas
SERCOS	Anillo	Fibra óptica	2 Mbps		255	Anillo ranurado-IEC 61491
SERIPLEX	Bus	Cable de 4 hilos apantallado	98 kbps	1500 m	300	Sondeo principal-subordinadas
WorldFIP	Bus	Par trenzado apantallado Fibra óptica	31.25 kbps 1 Mbps 2.5 Mbps 5 Mbps	1000 m 750 m 500 m	32	Centralizado (árbitro de bus)-CENELEC EN 50170 Vol. 3; NFC 46 601 a 607

Tabla 3 - Características buses de campo

En los puntos siguientes se tratarán algunos de los buses más utilizados en la actualidad [9].

2.7. El bus de campo PROFIBUS

PROFIBUS es uno de los buses de campo más utilizados actualmente en la industria, especialmente en aplicaciones de control y automatización. Como muchos otros buses se desarrolló en Alemania en el año 1987, su diseño nació de mano de un grupo de empresas y usuarios con el respaldo del gobierno de Alemania. La norma alemana encargada de hacer una descripción del protocolo PROFIBUS es la DIN 19245, años después esta se incorporó en la norma europea CENELEC EN 50170.

El protocolo PROFIBUS se compone de tres perfiles, cada uno de ellos se orientará a diferentes tipos de aplicaciones: PROFIBUS-FMS, PROFIBUS-DP y PROFIBUS-PA. PROFIBUS-PA es el perfil que se utiliza para la automatización de sistemas. La norma IEC 1158-2 es la encargada de definir el perfil PA y regula como se alimentan las estaciones mediante el propio bus de comunicaciones y la seguridad intrínseca de la misma. PROFIBUS-DP se trata de un perfil optimizado para sistemas de automatización distribuidos, en los que los tiempos de comunicación son tremendamente críticos, PROFIBUS-FMS este es una solución para tareas de comunicación industriales a nivel superior y en el nivel de dispositivos de campo.

PROFIBUS es un bus serie que permite interconectar dispositivos de campo. El sistema dispone de dos tipos de estaciones, una que sería la principal (maestra o activa) y dispondría de otra llamada subordinada (esclava o pasiva). Las estaciones maestro tienen la capacidad de controlar el bus de comunicaciones y pueden enviar mensajes cuando tienen acceso al mismo. Las estaciones esclavas en cambio no tienen acceso al bus y solo podrían contestar a peticiones realizadas por las estaciones principales. Evidentemente estas estaciones son más sencillas de realizar que las estaciones maestro.

PROFIBUS solo define tres niveles de los siete que dispone el modelo OSI: Capa Física, Capa de Enlace y Capa de Aplicación. Esto es muy habitual en los buses de campo. Además, se disponen de otras capas para la gestión de la red conocidas como FMA. El medio físico utilizado para la transmisión de la información es un hilo de cobre. Se trataría de un cable con un par de hilos trenzados y apantallados, aunque si las condiciones lo permiten se podrían utilizar cables sin apantallar, esto último es poco habitual en industria porque suelen ser espacios con muchas interferencias electromagnéticas. Para conectar los diferentes elementos al bus de comunicaciones se utiliza un conector tipo sub-D de 9 pines. PROFIBUS también permite las comunicaciones mediante cable de fibra óptica, esto permite el aumento de las distancias de trabajo y la protección frente a interferencias electromagnéticas.

Dependiendo del tipo de aplicación se pueden tener hasta un máximo de 127 estaciones conectadas al bus de comunicaciones, pero si la aplicación requiere de grandes velocidades de respuesta es recomendable limitar este número de estaciones conectadas. A pesar de ello hay que decir que PROFIBUS puede alcanzar velocidades de hasta 12 Mbps, esta sería una velocidad bastante elevada en comparación con otros buses de campo disponibles en el mercado. Es muy recomendable su uso en aplicaciones que requieran de comunicaciones en tiempo real [9].

2.8. El bus de campo WorldFIP

WorldFIP es un bus de campo desarrollado en Francia y muy utilizado en la industria. Hay un número muy amplio de empresas dedicadas a la fabricación de dispositivos y equipamientos que utilizan este bus para las comunicaciones de sus equipamientos desarrollados. Este protocolo es muy utilizado por empresas como: Cegeleg, Telemecanique, Efisysteme, Gespac, etc. CENELEC describe este protocolo en su norma EN 50170 que fue desarrollada en 1996.

Este bus de campo permite establecer las comunicaciones entre los niveles de sensores/actuadores y el nivel de las unidades de proceso (PLCs, controladores, etc.) en los sistemas de automatización. En el desarrollo de este bus como en otros muchos el aspecto económico ha jugado un papel muy relevante, la reducción de costes mediante la simplificación del cableado más el ahorro que se genera durante el proceso de diseño e instalación han jugado un papel muy relevante para su proliferación. También se han tenido muy en cuenta la reducción de costes de mantenimiento y ampliación de las instalaciones, la simplificación del cableado mediante las comunicaciones a través del bus en lugar de las comunicaciones punto a punto, el poder garantizar unas comunicaciones en tiempo real con unos tiempos de respuesta adecuados y disponer de unas comunicaciones seguras con sistemas de detección de errores también son un potente aliciente.

WorldFIP es un bus que puede trabajar con diferentes tipos de arquitecturas. La difusión de mensajes mediante broadcast ayuda a mantener la consistencia espacial y temporal de los datos para todos los elementos que se encuentran conectados al bus. El protocolo de WorldFIP al igual que en otros buses de campo solo define tres niveles de los siete que encontramos en el modelo OSI. Nuevamente se definen la Capa Física, la Capa de Enlace y por último la Capa de Aplicación. Pero si define una función de gestión de red que se puede comunicar con el resto de los niveles definidos. El medio utilizado para la comunicación es un cable de par trenzado apantallado, también admite la comunicación mediante fibra óptica. Además del cable en un bus del tipo WorldFIP se pueden encontrar los siguientes componentes:

- JB: permite un máximo de dos accesos para derivaciones.
- TAP: se trata de un punto de conexión al bus donde irá instalado un dispositivo de campo.
- REP: se trata de un repetidor que amplifica la señal en un tramo, esto permite disponer un bus de mayor longitud.
- DB: permite unir varios segmentos terminales al cable principal del bus.
- DS: nodos o abonados que se pueden desconectar localmente.
- NDS: nodos o abonados que no se pueden desconectar localmente.

El bus puede ser dividido en segmentos que se irán conectando entre si mediante repetidores, cada segmento dispondrá de un máximo de 32 dispositivos, y en toda la red no se pueden superar un máximo de 256 dispositivos conectados. El número de repetidores entre dos nodos no será superior a tres. Una caja DB permitirá la conexión de un máximo de 8 estaciones en un único punto de conexión.

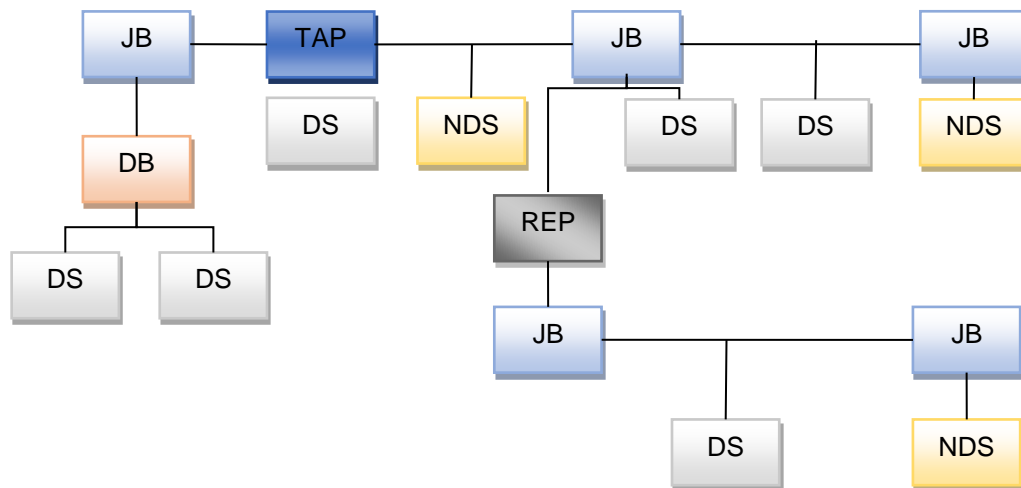


Ilustración 11 - Ejemplo bus WorldFIP [10]

El protocolo WorldFIP define tres posibles velocidades para la transmisión de datos sobre el par de cobre:

- 31,25 kbps. Para aplicaciones especiales.
- 1 Mbps. Velocidad estándar.
- 2,5 Mbps. Para aplicaciones especiales.

Para fibra óptica:

- 5 Mbps.

Dependiendo de la velocidad de funcionamiento del bus se definen las distancias máximas disponibles entre dos estaciones cualesquiera que se encuentran en la red. Esto siempre es sin la colocación de repetidores y como medio físico el cable de cobre de pares trenzados y apantallados. Las distancias máximas disponibles son las siguientes:

- 1900 metros a 31,25 kbps.
- 750 metros a 1 Mbps.
- 500 metros a 2,5 Mbps.

Las distancias anteriores pueden ampliarse con la utilización de repetidores, recordar que no se permiten más de tres repetidores entre dos estaciones.

Las tramas de bits transmitidas utilizan codificación Manchester. Este código permite la transmisión simultánea de datos y el reloj para poder sincronizar las

comunicaciones, recordar que estamos ante un bus de comunicaciones síncrono. Todas las tramas de este protocolo tienen una secuencia de inicio de trama, un campo de control y datos y una secuencia de fin de trama. Este protocolo tiene como base para su funcionamiento dos elementos clave: variables y mensajes. La preferencia en la transmisión la tienen los valores sobre los mensajes. Esto es de gran relevancia en aplicaciones de tiempo real donde la consulta de ciertos valores es crítica para el funcionamiento seguro del sistema.

Como conclusión podemos decir que este protocolo es altamente eficiente pudiendo transmitir información en tiempo real, lo que le permite ser de gran utilidad en su uso a nivel de sensores y de las unidades de proceso en aplicaciones que se utilizan en control y automatización industrial. Otra cosa tremendamente interesante es que es un protocolo transparente a Internet. Debido a esta característica en el año 2000 se tenía la intención de convertirlo en un protocolo de bus de campo para internet [9] [10].

2.9. El bus de comunicaciones CAN

El bus de comunicaciones CAN fue diseñado por la empresa Robert Bosch GmbH en los años 80, fue creado para facilitar las comunicaciones entre las diferentes unidades electrónicas disponibles en el interior del automóvil, ABS, frenos, inyección de combustible, etc. Pero con el paso de los años este bus se fue extendiendo a otros campos de aplicación diferentes para los que fue ideado. Posteriormente aparecieron además otros buses de campo que basan su funcionamiento en CAN, como por ejemplo CANopen y DeviceNet.

Este bus se utiliza en el automóvil como un bus de comunicaciones de lo que se conoce como sistemas distribuidos embebidos, es decir que permite las comunicaciones entre las diferentes unidades de control electrónicas que están empotradas en los sistemas que controlan. Dentro de los vehículos pueden existir varias redes CAN, aunque normalmente son 2 las redes que pueden encontrarse en el interior de los vehículos. Se tendrá una red de alta velocidad que se encargará de comunicar en tiempo real elementos como la inyección de combustible, ABS, frenado del vehículo, etc. Mientras que la red de baja velocidad se encargará de conectar los elementos menos relevantes como los elevadores, sistemas de climatización, etc. En el año 1998 se llegaron a vender 97 millones de controladores CAN, esto da una idea de la proliferación del uso de este bus. Estos datos de ventas nos hablan de forma clara de las importantes ventajas que presenta este bus de comunicaciones, entre estas nos encontramos el bajo coste, fiabilidad y su capacidad de trabajar en entornos muy duros. Esto sumado a la robustez, facilidad de uso e instalación, capacidad de trabajar en tiempo real, permitieron que este bus se utilizara no solo para lo que inicialmente fue diseñado, sino que también empezara a utilizarse en aplicaciones industriales. Dando soporte también a otros buses de campo como, CANopen, DeviceNet, SDS, CAN-Kingdom, etc.

Además de en la industria este bus empezó a utilizarse en aplicaciones de automatización naval, en redes de edificios inteligentes, ascensores, sistemas

de aire acondicionado, sistemas ferroviarios, aviación. En el campo de la medicina este bus se ha utilizado en máquinas de rayos X, scanner, etc.

La aparición de placas de bajo coste que permiten el desarrollo de prototipos de forma rápida y barata también ha ayudado a la proliferación del bus CAN. El bus está muy bien diseñado para la conexión de dispositivos de entrada/salida, sensores y actuadores. Muy importante es el buen funcionamiento que presenta el bus en sistemas que requieren comunicaciones en tiempo real, además tomo mucha relevancia el sofisticado sistema que le permite detectar errores y solventarlos.

Se puede decir que nos encontramos con uno de los mejores buses de comunicación para aplicaciones en tiempo real. Es uno de los buses de mejor relación prestaciones/coste, dispone de un grado de robustez altísimo. Debido a esta y a otras consideraciones este bus ha incrementado su presencia en multitud de sectores que se ha visto incrementado en el tiempo gracias a los buses de campo basados CANopen y DeviceNet. Debido a esto se ha elegido este bus para realizar un estudio más profundo que se desarrollara en los siguientes apartados del trabajo [9].

3. Introducción al bus CAN

3.1. Historia

El sistema de bus serie CAN, nació en febrero de 1986 de la mano de la empresa Robert Bosch GmbH. Este fue presentado en el congreso de Ingeniería de la Automoción. Desde su presentación hasta nuestros días este se ha convertido es uno de los protocolos líderes en la utilización del bus serie. El profesor Wolfhard Lawrenz de la Universidad Alemana de ciencias aplicadas de Brunswick-Wolfenbüttel, fue la persona encargada de poner el nombre al nuevo protocolo de comunicaciones.

A comienzo de la década de los 80 varios ingenieros de la empresa Bosch se encontraban evaluando diferentes protocolos de comunicaciones serie, el objetivo principal era el de buscar un bus que se pudiera utilizar en las comunicaciones de los elementos electrónicos que se encontraban en el interior de los automóviles, pero esa búsqueda resulto totalmente infructuosa. Es por esto por lo que este protocolo se ideó para aumentar la fiabilidad, seguridad y funcionalidad, además de obtener una reducción en el gasto de combustible, debido a que la reducción en el peso y la complejidad de las comunicaciones en los automóviles mediante la reducción del cableado iban a favorecer este hecho. Pronto ingenieros de otras compañías como Mercedes-Benz e Intel se involucraron en las fases iniciares de la creación del nuevo sistema.

No fue hasta mediados del año 1987 cuando la empresa Intel presentó el primer chip controlador de bus CAN, en concreto el 82526. Poco tiempo después la empresa Philips presentaría su alternativa con el chip 82C200. Los controladores CAN actuales son completamente diferentes a esos primeros chips que se presentaron en la década de los 80. Los chips actuales son distintos en cuanto a filtros de aceptación y control de mensajes. Las tecnologías elegidas por estos fabricantes en sus primeros chips eran completamente diferentes: Intel se decidió por usar el concepto FullCAN; mientras que Philips se decantó por implementar sus chips con BasicCAN. FullCAN requería menos carga de CPU que BasicCAN, pero en contraposición FullCAN estaba limitado en cuanto a la cantidad de mensajes que podían ser recibidos. Por el contrario, BasicCAN requería de menos silicio, algo que permitía tener un menor coste de fabricación. Hoy en día estos términos son totalmente obsoletos.

Como se ha comentado en puntos anteriores CAN nació con la idea de ser utilizado para las comunicaciones en el sector del automóvil, pero con el paso del tiempo este empezó a ser utilizado en aplicaciones de otros sectores diferentes para la que fue desarrollado inicialmente:

- La empresa finlandesa de ascensores Kone, fue una de las primeras empresas en beneficiarse de las ventajas que presenta CAN.
- La oficina de ingeniería suiza Kyaser introdujo su uso a los fabricantes de maquinaria textil, llegando incluso a fundar el Grupo de usuarios textiles CAN.

- En Holanda, Sistemas médicos Philips incorporo el bus para lar redes internas de sus máquinas de rayos X. La especificación para mensajes Philips, representó la primera capa de aplicación para redes CAN.
- Al final de la década de los 80 se realizaron todo tipo de propuestas para el uso del bus CAN en multitud de aplicaciones, como ejemplo la creación de un sistema basado en bus CAN para vehículos agrícolas.

En el año 1992 Mercedes-Benz incorporo CAN en sus automóviles de alta gama. El sistema se componía de dos redes de comunicaciones CAN, una red de alta velocidad, en la que se comunicaban las ECUs del motor, la unidad de control de la caja de cambios y el tablero de instrumentos; y una red de baja velocidad, para el sistema de aire acondicionado y del resto de dispositivos internos, ambas redes se interconectaban mediante Gateway. El hecho de que Mercedes-Benz empezara a utilizar el bus CAN hizo que otros fabricantes de automóviles comenzaran redes CAN en sus modelos de lujo, por ejemplo, BMW, Jaguar, Volvo, Saab y Volkswagen, con el tiempo se fueron incorporando otras empresas del sector.

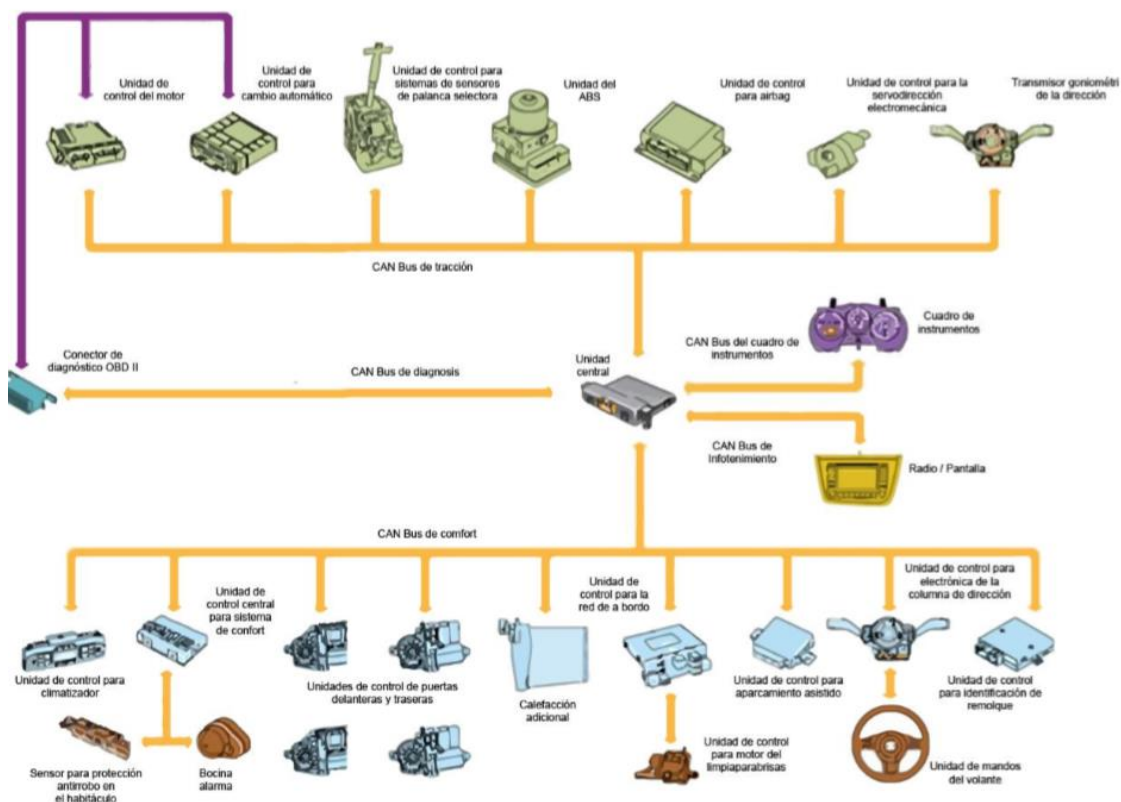


Ilustración 12 - Ejemplo bus CAN SEAT Altea [11]

En marzo de ese mismo año se fundó oficialmente la organización CiA, en esta se integraron usuarios internacionales y grupos de fabricantes. La primera publicación técnica apareció poco después trató sobre la capa física, en ella CiA recomendaba utilizar transceptores que cumplieran con la normativa ISO 11898. Una de las primeras tareas de CiA fue generar una especificación de la capa de

aplicación, esta se llamó CAL, esta se desarrolló a partir de los sistemas médicos diseñados por Philips PMS [11].

3.2. Estándar CAN 2.0

La especificación CAN 2.0 de Bosch se sometió a su estandarización internacional a comienzos de los años 90. Concretamente en noviembre de 1993, después de diversos conflictos, se publicó el estándar ISO 11898, este definía la capa física del bus CAN para velocidades de hasta 1Mbps. Al mismo tiempo un formato de CAN tolerante a fallos se incluyó en la norma ISO 11519-2. En 1995 este se amplió para incorporar la descripción del identificador CAN de 29 bits.

Por desgracia todas la estandarizaciones y especificaciones publicadas de CAN tenían errores o no estaban completas. Para que no se produzcan incompatibilidades, Bosch se cerciora de que todos los microcontroladores CAN cumplen con el modelo de referencia que ellos definieron.

Con el paso del tiempo las especificaciones de CAN se han ido revisando y estandarizando en diferentes secciones:

- La norma ISO 11898-1 describe la capa de transmisión de datos CAN.
- La ISO 11898-2 define la capa física CAN no tolerante a fallos.
- Mientras que la ISO 11992 hace referencia a la interfaz para camiones y remolques.
- ISO 11783 que se refiere a la maquinaria agrícola, definiendo los perfiles del uso de CAN en el US-protocol J1939.

Es conveniente recordar que CAN 2.0 soporta dos formatos de mensaje, estos se diferencian principalmente en la longitud que tiene el campo de arbitraje:

- Formato base: el campo de arbitraje contiene 11 bits y es conocido como CAN 2.0 A. En este caso el bit IDE se envía como dominante.

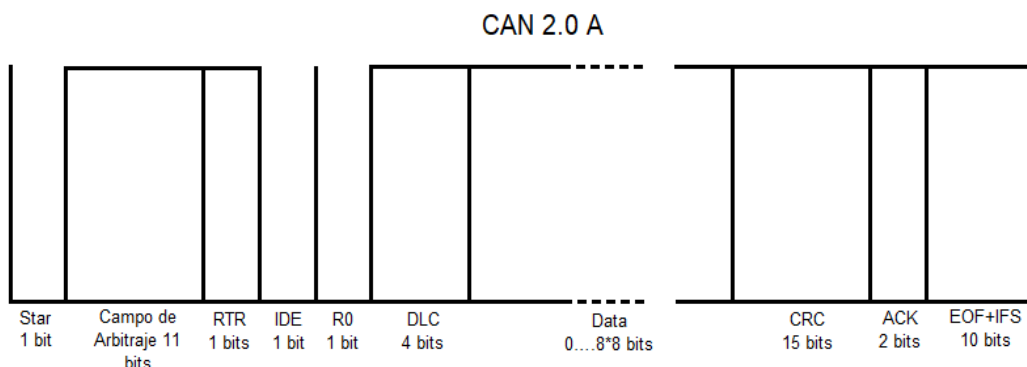


Ilustración 13 - Formato base

- **Formato extendido:** el campo de arbitraje contiene 29 bits y es conocido como CAN 2.0 b. En este caso el bit IDE se envía como recesivo.

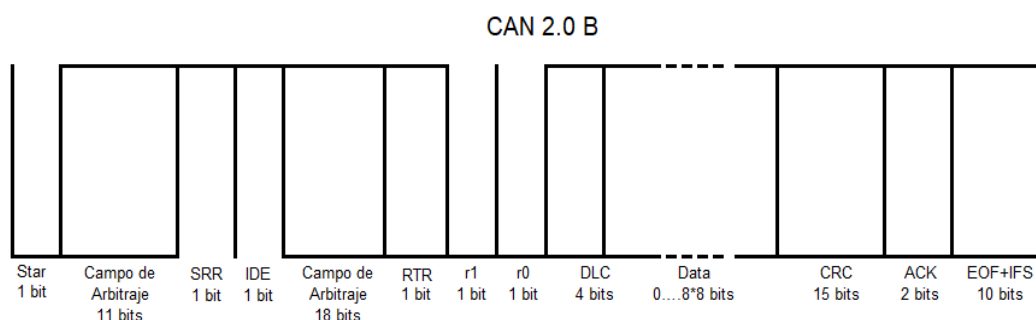


Ilustración 14 - Formato extendido

Los dos formatos de CAN 2.0 pueden coexistir a la vez en la misma red, pero en caso de que se produzca una colisión los mensajes de formato base tienen prioridad. Hay que decir que hay disponibles controladores para ambos tipos de mensajes, pero hay que decir que los controladores de formato extendido también pueden trabajar con los mensajes de formato base, mientras que los que trabajan solo con formato base no lo pueden hacer con el formato extendido.

Nivel	CAN 2.0 A	CAN 2.0 B
Aplicación	(Protocolo específico)	(Protocolo específico)
Presentación		
Sesión		
Transporte		
Red		
Enlace	CAN Object layer	LLC
	CAN Transfer layer	MAC
Físico	ISO 11898	ISO 11898

Tabla 4 - Capas definidas bus CAN

Como podemos ver en la tabla anterior, CAN se subdivide en capas de acuerdo con el modelo de OSI, de las cuales el estándar CAN solo define las dos primeras capas, la capa física y la capa de enlace. Una singularidad de CAN es que la capa de aplicación y enlace no están totalmente separadas, sino que tienen un cierto vínculo. Se puede ver también que, en la capa de enlace, las dos subcapas reciben nombres diferentes, pero las funciones que realizan son idénticas [11] [12] [13].

3.4. Características básicas del bus CAN

En este punto se presentarán las características básicas del bus CAN para de esta manera poder tener una visión general del mismo. Las características básicas del bus son las siguientes:

- Económico y sencillo: están son dos de las razones principales que motivaron su desarrollo. La necesidad de reducir los costes y la complejidad del cableado, son los principales motivos que llevaron al desarrollo de este bus de comunicaciones.
- Estandarizado: Un elemento que permitió el éxito del bus CAN es que se trata de un bus estandarizado en la ISO 11898, esta a su vez queda dividida en diferentes partes cada una de las cuales aborda diferentes aspectos de CAN. El estandarizado del bus permite que los componentes fabricados por diferentes empresas no tengan problemas de compatibilidad.
- Medio de transmisión adaptable: gracias al bus CAN se reduce mucho la necesidad de cableado en comparación con otros sistemas. Las comunicaciones en el bus CAN se realizan a través de par trenzado, a este par trenzado se le pueden añadir los cables de alimentación que permiten la alimentación de los nodos si es necesario. El bus CAN tiene la particularidad que le permite trabajar con un solo cable.
- Estructura definida: la comunicación que se produce entre los nodos se realiza a través del par trenzado, esta comunicación se realiza mediante bits que tienen una longitud limitada con una estructura definida en campos que conforman los mensajes.
- Programación sencilla: esta está basada en la escritura de registros de los dispositivos CAN.
- Número de nodos: la especificación básica no limita el número máximo de nodos, de forma que el número de nodos depende de las especificaciones de los controladores CAN. Aunque muchas especificaciones de los buses de campo limitan a 64 como el número máximo de nodos, en mi caso he visto buses de campo que permiten hasta un máximo de 120 nodos.
- Garantía de tiempos de latencia: CAN permite garantizar que se transmitirán cierta cantidad de datos en un tiempo concreto, esto quiere decir que la latencia de extremos a extremo no superara un nivel

específico de tiempo. Recordemos que CAN es un bus cuya transmisión se realiza siempre en tiempo real.

- Optimización del ancho de banda: Los métodos que utiliza CAN para enviar los mensajes a través de la red, permiten tener un mejor empleo del ancho de banda disponible.
- Desconexión autónoma de nodos defectuosos: si el sistema detecta que un nodo de la red está funcionando de forma defectuosa, el sistema puede desconectarlo de la red y seguir funcionando con total normalidad. De forma contraria, si se desea incorporar un nodo nuevo a la red se podrá hacer sin que este afecte al funcionamiento de esta, y sin necesitar de reprogramar el mismo.
- Velocidad flexible: ISO define dos tipos de redes CAN: una red de alta velocidad que llega hasta los 1 Mbps, esta se define en la norma ISO 11898-2, y tiene una red de baja velocidad que es tolerante a fallos que está definida en la ISO 11898-3.
- Relación velocidad-distancia: la velocidad del funcionamiento del bus también depende de la distancia que está cubriendo. La distancia máxima estimada es de 1000 metros, aunque esta se puede aumentar mediante bridges o repetidores. En la siguiente tabla mostramos la relación entre velocidad y distancia:

Velocidad (Kbps)	Tiempo de bit (μ S)	Longitud máxima Bus (m)
1000	1	30
800	1.25	50
500	2	100
250	4	250
125	8	500
50	20	1000
20	50	2500

10	100	5000
----	-----	------

Tabla 5 - Relación velocidad, distancia y tiempo de bit

- Orientado a mensajes: CAN es un protocolo orientado a mensajes y no a conexión, esto quiere decir que la información que se desea enviar se descompone en mensajes, a cada mensaje se le asigna un identificador y son encapsulados en tramas para su transmisión. El identificador de cada mensaje es único, a partir de este cada nodo decide si acepta o no dicho mensaje. Los mensajes además están priorizados para optimizar el ancho de banda
- Multidifusión (multicast): Esto permite que todos los nodos accedan al bus de forma simultánea con sincronización de tiempos.
- Medio compartido (broadcasting): la información es enviada por la red a todos los nodos de forma simultánea. Cada nodo tiene que decidir si la información le concierne o no.
- Detección y señalización de errores: CAN tiene una gran capacidad para la detección de errores, para ello dispone de cinco mecanismos que le permiten esta detección, tres de ellos a nivel de mensaje y dos a nivel de bit. Además, los errores pueden ser señalizados.
- Retransmisión automática de tramas erróneas: CAN no solo tiene la capacidad de detectar y señalar errores, sino que además puede retransmitir las tramas erróneas, esto le permite mantener la integridad de los datos. Estos procesos son completamente transparentes al usuario.
- Jerarquía multimaestro: CAN es un sistema multimaestro, esto quiere decir que puede tener más de un maestro al mismo tiempo funcionando en el bus, esto quiere decir que todos los nodos son capaces de transmitir, permitiendo de esta manera la construcción de sistemas inteligente y redundantes.

4. Niveles OSI de ISO bus CAN

Como se comentó en apartados anteriores muchos de los buses de campo no definen todas las capas que encontramos definidas en OSI, en concreto el bus CAN solo define dos de las capas definidas en OSI. La capa física que queda establecida en la norma 11898 y la capa de enlace que a su vez queda dividida en dos subcapas LLC y MAC. Durante este apartado se profundizará en el funcionamiento de la Capa Física y Capa de Enlace del bus CAN.

4.1 Capa física

El protocolo CAN como se ha visto en puntos anteriores define la capa física y la capa de enlace de datos basándose en el modelo OSI. Por el contrario, ISO define un estándar en el que incluye al protocolo CAN incorporando una parte extra que pertenece a la capa física. Debido a esto los puntos que se mostrarán a continuación estarán basado en el estándar ISO [14] [15] [16].

4.1.1 Codificación de bit

El bus CAN realiza la transmisión de las tramas de mensajes mediante codificación NRZ. Este es un método de codificación binaria en el que los unos se representan normalmente por un voltaje positivo, mientras que los ceros se pueden representar por un voltaje nulo o un voltaje negativo. Podemos ver esto en la imagen que tenemos a continuación, en ella se puede ver el Unipolar NRZ en el que los unos se representan por un valor positivo y los ceros por un valor nulo, mientras que en el (bi)polar NRZ el uno se representa por un valor positivo mientras que el cero queda representado por un valor negativo.

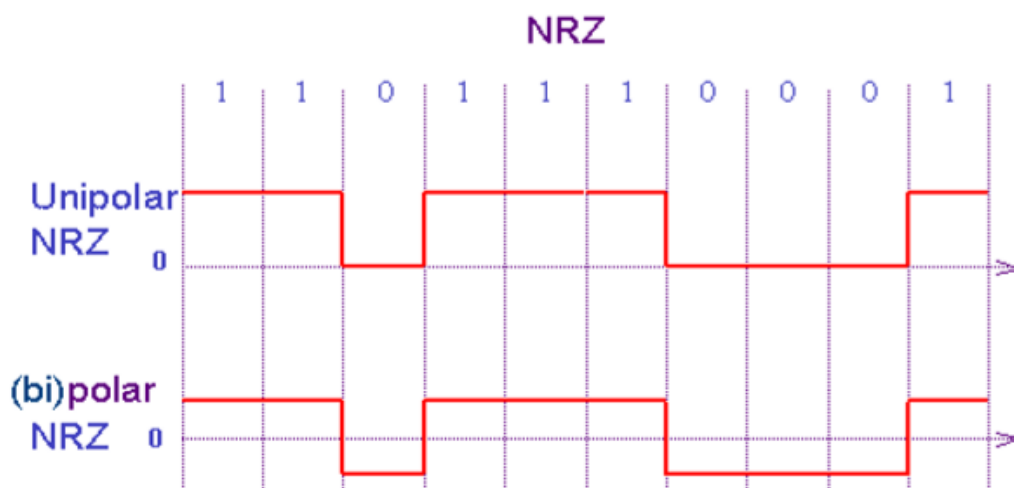


Ilustración 15 - Codificación NRZ

Esta codificación a diferencia de otras como la RZ, no tiene flancos de subida o bajada en los bits, esto se produce porque no se da un voltaje nulo tras cada voltaje positivo o negativo. Se puede ver la diferencia entre NRZ y RZ en las ilustraciones utilizadas como ejemplo.

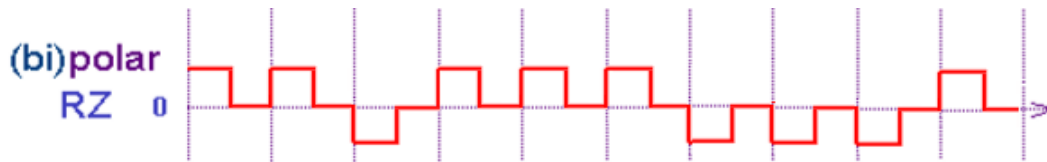


Ilustración 16 - Codificación RZ

En la codificación NRZ el nivel de la señal permanece constante durante el tiempo de bit, esto hace que solo se requiera de un tiempo para la representación de los bits. Esto quiere decir que en NRZ el nivel de señal puede permanecer constante durante grandes periodos de tiempo, esto se produce si se reciben muchos bits consecutivos del mismo valor. Se deben de implementar mecanismos para evitar que esto se produzca.

En concreto uno de los métodos más utilizados es el de violación digital (bit stuffing). Debido a la inexistencia de flancos durante el tiempo de bit, se pueden producir desincronizaciones en los nodos cuando se envían muchos bits seguidos con la misma polaridad.

El método de violación digital lo que hace es insertar bits sin información, estos bits sin información se colocarían de forma que no se sobrepasara el número de bits máximo con la misma polaridad. Estos bits sin información se introducirán siempre que se tengan 5 bits consecutivos con la misma polaridad, si la trama es de unos, se insertan ceros, si la trama es de ceros se inserta un bit de polaridad positiva y un bit de polaridad negativa.

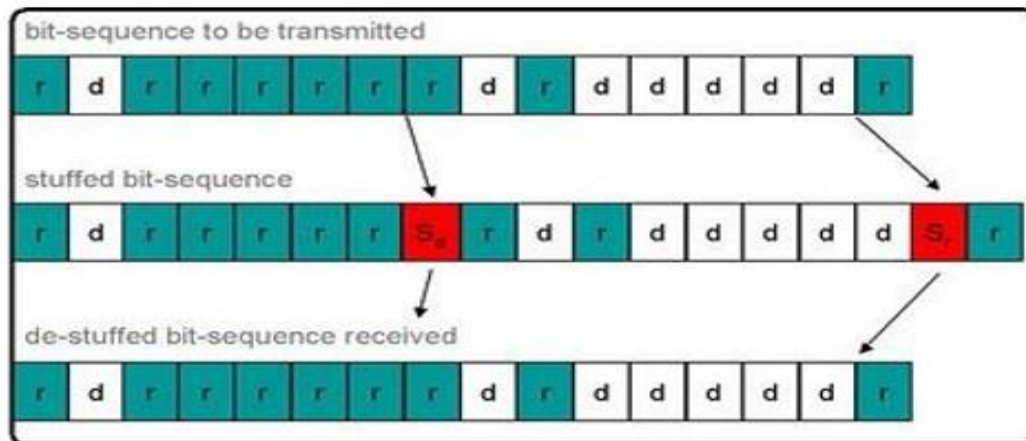


Ilustración 17 - Método de violación digital

El máximo número de bits de relleno que se insertarán en una trama se calculará mediante la siguiente fórmula:

$$S_{max} = \frac{n - 1}{4}$$

Donde n se trataría del número de bits de datos. Debido a que las tramas CAN tienen en el campo de control dos bits que están reservados y toman el valor dominante, los resultados obtenidos mediante la fórmula anterior no siempre coincidirá con el valor máximo de bits de relleno [14] [15] [16].

4.1.2. Estándar ISO 11519

Este estándar es utilizado para aplicaciones de baja velocidad. La velocidad máxima es de 125 Kbps, dentro del automóvil suele ocuparse de las comunicaciones de puertas, techo corredizo, luces y asientos.

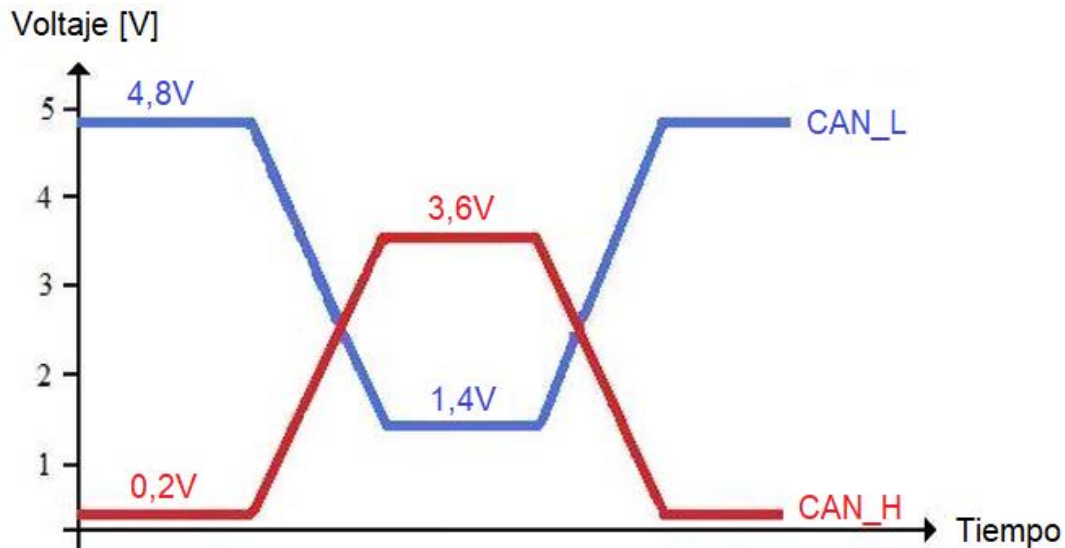


Ilustración 18 - Niveles voltaje estándar ISO 11519

Los nodos que se conectan a este bus interpretan dos niveles lógicos que son denominados dominante y recesivo:

- **Dominante:** La diferencia de tensión entre (CAN_H – CAN_L) es aproximadamente de unos 2V, con CAN_H = 3,5V y CAN_L = 1,5 V.
- **Recesivo:** La diferencia de tensión entre (CAN_H – CAN_L) es aproximadamente de unos 5V, con CAN_H = 0,2V y CAN_L = 4,8 V.

En el bus de baja velocidad se necesitarán dos resistencias por cada nodo que se encuentre conectado al bus, una R_{TH} para la señal CAN_H y R_{TL} para la señal CAN_L.

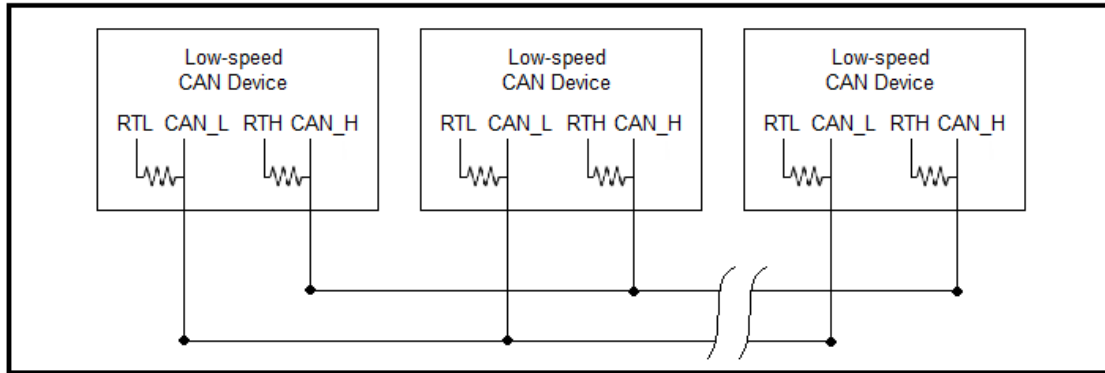


Ilustración 19 - Conexión de nodos a bus CAN

Al conectar los transceptores en el bus de baja velocidad mediante las dos resistencias permite detectar fallos en la red. El valor obtenido de la suma de todas las resistencias en paralelo debe encontrarse entre los 100Ω a 500Ω [14] [15] [16] [17].

4.1.3. Estándar ISO 11898

Las señales eléctricas del bus CAN de alta velocidad se establecieron en el estándar ISO 11898, la velocidad máxima a la que puede funcionar es de 1Mbps. Como se pudo ver en puntos anteriores dentro de los automóviles se utiliza para interconectar los elementos de mayor relevancia.

Voltaje [V]

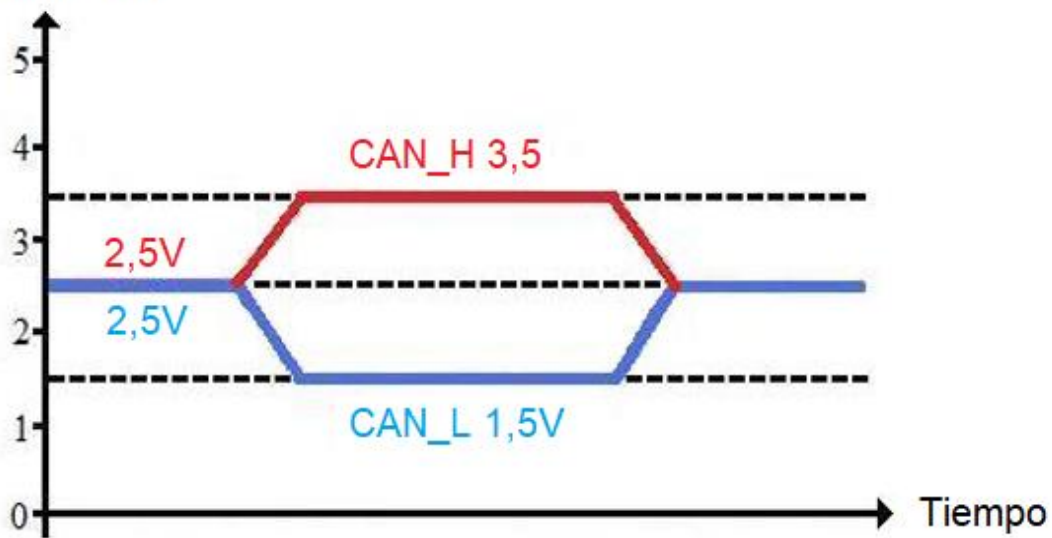


Ilustración 20 - Niveles de voltaje estándar 11898

Como en el estándar visto con anterioridad, nos encontramos con dos niveles lógicos que son el nivel dominante y recesivo:

- **Dominante:** La diferencia de tensión entre ($CAN_H - CAN_L$) es aproximadamente de unos 2V, con $CAN_H = 3,5V$ y $CAN_L = 1,5 V$.

- Recesivo: La diferencia de tensión entre (CAN_H – CAN_L) es aproximadamente de unos 0V, con CAN_H = CAN_L = 2,5 V.

La comunicación que se produce en el bus mediante un par de cables trenzados que constituyen una transmisión en línea. Esta transmisión debe de estar configurada mediante valores correctos, si esto no es así se con cada trama transferida se producirá una reflexión que puede originar fallos durante la comunicación. Para evitar esto se colocan dos resistencias de 120Ω, una a cada extremo del bus. El motivo por el cual se colocan dos es porque las comunicaciones son bidireccionales. Las resistencias deberán de ser capaces de disipar 0,25W de potencia [14] [15] [16].

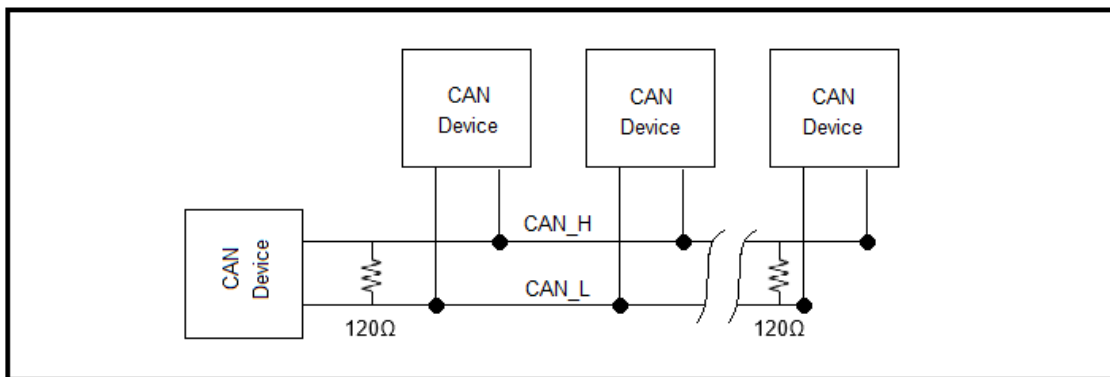


Ilustración 21 - Conexión de nodos a bus CAN

4.1.4. Características del cable y conectores CAN

La norma ISO 11898-2 es la que marca los tipos de cable que se pueden utilizar en las comunicaciones en el bus CAN, esta indica que la impedancia nominal debe ser de 120Ω y el retraso nominal de 5ns/m. La terminación de línea por de alta velocidad por ambos lados debe de ser de 120Ω como se indicó en puntos anteriores. La resistencia que debe de tener el cable debe ser 70mΩ/m. Como tenemos que tratar de minimizar esta impedancia extra que aportan los cables en función de su longitud, eso significa que cuando las distancia a cubrir es bastante grande nos interesa reducir esa impedancia que marca la norma ISO. Otra cosa importante es que, si se desea mantener la tensión diferencial para tener una red inmune a interferencias electromagnéticas, es conveniente poner unas resistencias de un mayor valor de los 120Ω. El valor de esta aumenta según aumenta la distancia a cubrir.

Cable de Bus			
Longitud Bus	Impedancia	Sección de cable	Resistencia de Terminación
0 ~ 40 m	70 mΩ/m	AWG23-AWG22	124Ω (1%)
40 ~ 300 m	<60 mΩ/m	AWG22-AWG20	127Ω (1%)
300 ~ 600 m	<40 mΩ/m	AWG20	150Ω ~ 300Ω
600 m ~ 1 km	<26 mΩ/m	AWG18	124Ω ~ 300Ω

Tabla 6 - Tipos de cableado disponibles

El estándar CAN no especifica ningún tipo de conector y por lo tanto cada aplicación puede tener un tipo de conector distinto. Sin embargo, hay algunos de ellos que se utilizan de forma habitual como el conector D-sub de 9 pines, en los que la línea CAN_L se encuentra en el pin dos y la señal CAN_H se suele encontrar en el pin siete [14] [15] [16] [17].

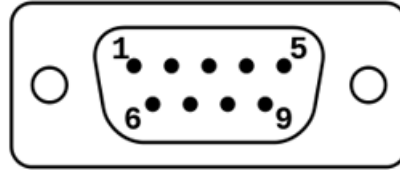
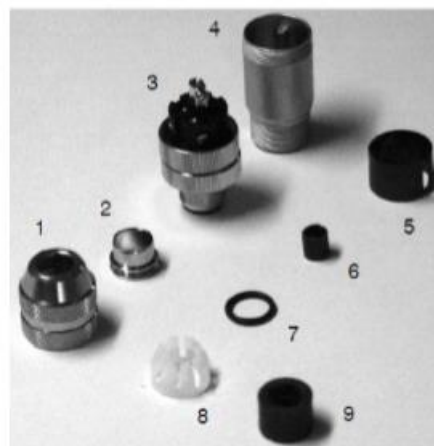


Ilustración 22 - D-sub de 9 pines

Otro de los conectores muy utilizados en buses CAN es el M12 de 5 pines, podemos ver de los elementos que se compone en la siguiente imagen.



- 1 Tornillo de empuje
- 2 Contera
- 3 Enchufe/conector
- 4 Cojinete
- 5 Anillo aislante
- 6 Junta tórica
- 7 Abrazadera
- 8 Sello
- 9

Ilustración 23 - Elementos conector M-12

En las siguientes imágenes podemos ver el proceso de montaje del conector, y el orden que llevarían algunos de los elementos que forman el conector.

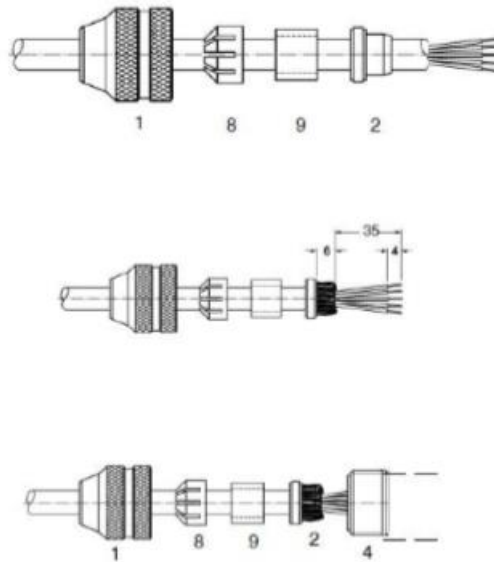


Ilustración 24 - Proceso de montaje conector M12

Pequeño resumen del proceso de montaje del conector M12:

- Pelar cable.
- Expandir la pantalla y colocarla alrededor de la contera (2).
- Cortar la trenza que sobresale.
- Pasar los cables por el cojinete (4).
- Montar contera 2, sello (9) y abrazadera (8).
- Atornillar ligeramente el casquillo (4) sobre el tornillo de empuje (1) para asegurar el cable.
- Deslizar el anillo aislante (6) sobre el cable azul, después de conectar el cable se desliza el anillo sobre la abrazadera.
- Atornillar la parte metálica de los cables al conector.
- Atornillar el cojinete (4) sobre el conector (3).
- Enroscar firmemente el cojinete (4) sobre el tornillo de empuje (1).

Como precaución recordare que la pantalla debe de estar conectada al pin 1 del conector.

Podemos ver el conector, que dispone de 5 pines donde se conectarán todos los cables del bus más el apantallamiento. Además de la tabla que nos indica el conexionado de los cables al mismo.

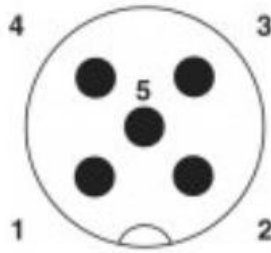


Ilustración 25 - Visión frontal de conector M12

PIN	Código de color	Asignación
1		Pantalla
2	Rojo	Voltaje de alimentación (+)
3	Negro	Voltaje de alimentación (-)
4	Blanco	CAN-High
5	Azul	CAN-Low

Tabla 7 - Asignación de pines conector M12

[18] [19]

4.2. Tiempo de Bit y Sincronización

Como hemos visto en el punto anterior CAN utiliza una codificación NRZ, esto quiere decir que no existe la posibilidad de incluir una señal de reloj dentro del flujo de datos, como se podría dar en otro tipo de codificaciones. Por lo tanto, todos los nodos sincronizan su reloj a través del tiempo nominal de bit, lo que se intenta es que el tiempo nominal de bit de cada uno de los nodos sea él mismo.

La unidad mínima de tiempo se consigue de la frecuencia del cristal y es denominada cuanto de tiempo (time quantum). Viene definido por la ecuación siguiente:

$$TQ = \frac{2 \cdot BRP}{F_{OSC}}$$

BRP es un pre escalador programable y F_{OSC} la frecuencia de oscilación de reloj. El tiempo nominal de bit viene dado por cuatro segmentos:

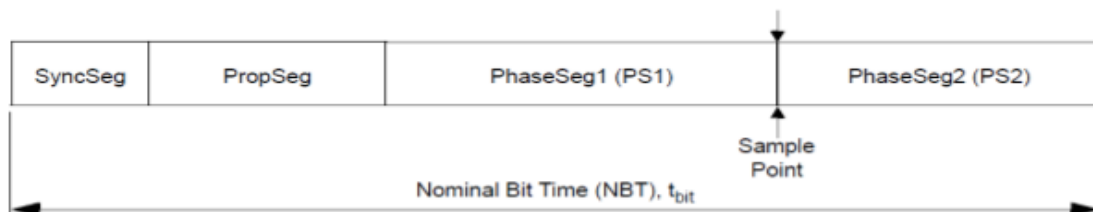


Ilustración 26 - Tiempo de bit y sincronización

- Syncseg: Este primer segmento del tiempo nominal de bit es la zona donde se espera recibir el flanco de subida o bajada de la señal. Es utilizado para la sincronización de todos los nodos de la red y su longitud es de 1 TQ.
- Proseg: Este segundo segmento es utilizado para compensar los posibles retardos producidos por el medio físico. Se le puede asignar un valor de 1 a 8 TQ.
- PhaseSeg1 y PhaseSeg2: Estos dos segmentos compensan los posibles retardos de fase de la señal. Estos segmentos también tienen un calor de cuanto de tiempo variable. Para PhaseSeg1 el valor se puede encontrar entre 1 a 8 TQ y PhaseSeg2 tendrá valores entre 2 a 8 TQ.
- Punto de muestreo (Sample point): Es el punto donde se interpretará si estamos ante un bit dominante o recesivo.

[15] [16] [17]

4.2.1. Tipos de sincronización

4.2.1.1 Hard sincronización

Hard synchronization se produce cuando hay un cambio de estado de recesivo a dominante en el bus, es decir cuando el bus pasa de no estar transmitiendo a transmitir. Este cambio se produce por SOF, este se encarga de indicar cuando se produce el comienzo de la transmisión de un mensaje. Esta es la señal que permite la sincronización de todos los nodos que están en la red, para ello se fija el valor de transición del bit dentro del segmento de sincronización Syncseg [15] [16] [17].

4.2.1.2. Resynchronization

Hard synchronization se encarga de sincronizar los nodos al principio de la transmisión, pero esto es insuficiente para poder controlar el tiempo de bit, ya que durante la transmisión se pueden producir desfases entre el transmisor y receptor mientras se produce la transmisión de la trama. Es Resincronización (Resynchronization), la encargada de corregir estas pequeñas desviaciones durante la transmisión.

Uno de los objetivos que se buscan es que el paso de recesivo a dominante se produzca dentro del Syncseg, esto es necesario para evitar que en el punto de muestreo no se pueda interpretar un estado de bit erróneo. A peras de ello puede haber ocasiones en los que esa transición se de en segmento diferente al deseado. Esto se corrige modificando el punto de muestreo, para ello lo que se hace es modificar la longitud de los segmentos PhaseSeg1 y PhaseSeg2. Lo que se hace es acortar o alargar dichos segmentos hasta una longitud dada por el SJW.

Dependiendo de donde se produzca la transición de bit se modificarán los segmentos PhaseSeg1 y PhaseSeg2 de distinta forma. Esto se debe a que la transición del bit se puede producir dentro de SyncSeg, antes o después del punto de muestreo.

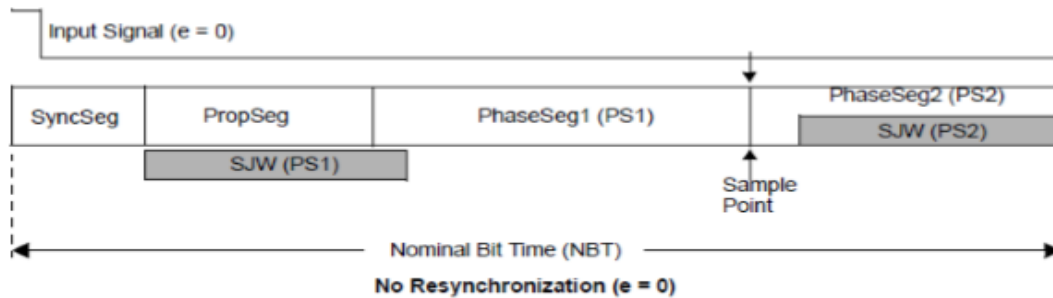


Ilustración 27 - Resincronización

En este primer ejemplo se puede ver como la transición se produce dentro de SyncSeg, esto quiere decir que no tenemos retraso de fase por lo que no se necesita modificar los segmentos de fase.

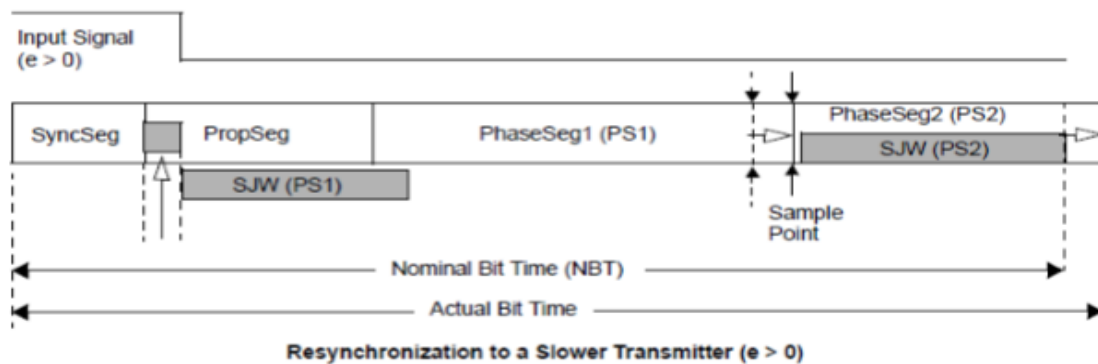


Ilustración 28 - Resincronización

En el segundo caso la transición se produce antes del punto de muestreo y fuera del segmento de sincronización. Para corregir esta situación se modifica el punto de muestreo, para ello se alarga PhaseSeg1 una cantidad igual al error de fase.

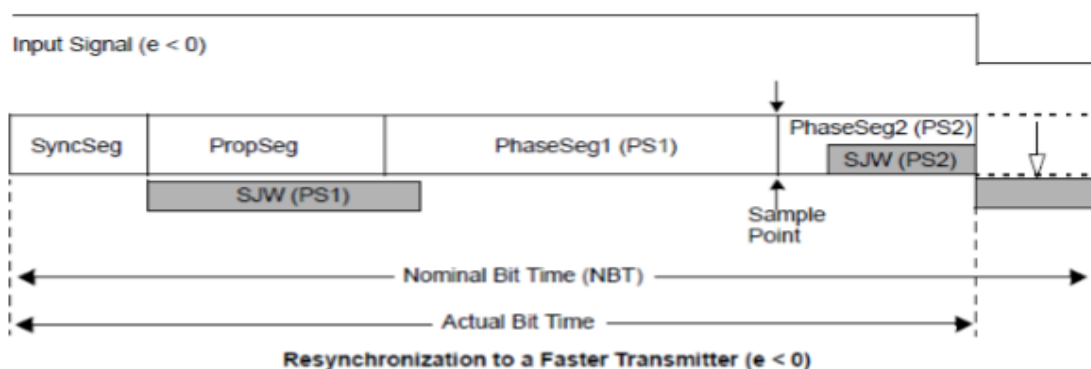


Ilustración 29 - Resincronización

Por último, vemos cuando la transición se produce después del punto de muestreo. Esto se resuelve restado a PhaseSeg2 el error de fase que tenemos [15] [16] [17].

4.3. Capa de Enlace

Esta capa es la encargada de controlar el flujo de información que se produce entre los diferentes nodos de la red. Es decir, es la encargada de la transmisión de los bits de datos en tramas de información, se ocupa de que dichos mensajes lleguen a su destino sin errores, además de controlar las secuencias de transmisión, de los acuses de recibo y si se produce un error durante la transmisión se encarga de retransmitir el mensaje con problemas. Como se mostró en puntos anteriores la capa de Enlace se divide en dos subcapas, cada una de ellas se ocupa de la realización de diferentes tareas. Estas subcapas son la MAC y LLC [11].

4.3.1. Subcapa MAC

Se podría decir que la subcapa MAC representa el núcleo del protocolo CAN. Es la encargada de presentar los mensajes recibidos a la subcapa LLC y además se ocupa de aceptar los mensajes para ser enviados a dicha subcapa por otro lado, se ocupa del mecanismo de arbitraje de acceso al medio.

La técnica de acceso al medio que utiliza CAN se denomina CSMA/CD+CR (Acceso Múltiple con Detección de Portadora, Detección y Resolución de Colisión). Cada nodo antes de poder enviar un mensaje debe de escuchar el bus durante un periodo, de esta manera podrá cerciorarse de que no se está produciendo la transmisión de ningún mensaje esto es lo que se denomina Carrier Sense (CS), una vez que se produce ese periodo sin actividad todos los nodos tienen la posibilidad de enviar mensaje es lo que conocemos como Multiple Access (MA). En el hipotético caso de que dos nodos comiencen la comunicación al mismo tiempo se producirá una colisión, esta colisión podrá ser detectada es lo que conocemos como Detection Collision (CD).

Debido al método de acceso al medio que utiliza el bus CAN que incorpora la característica Collision Resolution (CR), le permite resolver la colisión mediante la supervivencia de una de las que chocan durante la transmisión de los dos nodos. En cambio, hay otro tipo de redes que utilizan solamente la técnica CSMA/CD que hace que cuando varias tramas colisionan no pueda sobrevivir ninguna de ellas. En el caso de bus CAN la trama que sobrevive al choque es aquella que se identifica como de mayor prioridad. Para resolver la colisión se utiliza la asignación de bit dominante y bit recesivo que se vieron en apartados anteriores, dando prioridad al bit de tipo dominante. Sabiendo que el bit dominante equivale al valor lógico '0' y el bit recesivo corresponde al nivel lógico '1', lo que se haría es aplicar una función AND de todos los bits transmitidos de forma simultánea. Esto quiere decir que cuando los nodos empiezan a mandar los mensajes, cuando estos se encuentren dos bits diferentes en el bus se producirá una colisión, teniendo preferencia el bit dominante sobre el bit recesivo. Para entender mejor esto, los nodos que envían los mensajes de forma simultánea están escuchando el bus al mismo tiempo que envían los mensajes,

el nodo que manda el bit recesivo detectará que su mensaje a quedado modificado debido a la colisión y dejará de transmitir. Hay que recordar que los dos nodos seguirán transmitiendo mientras haya coincidencia entre los bits de los mensajes, cuando los bits dejen de coincidir prevalecerá el mensaje de bit dominante o de mayor prioridad. Los nodos que han tenido que dejar de transmitir su mensaje intentaran retransmitirlo lo antes posible. Para poder entender mejor el funcionamiento utilizaremos un ejemplo que podemos ver en la siguiente imagen.

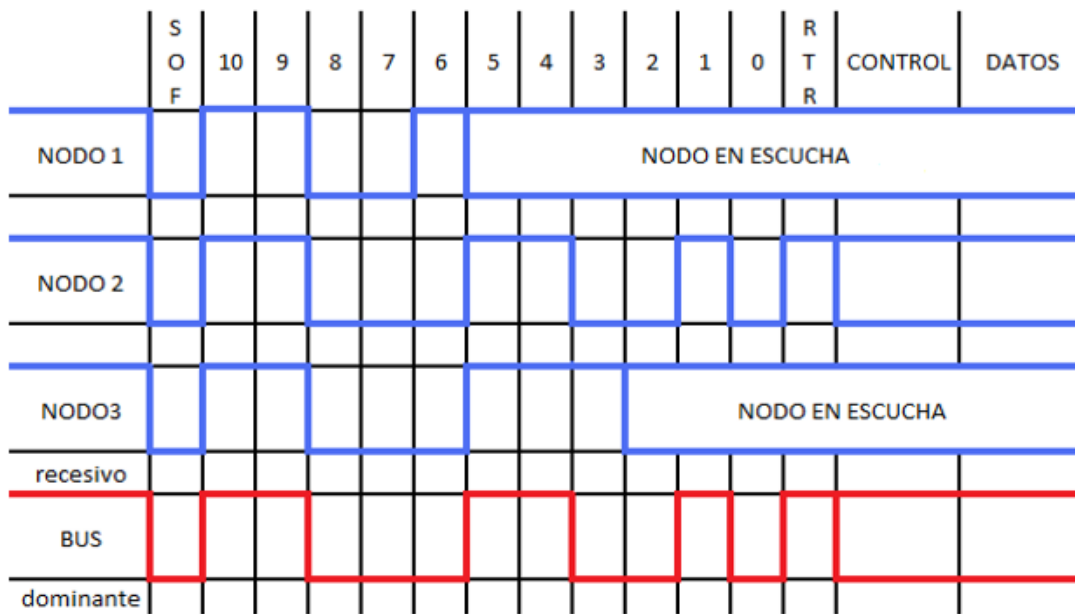


Ilustración 30 - Resolución de colisión

En este ejemplo disponemos de tres nodos que se encuentran transmitiendo al mismo tiempo sus mensajes, como podemos ver mientras los bits de los tres nodos son idénticos no se detecta ninguna colisión, por lo que los tres nodos siguen transmitiendo con normalidad. La primera colisión se produce entre el nodo 1 y 2 en el bit 6, como el bit del nodo 2 es dominante y el bit del nodo 1 es recesivo, se produce una modificación del bit 6 que el nodo 1 detecta de forma inmediata y deja de transmitir. Debido a que el nodo 1 a dejado de transmitir, se quedan los otros dos nodos transmitiendo, mientras los bits son coincidentes, es en el bit 3 donde se produce una colisión entre el nodo 2 y 3, como el bit del nodo 2 es dominante y el del 3 es recesivo el nodo 3 deja de transmitir. Como podemos ver el mensaje del nodo dos es prioritario y su mensaje sobrevive a la colisión, los nodos uno y tres intentaran retransmitir sus mensajes tan pronto como puedan.

Debemos de tener en cuenta que la especificación de CAN realizada por la empresa Bosch no indica cómo se establece si un bit es dominante o recesivo, esto quiere decir que no tiene asignado un valor físico. Según ISO 11898 cuando se utiliza par trenzado el nivel dominante corresponde a una tensión diferencial positiva en el bus, mientras que el bit recesivo corresponde a ausencia de tensión o un cierto valor negativo. Esta técnica permite combinar dos factores muy deseados en aplicaciones industriales distribuidas: la posibilidad de fijar la

latencia en la transmisión de mensajes entre los nodos y el funcionamiento multimaestro sin tener la necesidad de gestionar el acceso al medio desde las capas de software del protocolo. La prioridad de la transmisión estará definida por los bits que están contenidos en el mensaje CAN a transmitir.

Hay que recordar que a cada nodo se le debe de asignar un identificador de mensaje único, esto se hace para poder evitar que dos nodos que inicien su transmisión de forma envíen mensajes diferentes con el mismo identificador. El protocolo CAN establece que cada mensaje es único en el sistema, esto quiere decir que cada variable será transmitida por un nodo en concreto, con un identificador concreto, con una longitud fija concreta y coherente con la codificación de la información en el campo de datos [11].

4.3.2. Subcapa LLC

Para hablar de la subcapa LLC quiero recordar que CAN es un bus que está orientado a mensajes y no a conexión, una vez dicho esto podemos decir que LLC es la parte alta de la capa de enlace de datos y se encarga de definir las tareas que son independientes al método de acceso al medio. Para ello proporciona dos tipos de servicios de transmisión sin conexión que son:

- *Servicio de transmisión de datos sin reconocimiento:* este se encarga de proporcionar al usuario de la subcapa, las herramientas para poder llevar a cabo el intercambio de unidades de datos de servicio de enlace LSDU, esto se hace sin establecer una conexión de enlace de datos. La transmisión se puede realizar punto a punto, multidifusión o difusión.
- *Servicio de petición de datos remota sin reconocimiento:* este permite al usuario de la subcapa LLC las herramientas para realizar la solicitud a un nodo remoto que transmita sus LSDUs sin que sea necesario el establecimiento de una conexión de enlace de datos.

Asociados a los diferentes tipos de servicios se dispone de dos tipos de tramas llamadas, trama de datos LLC y trama remota LLC. Ambas definen identificadores de 11 bits, como vimos en puntos anteriores formato base o estándar y de 29 bits que sería el formato extendido.

Las funciones que tiene la subcapa LLC son las siguientes:

- *Filtrar mensajes (frame acceptance filtering):* el identificador de la trama no indica el nodo destino de la información contenida en el mensaje, es mediante esta función es la que permite a todo nodo receptor en la red si el mensaje está destinado a él o no.
- *Notificar sobrecarga (overload notification):* si el nodo receptor no está en condiciones de recibir un mensaje, este puede realizar la petición de retrasar la transmisión de la siguiente trama de datos o remota. Para ello se transmite una trama de sobrecarga, esta trama puede ser generada por cualquier nodo que se encuentre conectado, pero que debido a sus

condiciones no está en situación de iniciar la recepción de un nuevo mensaje. Esta trama de sobrecarga permite el retraso del inicio de la transmisión de un nuevo mensaje, cada uno de los nodos conectados puede generar un máximo de dos tramas de sobrecarga para retrasar el envío del nuevo mensaje.

- Proceso de recuperación (recovery management): la subcapa LLC también tiene la capacidad de retransmitir de forma automática los mensajes, que producto de una colisión no se han podido transmitir y de aquellos mensajes que presentan errores durante la transmisión.

[11]

4.3.3. Tramas

En el protocolo CAN el nodo emisor transmite los mensajes a todos los nodos conectados a la red sin especificar el destinatario de este, todos los nodos que están conectados al bus escuchan los mensajes enviados y luego se encargan de filtrar lo que les interesa o no.

CAN define diferentes tipos de tramas que permiten gestionar la transferencia de mensajes:

- Trama de datos: esta es la que se encarga normalmente de poner la información en el bus.
- Trama de información remota: esta trama se puede utilizar por un nodo para reclamar la transmisión de una trama de datos con la información asociada a un identificador concreto. El nodo que tenga la información que se encuentra asociada a ese identificador se encargará de su transmisión.
- Trama de error: esta trama se genera cuando algún nodo detecta un error definido.
- Trama de sobrecarga: esta trama se genera cuando algún nodo necesita de más tiempo para procesar los mensajes recibidos. De forma que no se le envíen más.
- Espacio entre tramas: las tramas de datos quedan separadas por un espacio definido. Esta secuencia predefinida es conocida como espaciado inter-trama.
- Bus en reposo: Cuando el bus no se está utilizando este se pone en reposo, esto quiere decir que se mantiene el nivel recesivo de forma constante.

En el bus CAN los nodos transmiten los datos de forma espontánea a través de las tramas de datos, esto se puede producir de forma cíclica o debido a eventos que se producen en el nodo. La trama de interrogación remota se suele utilizar

cuando se detecta un nodo nuevo en la red, esto permite actualizar la información de la red, también para la detección de estos [11].

4.3.3.1. Trama de datos

Es la trama que utilizan los nodos para enviar información a través de la red. Esta puede incluir entre 0 y 8 bytes de información. Los mensajes de datos se dividen en datos en los que se envía la información y en los que se incorporan las especificaciones del protocolo CAN:

Trama de datos estandar

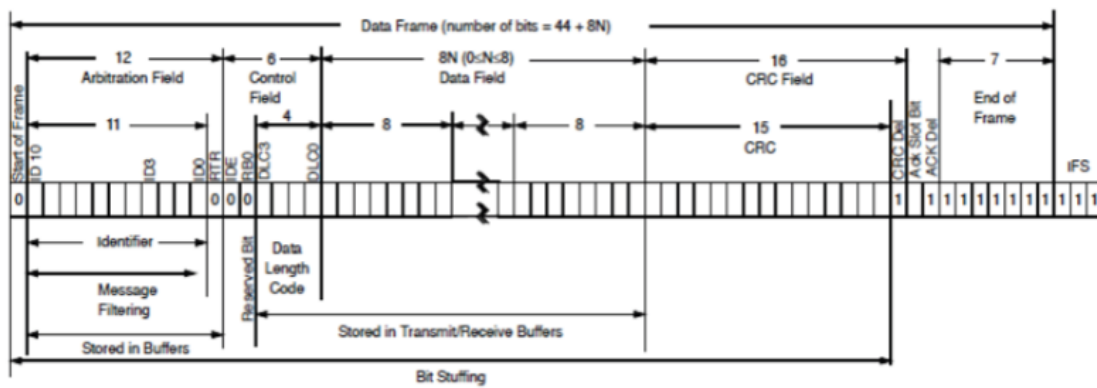


Ilustración 31 - Trama de datos estandar

Trama de datos extendida

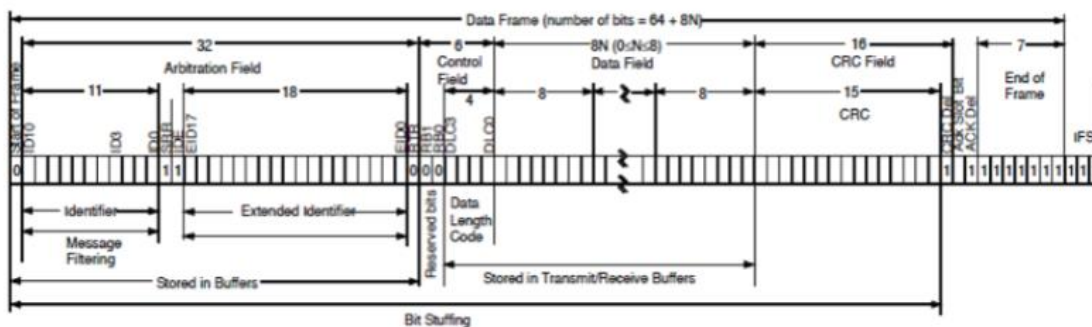


Ilustración 32 - Trama de datos extendida

- **Inicio de Trama (SOF):** se trata de una celda de un solo bit, este siempre es dominante y sirve para indicar el inicio del mensaje, permite que todos los nodos se sincronicen.
- **Celda de Arbitraje (Arbitration Field):** esta celda contiene el identificador del mensaje, por lo que es la encargada de determinar la prioridad de los mensajes en caso de colisión:

- La trama estándar dispone de un campo de arbitraje de 11 bits seguido del bit RTR, en este caso RTR será un bit dominante.
- La trama de datos extendida dispone de 11 bits de identificador base y 18 bits de identificador extendido. El bit RTR es sustituido por el bit SRR, que en este caso será recesivo. Esto permite que en caso de colisión la trama de formato estándar siempre prevalece sobre la extendida.
- Celda de Control (Control Field): este campo se compone de dos bits reservados para uso futuro, además dispone de otros cuatro bits adicionales que indican el número de bytes de datos que se están utilizando. El primero de esos bits llamado IDE indica si nos encontramos ante una trama CAN estándar (IDE dominante) o ante una trama de CAN extendida (IDE recesivo). El segundo bit de la celda llamado r0 se encuentra reservado para usos futuros y es siempre recesivo. Los otros cuatro bits llamados DLC (código de longitud) indican el número de bytes de datos que tiene el mensaje que oscila entre 0 y 8 bytes.
- Celda de Datos (Data Field): esta es la celda que tiene los datos que se envían en el mensaje, su tamaño puede oscilar entre 0 y 8 bytes.
- CRC: se trata de un código de redundancia cíclico, gracias a él se puede determinar si se han producido errores en la transmisión de la trama.
- Celda de reconocimiento (ACK): se trata de una celda que permite conocer si el mensaje ha llegado correctamente a su destino. Para ello el nodo emisor pone el bit en modo recesivo, mientras que el nodo o nodos receptores ponen el bit como dominante para indicar que se ha recibido el mensaje.
- Fin de trama (EOF): la celda de fin de trama dispone de siete bits recesivos enviados de forma sucesiva e indican el final de la trama.
- Espacio entre tramas (IFS): tiene un mínimo de tres bits recesivos.

[11]

4.3.3.2. Trama remota

Los nodos tienen la capacidad de pedir información a otros nodos. Si un nodo pide una información a otro nodo y este tiene la información requerida transmite dicha información solicitada que puede ser recibida además de por el nodo solicitante por otros nodos que pueden estar interesados.

Trama Remota

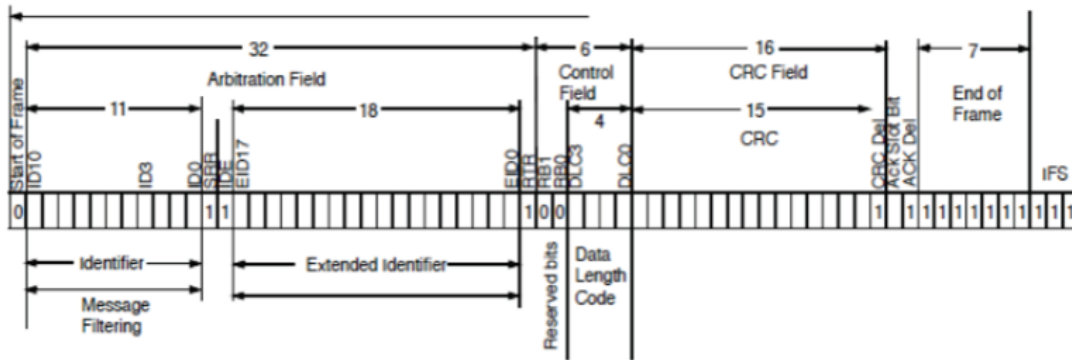


Ilustración 33 - Trama remota

Este tipo de trama envía un identificador del nodo del que se requiere la información, se diferencia de la trama de datos en que no dispone de campo de datos, además el bit RTR toma un valor recesivo. Si se envía un mensaje de datos con el mismo identificador que la petición remota y se produce una colisión, el mensaje de datos prevalecerá debido a que su RTR lleva un valor dominante [11].

4.3.3.3. Trama de error

Estas tramas pueden ser enviadas por cualquier nodo que detecte un error durante la transmisión. Esta trama dispone de los siguientes campos:

Trama de error

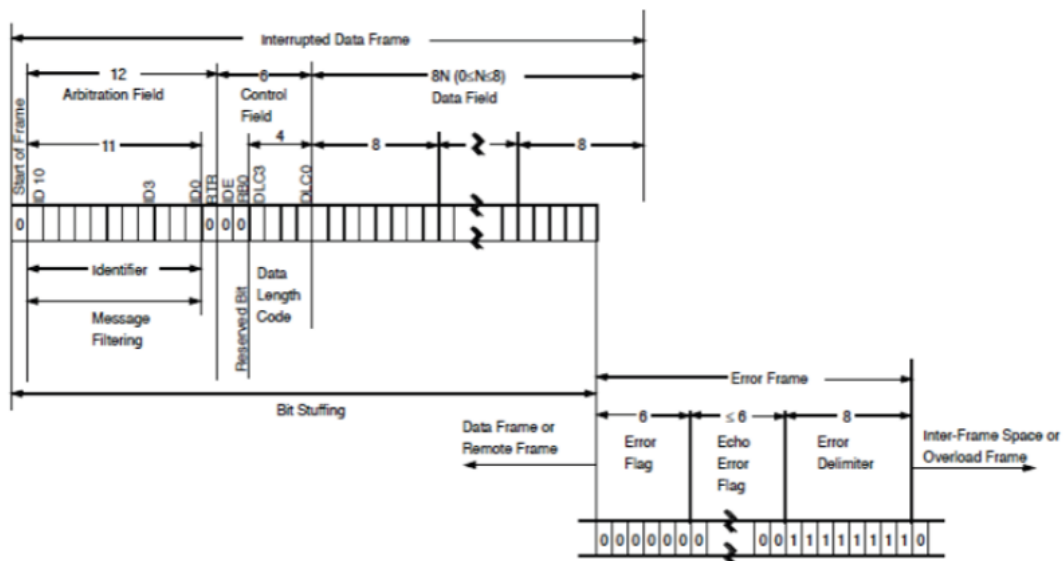


Ilustración 34 - Trama de error

- Indicador de Error (Error Flag): es distinto dependiendo el estado de error del nodo que detecta el error.
- Delimitador de Error (Error Delimeter): este dispone de 8 bits recesivos de forma consecutiva y permite a los nodos recuperar la comunicación después de producirse el error.

Si un nodo se encuentra en estado de error “Activo” y detecta un error en la transmisión de forma inmediata interrumpe la transmisión de mensaje, produciendo una indicación de error activo, esto consiste en enviar una secuencia de 6 bits dominante de forma sucesiva. Debido al envío de esta secuencia se rompe la regla de relleno de bits y esto produce que otros nodos generen tramas de error. Esto hace que el indicador de error se pueda extender entre 6 y 12 bits dominantes sucesivos. Por último, se recibe el campo de delimitación de error que este compuesto por 8 bits recesivos. Entonces el nodo interrumpido reinicia la comunicación reenviando el mensaje interrumpido anteriormente.

Si un nodo se encuentra en estado de error “Pasivo” cuando detecta un error, este se encarga de transmitir un indicador de error pasivo, este indicador es seguido por el campo delimitador de error. El indicador de error pasivo se conforma con él envío de 6 bits recesivos seguidos, esto hace que la trama de error de tipo pasivo disponga de una secuencia de 14 bits recesivos. Esto hace que la trama de error pasivo no afectará a ningún otro nodo conectado a la red, a excepción de cuando el error se detecta por el propio nodo que está transmitiendo. En ese caso los demás nodos también transmitirán tramas de error debido a que detectarán una violación en las reglas de relleno.

Tras señalar el error mediante la trama de error adecuada cada nodo transmite bits recesivos, hasta que este recibe un bit recesivo, después de eso transmite otros siete bits recesivos de forma consecutiva para finalizar el tratamiento del error.

La regla de relleno de la que hemos hablado en los párrafos anteriores consiste en que cada vez que se envían cinco bits consecutivos con el mismo valor se introduce uno de valor inverso.

Los tipos de errores que puede ser detectados por los dispositivos CAN son los siguientes:

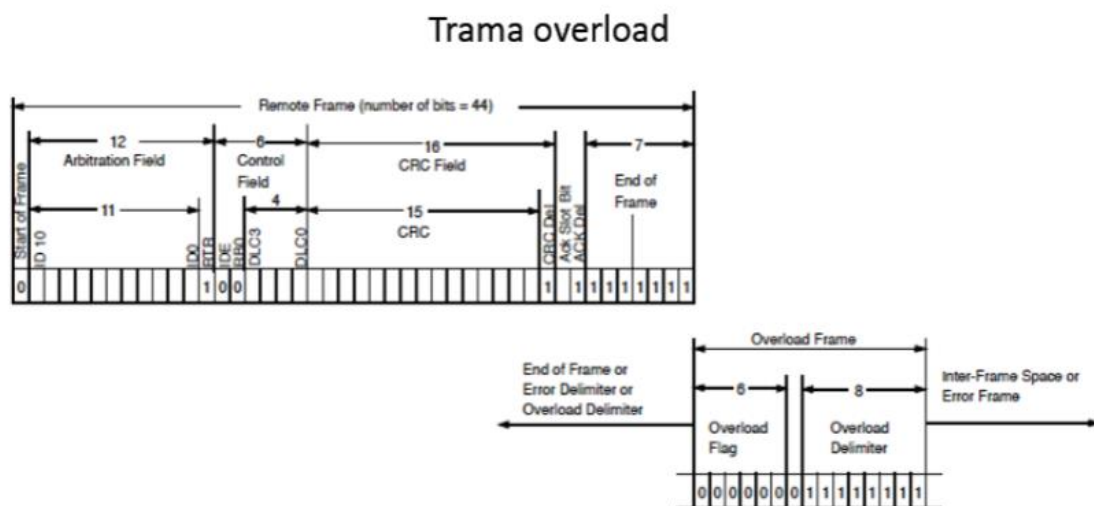
- Error de bit: Lo primero que tendríamos que indicar es que el módulo emisor está monitorizando el bus mientras manda el mensaje. Si el módulo detecta cualquier cambio de polaridad con respecto al mensaje transmitido se considerará un error de bit. Hay dos excepciones que es cuando el cambio en el mensaje se produce en el campo de arbitraje o en el bit de reconocimiento. Además, no se considera error de bit a la detección de un bit dominante por un nodo en estado de error pasivo que envía una trama de error pasivo.

- Error de relleno: Se considera error de relleno a la detección de seis bits consecutivos del mismo signo, en cualquier campo que siga la técnica de relleno de bits, esto quiere decir que cada cinco bits consecutivos del mismo signo se tienen que incorporar uno diferente de relleno.
- Error de CRC: Esto se produce cuando el receptor comprueba el CRC, esto es compara el CRC recibido con el calculado que el obtiene a partir de la trama. El campo CRC está compuesto por 15 bits y una distancia de Hamming de seis, por lo que es capaz de detectar errores de cinco bits por mensaje. Esto permite detectar errores en la trama por incidencias en el medio físico como por ejemplo el producido por el ruido.
- Error de forma: Se produce cuando un campo de formato fijo se recibe alterado como bit.
- Error de reconocimiento: Se produce cuando ningún nodo cambia el bit de reconocimiento a dominante.

[11]

4.3.3.4. Trama de sobrecarga

Esta trama puede ser enviada por cualquier nodo que no se encuentre en condiciones de recibir nuevos mensajes. Este tipo de trama tiene el mismo formato que la trama de error activo. Pero al contrario que la trama de error activo, esta solo puede ser enviada en el espacio que se produce entre tramas. De esta manera se puede diferenciar de la trama de error activa que solo puede enviarse durante la transmisión de un mensaje.



La trama de sobrecarga tiene dos campos como la trama de error:

- El indicador de sobrecarga (Overload Flag): este campo consta de 6 bits dominantes, estos pueden ser seguidos por los generados por otros nodos, por lo que se puede extender hasta un máximo de 12 bits dominantes.

- El delimitador de sobrecarga (Overload Delimeter): tiene 8 bits recesivos y marca el final de la trama de sobrecarga.

[11]

4.3.3.5. Espacio entre tramas

El espacio entre tramas se encarga de separar cualquier tipo de trama de la siguiente trama de datos o interrogación remota. El espacio entre tramas se compone como mínimo de tres bits recesivos. Esta secuencia de bits es conocida como intermission, una vez que ha finalizado un nodo en estado de error activo puede iniciar una nueva transmisión o el bus se pondrá en reposo. Un nodo en estado de error pasivo deberá de esperar otros 8 bits recesivos adicionales para iniciar la transmisión. Esto permite que los nodos en estado activo tengan preferencia en la transmisión con los nodos que están en estado pasivo [11].

5. Ejemplo de nodo CAN

Aunque inicialmente se pensó solamente en un estudio teórico del funcionamiento del protocolo CAN, esta idea inicial se fue modificando a lo largo del desarrollo del trabajo. El objetivo principal de este cambio es poder realizar un análisis más concienzudo del funcionamiento de este protocolo de comunicación. Permitiendo de esta manera poner en alza todos los conocimientos obtenidos durante las fases anteriores del trabajo.

Como el tiempo disponible es bastante limitado el ejemplo utilizado para mejorar los conocimientos sobre este bus, consiste en conectar dos Raspberry Pi mediante un bus de comunicaciones CAN.

Se empezará dando una breve descripción de las Raspberry tanto del hardware como del software utilizado, para a posteriori mostrar los diseños utilizados para la comunicación mediante CAN, realizando una breve descripción de los componentes utilizados y el método de fabricación empleado.

5.1. Raspberry Pi

Raspberry Pi se trata de un ordenador de placa reducida y de bajo coste, este podría considerarse como un ordenador de pequeño tamaño, se desarrolló en Reino Unido por la fundación Raspberry Pi, cuyo objetivo es el de iniciar a niños y adultos en el manejo de ordenadores y todo lo relacionado con ellos.

La idea de la fundación es la de disponer de un ordenador que permita a personas de todas las edades acercarse al mundo de la computación y programación.

Los desarrolladores de Raspberry fueron Eben Upton, Rob Mullins, Jack Lang and Alan Mycroft, todos trabajadores de la Universidad de Cambridge. La idea de desarrollar Raspberry Pi surgió en el año 2006, el motivo que propicio su diseño es que se dieron cuenta que los niños en la década de los 90 tenían mucha experiencia programando, en cambio 10 años después los niños solo eran capaces de realizar diseño web.

Los primeros diseños que realizaron de Raspberry Pi se realizaron con el microcontrolador Atmel ATmega644.

No fue hasta mayo de 2009 cuando se creó la Fundación Raspberry Pi, esta se fundó en Caldecote, South Cambridgeshire en Reino Unido, la Fundación Raspberry Pi se fundó como una asociación sin ánimo de lucro y es regulada por la Comisión de Caridad de Inglaterra y Gales.

El objetivo principal de la misma es el uso y entendimiento de los ordenadores en los niños. Lo que trataron es de crear un ordenador barato y portable, que permitiera a los niños adquirir conocimientos a nivel informático.

En agosto de 2011 se fabricaron las primeras placas Alpha, cuyas características eran las mismas que el modelo B, pero las primeras ventas del modelo B no se llevaron a cabo hasta el 29 de febrero de 2012.

Fue el 4 de febrero de 2013 cuando se iniciaron las ventas del modelo A, este disponía de 256Mb de RAM y no disponía de puerto ethernet, aunque su precio era inferior [20] [21].

5.1.1. Hardware

Hoy en día Raspberry dispone de varios modelos que podremos ver en la tabla siguiente, estos se basan en los diseños iniciales de la Raspberry Pi. Algunas diferencias entre los modelos antiguos y los más recientes se encuentran en el almacenamiento, procesadores, comunicaciones, etc. En la tabla se podrán ver las diferencias entre los diferentes modelos existentes, la tabla disponible contabiliza los modelos hasta marzo de 2018.

	Raspberry Compute Module	Raspberry Pi Zero	Raspberry Pi Model A+	Raspberry Pi Model B+	Raspberry Pi 2 Model B+	Raspberry Pi 3 Model B
	Raspberry Pi Compute Module 3 (+)	Raspberry Pi Zero W			Raspberry Pi 2 v1.2	Raspberry Pi 3 Model B+(#)
Precio	\$25 USD	\$5 USD \$10 USD	\$20 USD	\$25 USD	\$35 USD	\$35 USD
SoC	BCM2835 BCM2837(+)	BCM2835	BCM2835	BCM2835	BCM2836 BCM2837 en v1.2	BCM2837 BCM2837B0(#)
Fabricante	Broadcom	Broadcom	Broadcom	Broadcom	Broadcom	Broadcom
CPU	ARM1176JZF-S ARM Cortex-A53(+)	ARM1176JZF-S	ARM1176JZF-S	ARM1176JZF-S	ARM Cortex-A7ARM Cortex-A53 en v1.2	ARM Cortex-A53
Instrucciones	ARMv6 32bits ARMv8 64bits (+)	ARMv6 32bits	ARMv6 32bits	ARMv6 32bits	ARMv7 32bits ARMv8 64bits en v1.2	ARMv8 64bits
Cores	Single-core Quad-core (+)	Single-core	Single-core	Single-core	Quad-core	Quad-core
Velocidad	700MHz 1200MHz (+)	1000 MHz	700MHz	700MHz	900 MHz	1200MHz 1400MHz (#)
RAM	512MB 1GB	512MB	512MB	512MB	1GB	1GB
Almacenamiento	4GB eMMC flash internos versión LE sin memoria interna (+)	MicroSD slot	MicroSD slot	MicroSD slot	MicroSD slot	MicroSD slot
GPU	250MHz Broadcom Video Core IV 400MHz Broadcom Video Core IV (+)	250MHz Broadcom Video Core IV	250MHz Broadcom Video Core IV	250MHz Broadcom Video Core IV	250MHz Broadcom Video Core IV	400MHz Broadcom Video Core IV
Conexiones	200-pin tipo DDR2 SO-DIMM	miniHDMI 1 x micro USB2 40 GPIO pins MIPI camera connector*** Wi-Fi Bluetooth *****	HDMI 1x USB2 port 40 GPIO pins MIPI camera connector MIPI display DSI Vídeo compuesto (PAL y NTSC) vía 3.5 mm TRRS jack compartido con audio estéreo	HDMI 4x USB2 ports 10/100 Ethernet 40 GPIO pins MIPI camera connector MIPI display DSI Vídeo compuesto (PAL y NTSC) vía 3.5 mm TRRS jack compartido con audio estéreo	HDMI 4x USB2 ports 10/100 Ethernet 40 GPIO pins MIPI camera connector MIPI display DSI Vídeo compuesto (PAL y NTSC) vía 3.5 mm TRRS jack compartido con audio estéreo	HDMI 4x USB2 ports 10/100 Ethernet 10/100/300(#) Ethernet 40 GPIO pins MIPI camera connector MIPI display DSI Vídeo compuesto (PAL y NTSC) vía 3.5 mm TRRS jack compartido con audio estéreo Wi-Fi 2.4Ghz Bluetooth 4.1

						Wi-Fi 2,4/5GHz (#) Bluetooth 4.2 (#)
Tamaño	Largo: 6.8cm Ancho: 3cm Peso: 7g	Largo: 6.5cm Ancho: 3cm Peso: 9g	Largo: 6.5cm Ancho: 5.7cm Peso: 23g	Largo: 8.6cm Ancho: 5.7cm Peso: 45g	Largo: 8.6cm Ancho: 5.7cm Peso: 45g	Largo: 8.6cm Ancho: 5.7cm Peso: 45g
Alimentación	5 V a 2A 5 V a 3A+ (+)	5 V a 2A micro USB	5 V a 2A micro USB	5 V a 2A micro USB	5 V a 2A micro USB	5 V a 3A micro USB 5 V a 2.5A (#) micro USB

Tabla 8 - Tabla de modelos de Raspberry Pi [22]

En mi caso para poder establecer la comunicación CAN he utilizado una Raspberry 2 Model B V1.2 y Model B+. Muestro unas fotos de ambas para poder verificar los modelos utilizados.

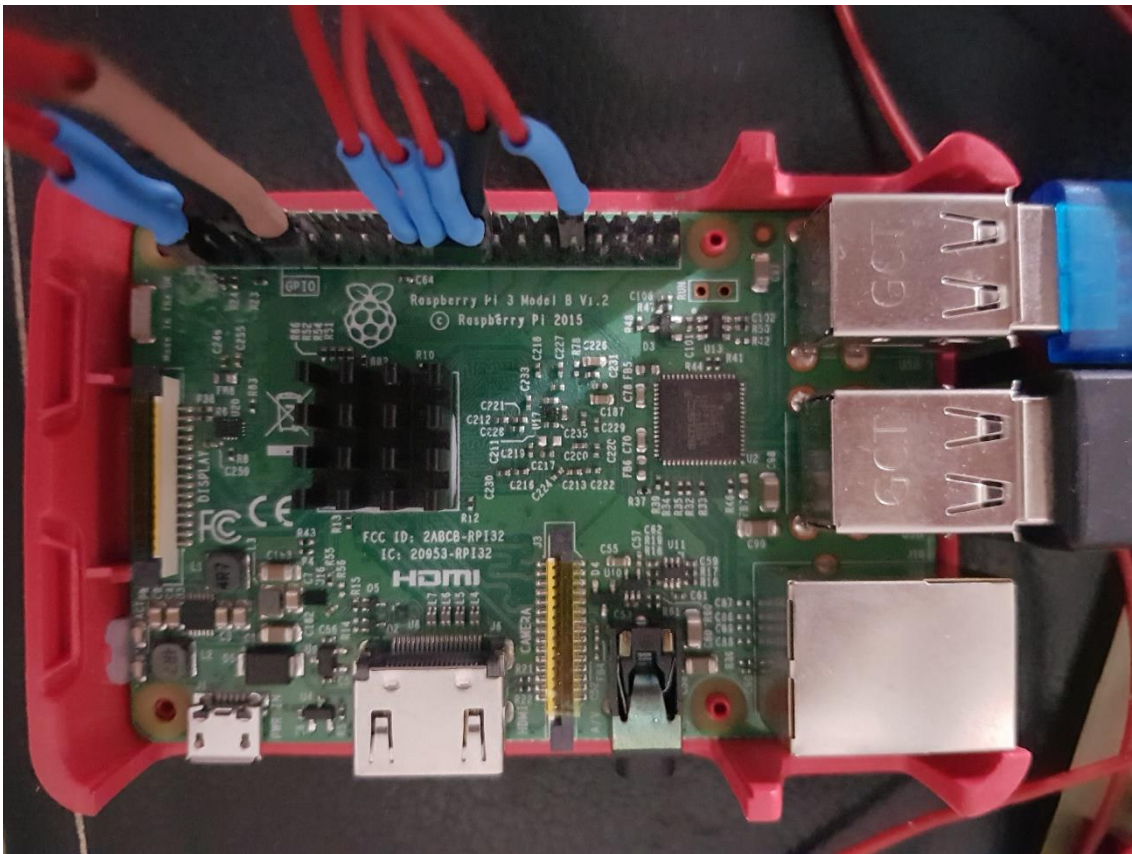


Ilustración 36 - Imagen Raspberry Pi

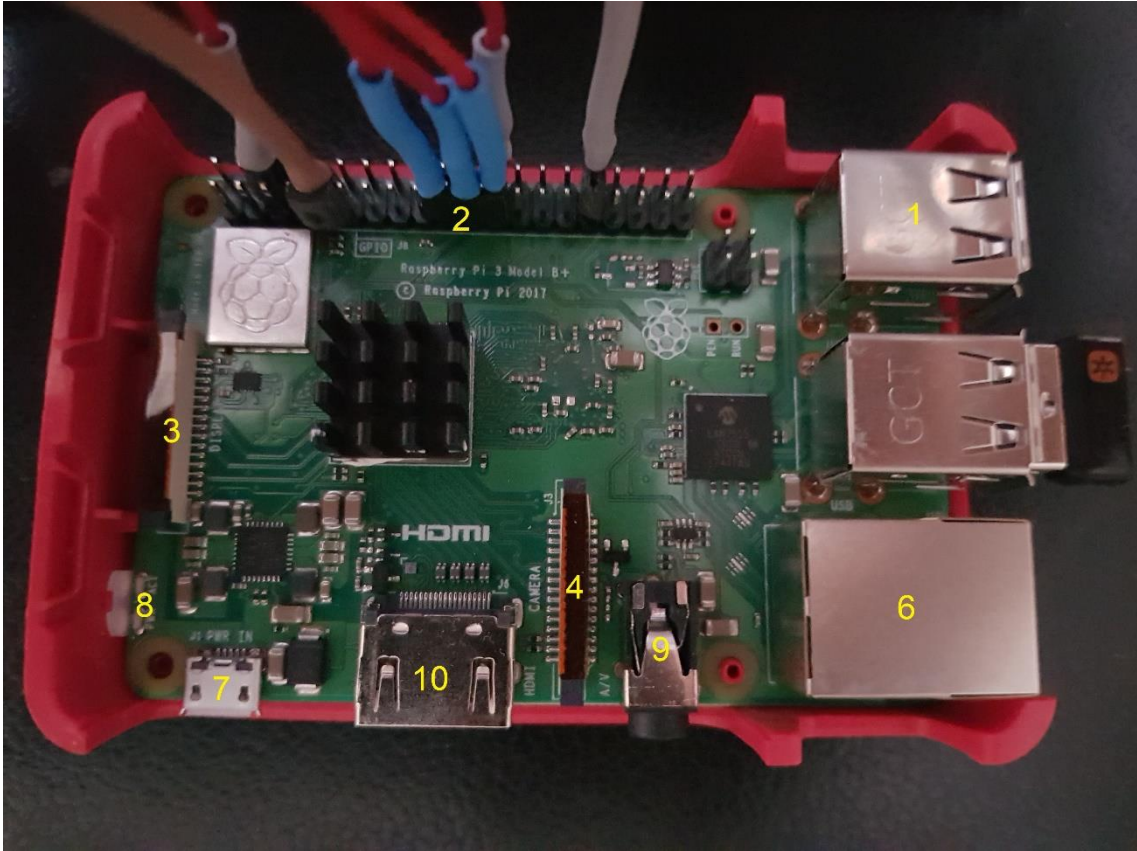


Ilustración 37 - Descripción componentes principales Raspberry Pi

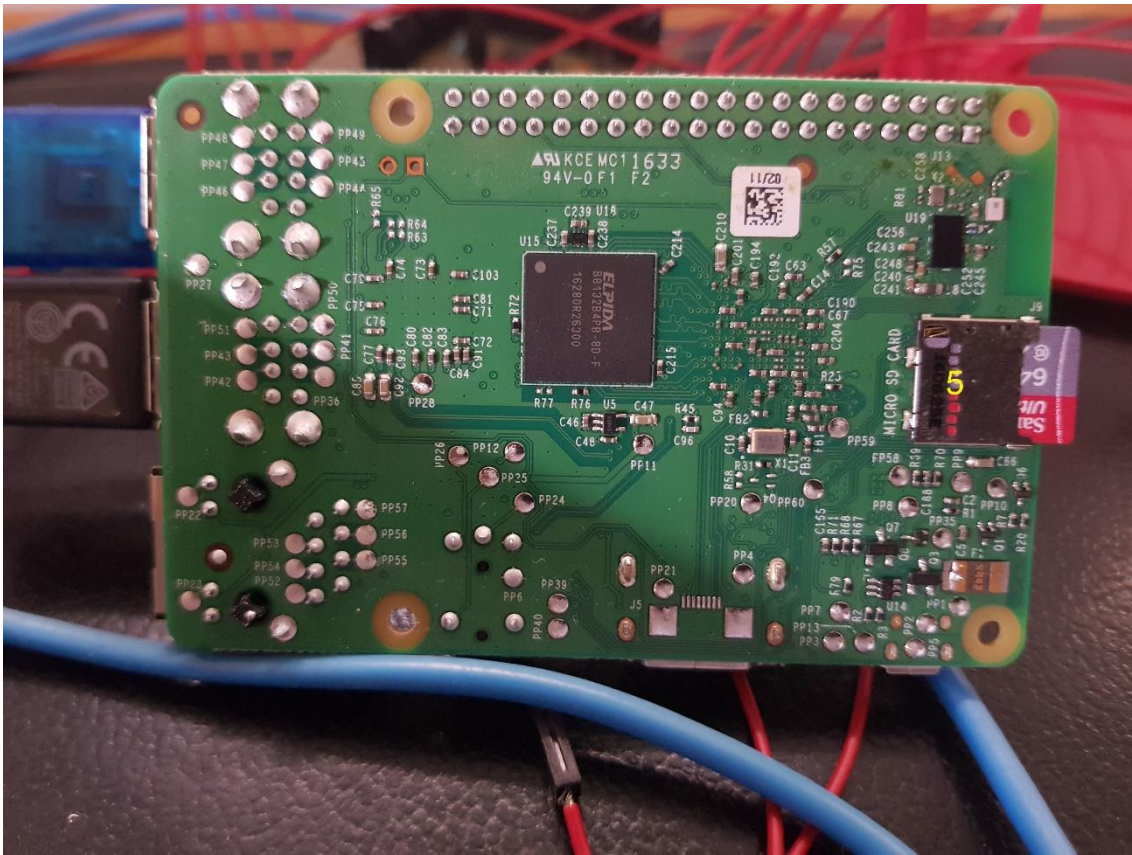


Ilustración 38 - Cara de pistas Raspberry Pi

N.º	Identificador
1	Puertos USB
2	Pines GPIO
3	Conector DSI Display Module
4	Conector CSI Camera Module
5	Ranura para la tarjeta microSD
6	Puerto Ethernet (RJ45)
7	Alimentación micro USB
8	Leds indicadores
9	Salida de audio
10	Salida HDMI

Tabla 9 - Tabla de componentes de Raspberry Pi

5.1.1.1. Puertos USB

Los dos modelos de Raspberry Pi utilizadas para la comunicación disponen de cuatro puertos USB 2.0, aunque los puertos se encuentran gestionados por chips diferentes. La Raspberry Pi 3 B+ utiliza el chip LAN 7515, este integrado permite disponer de un gran número de puertos USB en una placa pequeña. Este chip a diferencia del otro dispone de Gigabit Ethernet, la imagen utilizada es la del LAN 7500 que no es el utilizado por Raspberry Pi y la principal diferencia es que solo dispone de 1 puerto USB en lugar de los 4 USB de que dispone la Raspberry Pi. El motivo por el que se utiliza la imagen de este chip es la ausencia de

documentación del LAN 7515, esta ausencia de documentación parece que se debe a que es un chip diseñado expresamente para Raspberry Pi.

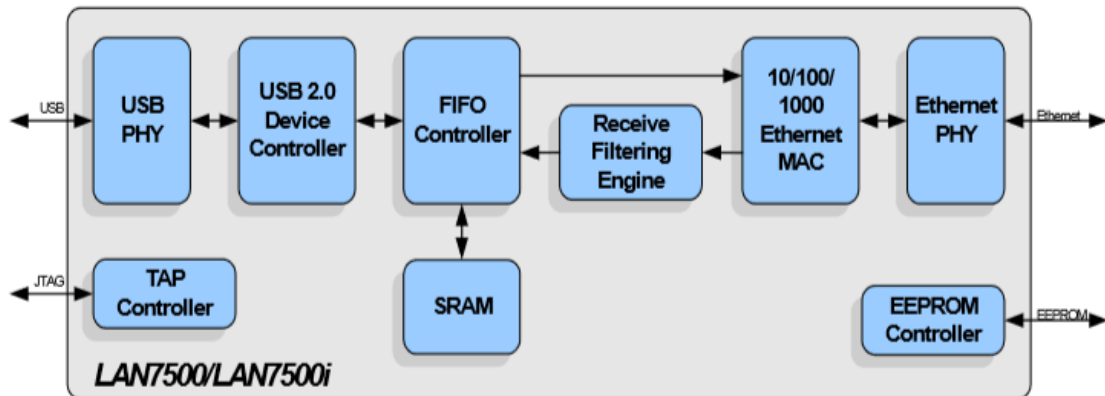


Ilustración 39 - LAN7500/LAN7500i [23]

La Raspberry Pi 3 B V1.2 tiene el chip LAN 9514-JZX, este también permite disponer de 4 puertos USB integrados en la placa.

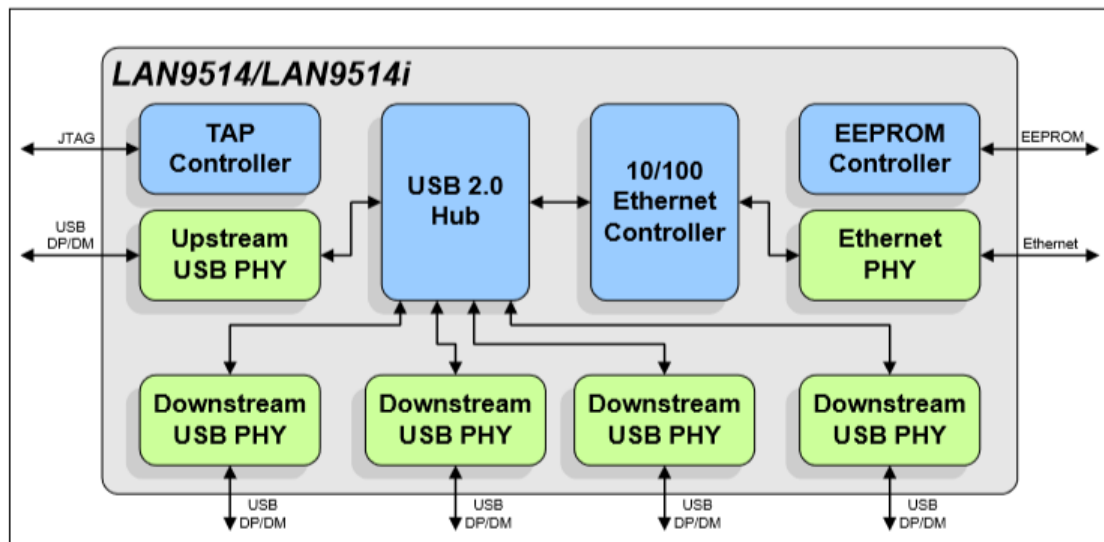


Ilustración 40 - LAN9514/LAN9514i [24]

5.1.1.2. Conector GPIO

GPIO se trata de un sistema de entradas y salidas de propósito general, dispone de un conjunto de pines que se pueden utilizar como entradas y salidas para múltiples usos. GPIO se incorpora en todos los modelos de Raspberry Pi, aunque hay algunas diferencias de unos modelos a otros.

El conector GPIO como todos los elementos que componen Raspberry han ido evolucionando con el paso del tiempo, Los modelos originales disponían de un conector GPIO de 26 pines, en cambio los modelos actuales de Raspberry Pi disponen actualmente de conectores de 40 pines. Sin embargo, hay que

comentar que la compatibilidad es total, debido a que dentro de los 40 pines nos encontramos con los 26 pines que existían en las Raspberry originales.

Pin#	NAME		NAME	Pin#
01	3.3v DC Power	●	DC Power 5v	02
03	GPIO02 (SDA1 , I2C)	●	DC Power 5v	04
05	GPIO03 (SCL1 , I2C)	●	Ground	06
07	GPIO04 (GPIO_GCLK)	●	(TXD0) GPIO14	08
09	Ground	●	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	●	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	●	Ground	14
15	GPIO22 (GPIO_GEN3)	●	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	●	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	●	Ground	20
21	GPIO09 (SPI_MISO)	●	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	●	(SPI_CE0_N) GPIO08	24
25	Ground	●	(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)	●	(I2C ID EEPROM) ID_SC	28
29	GPIO05	●	Ground	30
31	GPIO06	●	GPIO12	32
33	GPIO13	●	Ground	34
35	GPIO19	●	GPIO16	36
37	GPIO26	●	GPIO20	38
39	Ground	●	GPIO21	40

Ilustración 41 - Pines conector GPIO

Podemos decir que la mayoría de los pines son entradas/salidas digitales de propósito general. Estos se pueden configurar como entradas o salidas, esto da gran flexibilidad a la hora de trabajar con el conector GPIO. Hay que recordar que al tratarse de entradas/salidas digitales trabajan con unos y ceros. Hay que tener en cuenta que la tensión de funcionamiento de GPIO es de 3,3V, esto es tremendamente importante porque si se intentara trabajar con tensiones mayores a 3,3V el GPIO podría estropearse. Para complementar la información de los pines vamos a fijarnos en la imagen que tenemos a continuación, en ella podemos ver que algunos pines se pueden configurar conjuntamente. Vamos a explicar cuáles son sus funcionalidades de forma más detallada:

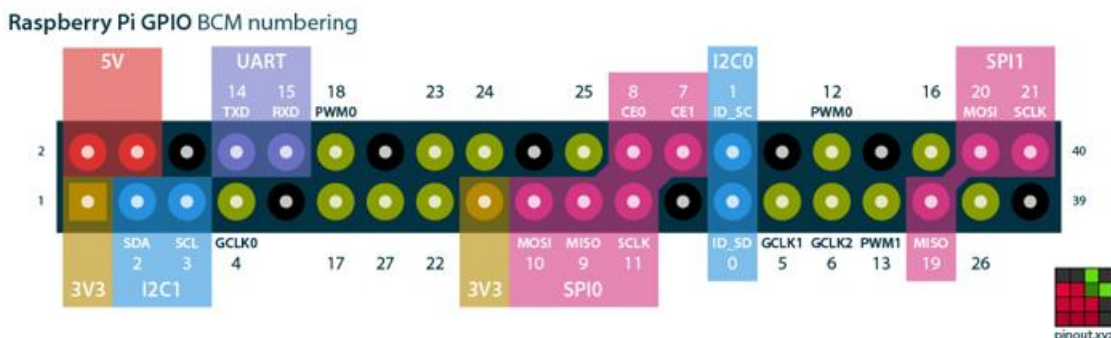


Ilustración 42 - Configuración pines GPIO

- Los pines 3 y 5 pueden ser configurados como una interfaz I2C, esto les permitiría trabajar con periféricos que funcionan bajo este protocolo.

- Los pines 8 y 10 se pueden configurar como interfaz UART para un puerto serie convencional. De hecho, Raspberry utiliza esta configuración por defecto, ya que la UART se utiliza como consola.
- El pin 12 puede ser configurado como salida PWM. Aunque los pines 12 y 13 pueden ser configurados como interfaz I2S, para ello hay que disponer de pines que en muchas ocasiones no estarán disponible.
- Los pines 19, 21, 23, 24 y 26 pueden ser configurados como la interfaz SPI0 para interactuar con periféricos que funcionen con este protocolo.
- Los pines 27 y 28 están reservados. Se utilizan para incorporar de forma opcional una memoria de expansión conforme a la especificación HAT. Estos pines se configuran inicialmente como salidas, todos los demás pines se configuran inicialmente como entradas para no tener problemas.
- Los pines 29, 31, 32, 33, 35, 36, 37, 38 y 40 son pines que no se encontraban en las Raspberry originales. Estas por lo tanto permiten acceder a nuevos usos. Los pines 32, 33 y 35 se pueden utilizar para salidas PWM. Además, estos pines adicionales pueden configurar otra interfaz SPI1.

[25] [26] [27]

5.1.1.3. Conector DSI

El puerto DSI permite la conexión de pantallas LCD a la GPU de la Raspberry pi [28]. Este se compone de 15 pines que permiten que se pueda conectar a la Raspberry Pi pantallas LCD externas. En este caso no se utiliza este conector puesto que se conecta a una pantalla mediante el HDMI.

N.º Pin	Función
1	Ground
2	Data Lane 1 N
3	Data Lane 1 P
4	Ground
5	Clock N
6	Clock P
7	Ground
8	Data Lane 0 N
9	Data Lane 0 P
10	Ground
11	
12	
13	Ground
14	3,3V
15	3,3V

Tabla 10 - Descripción puerto DSI

5.1.1.4. Conector CSI

Se trata de un conector de 15 pines, por el cual se puede conectar una cámara a la Raspberry Pi [29].

N.º Pin	Nombre	Propósito
1	Ground	Ground
2	CAM1_DN0	Data Lane 0
3	CAM1_DP0	
4	Ground	Ground
5	CAM1_DN1	Data Lane 1
6	CAM1_DP1	
7	Ground	Ground
8	CAM1_CM	MIPI Clock
9	CAM1_CP	
10	Ground	Ground
11	CAM_GPIO	
12	CAM_CLK	
13	SCL0	I2C Bus
14	SDA0	
15	3,3V	Power Supply

Tabla 11 - Conector CSI

5.1.1.5. Ranura para tarjeta microSD

La Raspberry Pi no dispone de disco duro, debido a esto esta dispone de una ranura para memorias microSD, digamos que la memoria microSD hace las funciones de disco duro. Lo que se hace es instalar el sistema operativo en la microSD, por lo que el arranque del sistema se hará desde la tarjeta, como esta tiene que albergar el sistema operativo debe de disponer de un tamaño suficiente. Las tarjetas deben de tener al menos 2GB, aunque debido a los precios actuales yo suelo trabajar con tarjetas de 32GB o de 64GB. Hay que recordar que también como complemento para guardar archivos se pueden utilizar memorias USB.

5.1.1.6. Puerto Ethernet

El puerto Ethernet de la Raspberry Pi se gestiona por el mismo integrado encargado de gestionar los USB. En la Raspberry Pi solo se dispone de un conector RJ-45, en mi caso las velocidades disponibles son diferentes porque utilizan integrados diferentes. Mientras que una solo proporciona conectividad a 10/100 Mbps, la otra proporciona velocidades de 10/100/1000 Mbps.

Este conector permite conectar la Raspberry a todo tipo de dispositivos Ethernet. Los modelos antiguos no disponían de Wi-Fi incorporado, pero los modelos actuales sí que incorporan la posibilidad de tener conexiones inalámbricas. Las velocidades inalámbricas dependerán del modelo utilizado.

5.1.1.7. Alimentación

La Raspberry Pi no dispone de botón de encendido y apagado, esto es principalmente para ahorrar en tamaño y costes. Para la alimentación de la placa se utiliza un conector micro USB tipo B. La tensión de alimentación es de 5V, las corrientes de funcionamiento dependen del modelo de placa. En mi caso utilizo cargadores de 5V y 2.4 A, en concreto se utiliza el cargador DURACELL modelo DRACUUSB2-EU, aunque para este trabajo una de las placas me ha dado problemas con la alimentación, aunque esto se detallara en apartados posteriores [30] [31].

N.º de Pin	Función
1	Vcc (+5V)
2	D-
3	D+
4	ID
5	GND

Tabla 12 - Pines del conector de alimentación

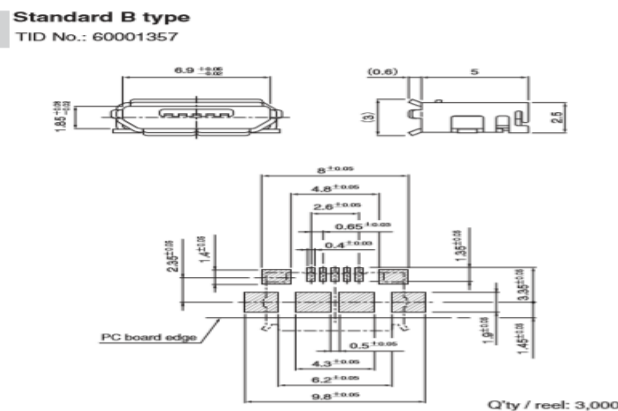


Ilustración 43- Conector de alimentación Raspberry Pi

5.1.1.8. LEDs indicadores

La Raspberry dispone de dos diodos LEDs, uno de los diodos es de color rojo y el otro de color amarillo. Estos diodos pueden ayudar a detectar posibles problemas de funcionamiento de nuestra Raspberry. Comenzaremos con el diodo rojo, este es el que determina si la alimentación de la placa es correcta o tiene problemas:

- El LED rojo no se enciende y no se ve nada por pantalla: La fuente de alimentación no está conectada correctamente o se ha estropeado. Si la fuente de alimentación funciona perfectamente nuestra Raspberry podría estar estropeada.
- El LED rojo parpadea: si el led rojo parpadea puede significar que la alimentación es de mala calidad o no está funcionando como debiera. En estos casos es conveniente cambiar la fuente de alimentación.

El diodo LED amarillo es el que nos muestra la actividad de la placa, en este caso el análisis se hace con el LED rojo encendido y el LED amarillo no se enciende:

- Problemas tarjeta SD/microSD: se puede producir cuando el sistema operativo no está bien instalado en la tarjeta o si la tarjeta esta estropeada. Para lo primero habría que formatear la tarjeta y volver a grabar el sistema operativo. Lo segundo solo se puede arreglar cambiando la tarjeta SD/microSD.
- Dispositivo conectado que interfiere en nuestro arranque: si el LED amarillo de actividad parpadea durante 20 segundos, significa que algo que este enchufado a nuestra Raspberry interfiere con nuestro arranque. Para poder conocer cuál es el problema se recomienda comprobar cada uno de los componentes conectados de forma independiente hasta detectar el elemento que interfiere en el arranque normal de la Raspberry Pi.
- Voltaje o corriente insuficiente: si la fuente de alimentación no proporciona la tensión o intensidad suficiente la Raspberry Pi no arrancará. Lo recomendable en estos casos es cambiar la fuente de alimentación.
- Error en archivo de sistema bootcode.bin: Este archivo puede generar problemas con algunas tarjetas SD/microSD. Para resolverlo se recomienda descargar un nuevo archivo [33].

El último análisis se realizará con el LED rojo encendido y el LED amarillo parpadeando con un patrón determinado. Esto se produce en las últimas actualizaciones de firmware:

- 1 parpadeo: en este caso lo más probable es que tengas una Raspberry de Micron. Para poder estar seguro se debería comprobar el procesador, si este tiene una M con una órbita alrededor en ella, para la resolución del problema es conveniente actualizar al software más reciente. Además, debes de comprobar que la tarjeta SD/MicroSD sea al menos de 4Gb.
- 3 parpadeos: esto se produce cuando el archivo start.elf no se encuentra.
- 4 parpadeos: el archivo start.elf no se puede arrancar es posible que este corrupto.
- 7 parpadeos: el archivo kernel.img no se encuentra.
- 8 parpadeos: no se reconoce la SDRAM. En este caso se recomienda cambiar a un nuevo firmware también es posible que la SDRAM esté dañada.

En las Raspberrys que tienen un firmware anterior al 20 de octubre de 2012, el significado de los parpadeas es algo diferente:

- 3 parpadeos: esto se produce cuando el archivo loader.bin no se encuentra.
- 4 parpadeos: el archivo loader.bin no se puede arrancar es posible que este corrupto.
- 5 parpadeos: archivo start.elf no se encuentra.
- 6 parpadeos: archivo start.elf no se puede arrancar.
- 7 parpadeos: el archivo kernel.img no se encuentra.

Esta es una pequeña ayuda para detectar algunos problemas de funcionamiento en la Raspberry pi [32].

5.1.1.9. Salidas de audio

La Raspberry Pi dispone de un Jack de 3,5mm para poder obtener el audio, además del HDMI. Si utilizamos el HDMI de la Raspberry Pi, este nos transmite el video y el audio juntos mediante un único cable. Si no se puede utilizar el HDMI entonces se puede realizar mediante el Jack de 3,5mm.

5.1.1.10. Conector HDMI

Se trata de un conector que permite la transmisión de video y audio, por ejemplo, a un monitor o TV.

5.1.1.11 SoC

En este caso las dos tarjetas utilizan diferentes modelos del BCM2837, las principales diferencias entre uno y otro se pueden ver en la tabla que muestra los diferentes modelos existentes y sus características.

5.1.2. Software

Las Raspberry Pi puede funcionar con diferentes sistemas operativos, el software a utilizar dependerá del uso que se quiera dar a la Raspberry Pi. La mayoría de los sistemas operativos están basados en distribuciones Linux, ya que este tipo de software es libre y no tiene ningún tipo de coste de licencia, estos permiten disponer sistemas operativos sencillos y ligeros. Es verdad que actualmente se dispone de algún sistema operativo basado en Windows, pero no su uso no se encuentra demasiado extendido.

En la tabla que se mostrara a continuación se citaran los sistemas operativo más usados, así como las aplicaciones utilizadas para facilitar la instalación de estos.

Proyecto	Fecha	Página Web	Categoría	Sistema operativo
Raspbian	13-nov.-2018	Raspberry Pi Downloads	Sistema Operativo	
Soft-float Debian "Wheezy"	16-ago.-2012	Raspberry Pi Downloads	Sistema Operativo	
Arch Linux ARM	13-ago.-2013	Raspberry Pi Downloads	Sistema Operativo	
RISC OS	10-jul.-2013	Raspberry Pi Downloads	Sistema Operativo	
FreeBSD	01-dic.-2018	FreeBSD	Sistema Operativo	
openSuse	03-dic.-2018	openSuse	Sistema Operativo	
Occidentalis	31-ago.-2012	Adafruit learning System	Sistema Operativo	
Pisces Image	10-jul.-2012	Pisces Images	Sistema Operativo	
Plan 9	30-nov.-2012	Bendyworks	Sistema Operativo	
Gentoo Linux	29-sep.-2012	Gentoo Linux	Sistema Operativo	
Hexxeh		Hexxeh Images	Sistema Operativo	
Puppi	16-feb.-2013	Puppy School Raspberry Pi	Sistema Operativo	
PiLFS	16-dic.-2012	PiLFS	Sistema Operativo	
Pidora FedoranRemix	09-ago.-2013	Fedora Remix	Sistema Operativo	
RaspBMC	02-oct.-2013	Raspberry Pi Downloads	Sistema Operativo	
Win32Disk Imager	04-jun.2013	Sourceforge	Instalador SO	
RMPrepUSB	26-jun.-2013	RMPrepUSB	Instalador SO	
BerryBoot	13-feb.-2013	BerryBoot	Instalador SO	
NOOBS 3.0	16-nov.-2016	NOOBS	Instalador SO	
PwnPi	14-dic.2012	PwnPi	Seguridad	
ha-pi	06-jul.-2012	ha-pi	Seguridad	
OSMC	Ago.-2013	OSMC	XBMC Media Center	


LibreELEC		LibreELEC	XBMC Media Center	
RaxPlex	21-Mar.-2017	RaxPlex	XBMC Media Center	
SqueezePlug on Max2Play		Max2Play	XBMC Media Center	
Pi MusicBox		Pi MusicBox	XBMC Media Center	
RPi-TC		RPi-TC	Dispositivo	
Raspbx	04-abr.-2018	Raspbx	VOIP	

Tabla 13 - Software Raspberry Pi

Una vez visto el listado de algunos sistemas operativos, es conveniente realizar una pequeña ampliación sobre los más relevante o utilizados por la Raspberry pi:

- **Raspbian OS:** esta distribución es una de las más importantes, debido a que es una de las distribuciones más completa y optimizada, de hecho, es la recomendada y por lo tanto la que cuenta con el apoyo oficial. Raspbian OS está basada en una distribución Debian, pero que se ha optimizado para ser usada en la Raspberry Pi. Esta permite gestionar el hardware de la Raspberry de forma rápida y sencilla, para ello incorpora herramientas, como el IDLE para Python, Scratch y otras muchas aplicaciones que se pueden descargar desde internet.
- **RISC OS:** este junto con el sistema operativo que está basado en Windows, son dos de los sistemas operativos que no están basados en Linus. RISC OS es un sistema desarrollado en Inglaterra por la empresa Acorn Computers y se sirve bajo licencia Open-Source. Este, aunque no dispone de un catálogo tan amplio de aplicaciones como otros sistemas operativos, también es considerado como uno de los sistemas oficiales de Raspberry Pi, esto es debido a que se desarrolló en torno a la plataforma ARM desde cero.
- **Arch Linux:** este sistema operativo se caracteriza por su simplicidad, elegancia, coherencia del código y minimalismo. Pero a pesar de todo esto Arch Linux es conocida por ser un entorno poco amigable y solo recomendable para personas con grandes conocimientos. Este se puede instalar en Raspberry Pi ya que soporta ARM.
- **Pidora:** este proviene de la distribución Linux Fedora, que está especialmente diseñada para funcionar en ARM. Respecto al resto de características son similares a su hermano mayor Fedora.
- **OpenELEC:** esta distribución al igual que LibreELEC, están pensadas para poder crear un centro multimedia con la Raspberry Pi. Con estas distribuciones podemos disponer de todo el contenido multimedia y

acceso a Internet para poder convertir nuestra TV en una smartTV. Kodi se creó inicialmente para la consola Xbox, pero el éxito conseguido hizo que el sistema se portara a otras plataformas. Dispone de elementos como el reproductor de audio, video, presentación de diapositivas, visores de imágenes, reportes de clima, IPTV y otras funciones que se pueden obtener mediante la instalación de plug-ins.

[34]

5.1.2.1. Instalación del Sistema Operativo

El sistema instalado en las Raspberrys Pi es Raspbian, este sistema es el más indicado para poder establecer las comunicaciones de las tarjetas mediante el bus CAN.

Para poder llevar a cabo la instalación se necesitan dos tarjetas microSD, en este caso se utiliza una de 32GB y otra de 64GB, además se necesita que el ordenador disponga de un lector de tarjetas SD. Como las tarjetas utilizadas son microSD y el ordenador dispone de una ranura SD se requiere de un adaptador.



Ilustración 44 - Ranura SD y Tarjeta microSD con adaptador

Lo primero que hacemos es descargar la imagen de la distribución elegida, en este caso Raspbian. En segundo lugar, se descarga el software que permite la grabación de la imagen, en este caso se utilizó Win32 Disk Imager. Una vez instalado en el ordenador pasamos a instalar el software en la microSD. Para instalar la distribución lo primero es formatear la tarjeta microSD en segundo lugar, se instala la imagen con el programa Win32 Disk Imager.

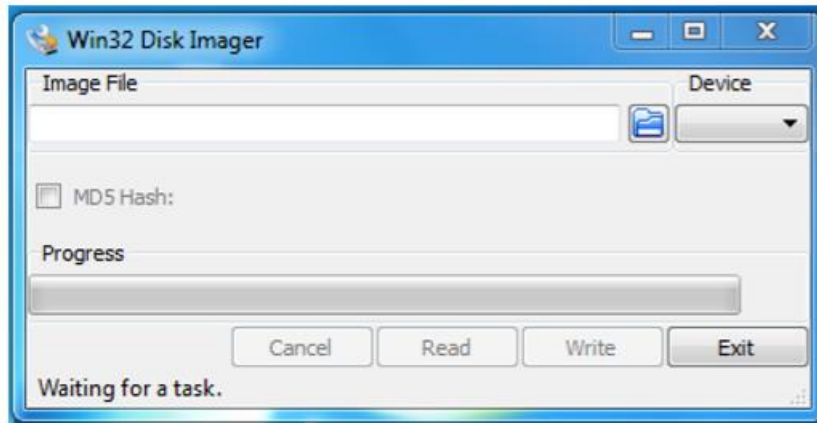


Ilustración 45 - Imagen Win32 Disk Imager

Para ello con el programa Image File se escoge el fichero que contiene la imagen del Sistema Operativo, mediante Device se escoge la unidad asociada a la tarjeta microSD y mediante Write se pasa el sistema a la tarjeta.

Una vez finalizado el proceso ya tenemos una tarjeta con el Sistema Operativo instalado, solo tenemos que insertarla en la ranura microSD de la Raspberry Pi y empezar a trabajar.

5.2. Nodo CAN

Durante los siguientes apartados se expondrán todos los pasos dados para llevar a cabo todo el proceso de realización del PCB de un nodo CAN. Se tratará de describir pormenorizadamente el software utilizado, la elección de componentes y por último el proceso de fabricación.

5.2.1. Software utilizado

Para la realización del esquema del circuito como del diseño PCB del nodo se ha utilizado el software KiCad. Este programa es un paquete de software libre utilizado para la automatización de diseño electrónico. Este permite el diseño de esquemas de circuitos electrónicos que posteriormente pueden ser convertidos en PCB.

Dentro de KiCad se dispone de las herramientas necesarias para crear una lista de componentes, ilustraciones, archivos Gerber y vistas 3d de la PCB y sus componentes.

KiCad se organiza en cinco partes:

- KiCad este es el administrador de proyectos
- Eeschema se trataría del editor de esquemas
- Pcbnew entorno que permite el desarrollo de los circuitos impresos

- Gerbview es el visualizador de archivos Gerber.
- Bitmap2Component herramienta que permite convertir imágenes a footprints para realizar “PCB artwork”.

KiCad es un programa multiplataforma, desarrollado en C++ con wxWidgets que puede ejecutarse en FreeBSD, Linux, Microsoft, Windows y Mac OS X. Es un programa que dispone de un amplio listado de componentes, además de permitir la agregación de componentes de forma personalizada. Los componentes personalizados pueden estar disponibles solamente para un proyecto concreto o para su uso en cualquier proyecto. También permite incorporar componentes de otras aplicaciones EDA. Los archivos de configuración están en texto plano, esto permite interconectar los sistemas de control de versiones, además de los scripts de generación de componentes automatizados.

Eeschema permite el uso de hojas de esquemas jerárquicas, la creación de símbolos personalizados. Los símbolos esquemáticos a su vez están asociados a huellas de Pcbnew. Se pueden elegir las huellas más adecuadas a los componentes, de esta forma no habrá problemas a la hora de colocarlos en placa.

Pcbnew dispone de hasta 32 capas de cobre y 32 capas técnicas, en este caso solo se necesita una capa de cobre [34] [35] [36].

5.2.2. Esquemático

En el esquemático del circuito se realizan todas las conexiones de los componentes del circuito, a partir del esquema se podrá generar el PCB. La herramienta utilizada es Eeschema.

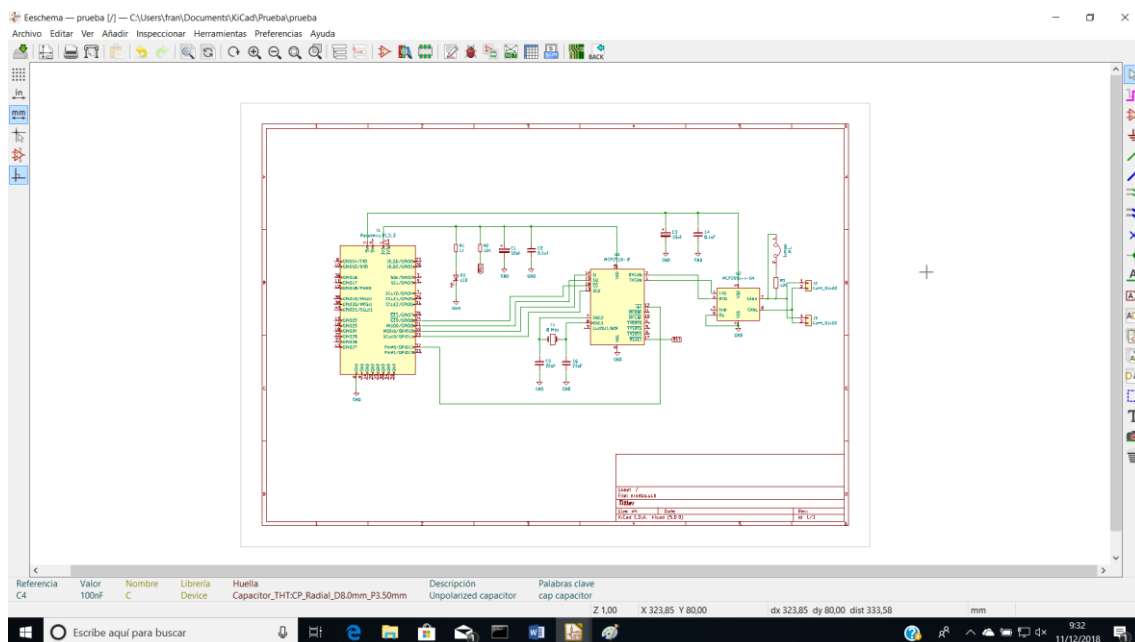


Ilustración 46 - Captura herramienta Eeschema

Para la realización del esquemático se ha tenido que realizar el componente MCP2515-IP que no estaba disponible en las librerías del programa. Una vez que se han colocado los componentes y se han conectado entre ellos, se eligen las huellas más adecuadas para realizar el PCB. Una vez realizado todo esto se pasa a sacar el netlist que permitirá realizar el PCB.

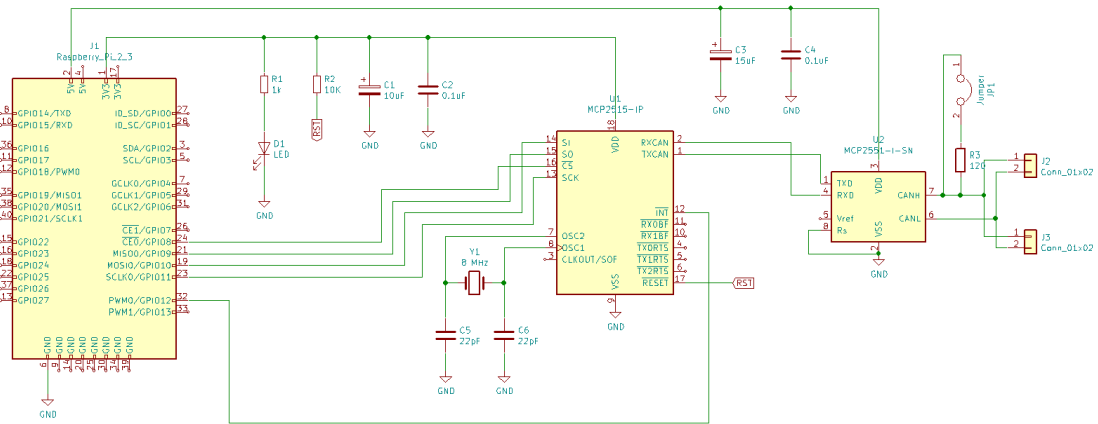


Ilustración 47 - Esquema utilizado para obtener PCB [37]

Muchos de los componentes utilizados no son los más adecuados, pero se decidió utilizarlos para reducir los costes de fabricación, por ejemplo, los conectores J2 y J3 se debería de cambiar por dos conectores DB9. Además, en dicho conector sería recomendable incorporar una masa de referencia y alimentaciones adicionales poder alimentar otros nodos por el bus si fuese necesario.

Se utilizan dos conectores J2 y J3 para poder interconectar varios nodos al bus de forma sencilla. El jumper JP1 permite añadir al circuito la resistencia de 120Ω o eliminarla dependiendo si nos encontramos ante un nodo final o un nodo intermedio.

Otros elementos que se podrían incorporar al diseño son jumpers en las alimentaciones de 3,3V y 5V, sumados a la incorporación de un conector para obtener la alimentación de forma externa en lugar de la Raspberry Pi [35].

5.2.3. Listado de componentes

El listado de componentes nos lo proporciona KiCad a partir del esquema creado. Hay que recordar que las placas a fabricar son dos, por lo que el número de componentes que hay que adquirir sería del doble de los que se disponen en el listado. Debido a que este no se puede ver con claridad en la imagen proporcionada se incorpora una tabla con los datos más relevantes.

El componente MCP2551 tuvo que ser creado puesto ya que no se disponía del modelo concreto, el chip disponible en KiCad no coincidía en el número de entradas/salidas. Por lo que se creó un componente y se le asignó una huella adecuada para él.

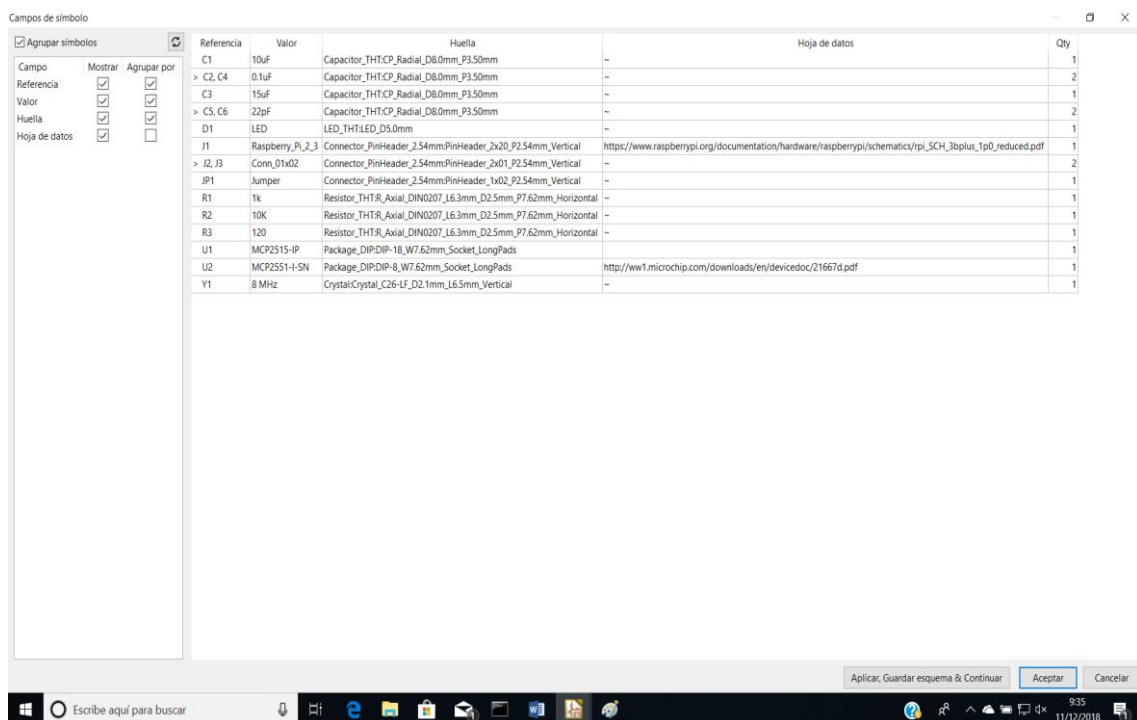


Ilustración 48 - Captura componentes esquema

Referencia	Valor	Cantidad
C1	10µF	1
C2, C4	0.1µF	2
C3	15µF	1
C5, C6	22pF	2
D1	LED	1
J1	Raspberry_Pi_2_3	1
J2, J3	Conn_01x02	2
JP1	Jumper	1
R1	1KΩ	1
R2	10KΩ	1
R3	120Ω	1
U1	MCP2515-IP	1
U2	MCP2551-I-SN	1
Y1	8MHz	1

Tabla 14 - Lista de componentes PCBs

5.2.3.1. MCP2515-I/P

El MCP 2515 es el microcontrolador encargado de implementar las especificaciones del protocolo CAN, este puede transmitir tanto tramas estándar como tramas extendidas. La comunicación se realiza mediante el protocolo serie SPI.

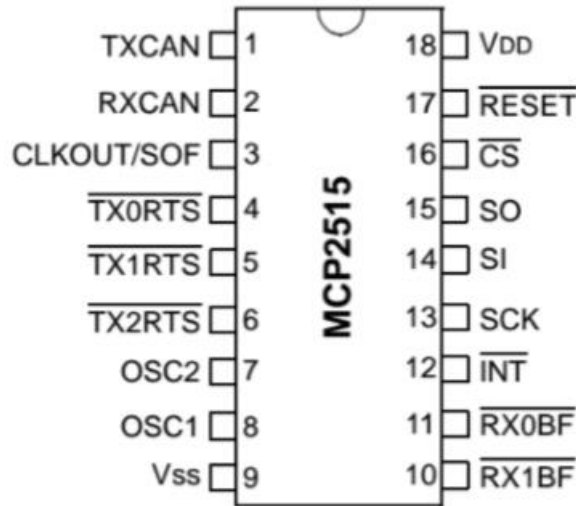


Ilustración 49 - Chip MCP2515

El encapsulado utilizado es el PDIP de 18 pines, debido a que la implementación física del circuito se ha realizado sobre una placa de circuito impreso. El circuito dispone de las entradas de alimentación VSS y VDD, además dispone de las entradas OSC1 y OSC2 que son utilizadas para introducir una señal de reloj. El diagrama de bloques del chip es el siguiente.

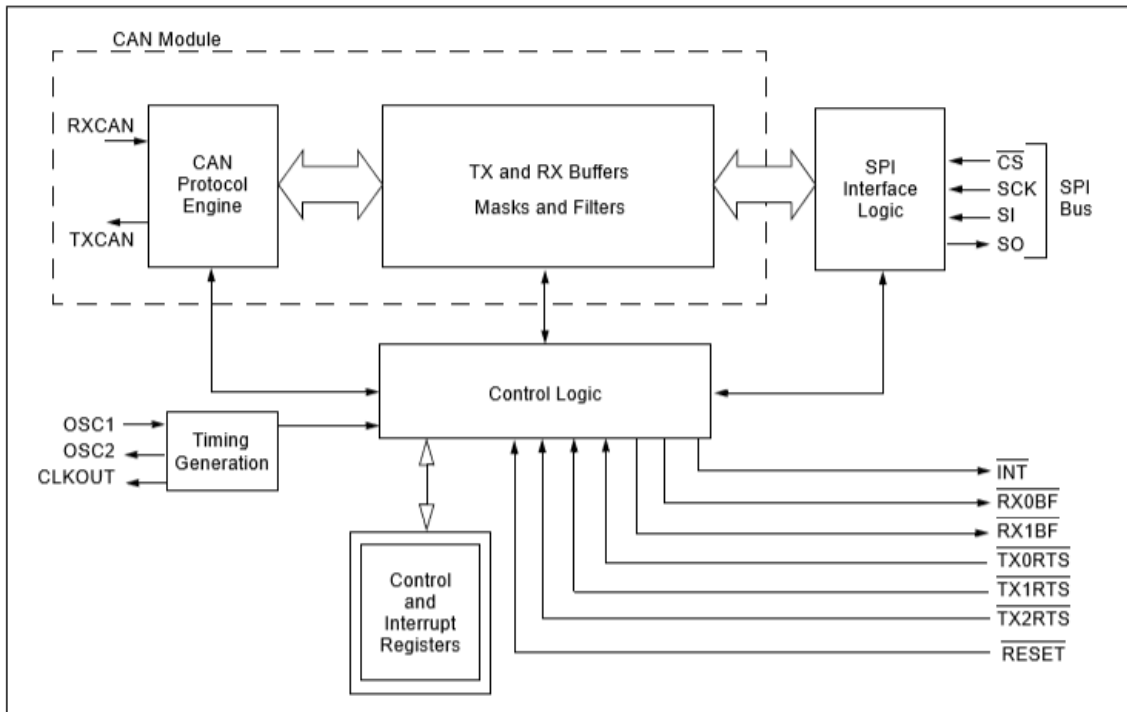


Ilustración 50 - Diagrama de bloques chip MCP2515

Este integrado también dispone de cuatro entradas específicas para la comunicación mediante SPI (pines CS, SO, SI y SCK), dos son los pines utilizados para comunicarse con el transceptor CAN (TXCAN y RXCAN), tiene

una entrada de RESET y una salida específica para las interrupciones, esta se activa para comunicar que el chip ha recibido una trama CAN [38].

5.2.3.2. MCP2551-I/P

MCP2551 es el transceptor y se encarga de controlar la parte física en las comunicaciones mediante el protocolo CAN. Principalmente se encarga de controlar los niveles de tensión y la gestión de la codificación de los bits en el medio. La velocidad máxima de operación es de 1MB/s y soporta un máximo de 112 nodos de forma simultánea en la red.

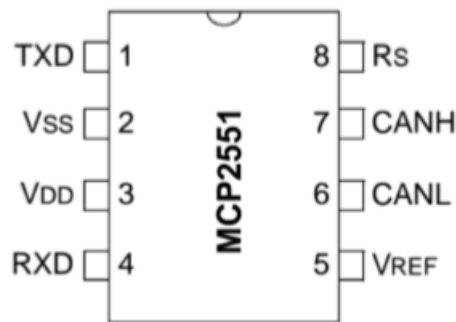


Ilustración 51 - Chip MCP2551

El encapsulado utilizado es un PDIP de 8 pines, la implementación física se ha realizado sobre una placa de circuito impreso. Este dispositivo dispone de 2 patillas, RXD y TXD, estas se utilizan para comunicarse con el módulo MCP2515, y de otras dos patillas CAN_H y CAN_L que se encargan de la comunicación en el bus, las patillas de alimentación son VSS y VDD [39].

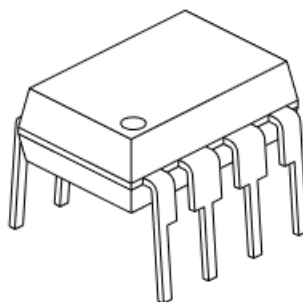


Ilustración 52 - Chip MCP2551

5.2.4. Diseño PCB

El PCB se realiza en el entorno Pcbnew, este entorno permite el diseño de la PCB que se fabricará posteriormente. En primer lugar, se colocan todos los componentes, una vez colocados los componentes se empiezan a unir mediante las pistas. Es verdad que a lo largo de la colocación de las pistas que unen los componentes, se ha requerido del cambio de posicionamiento de algunos de estos componentes por problemas a la hora de colocar las pistas.

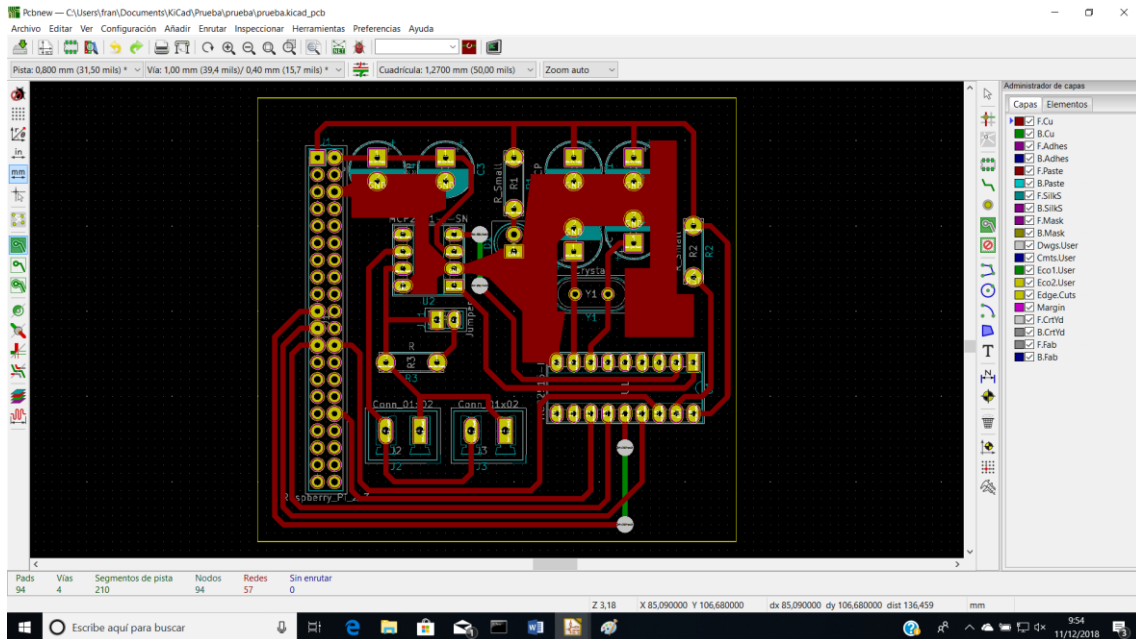


Ilustración 53 - Captura Pcbnew

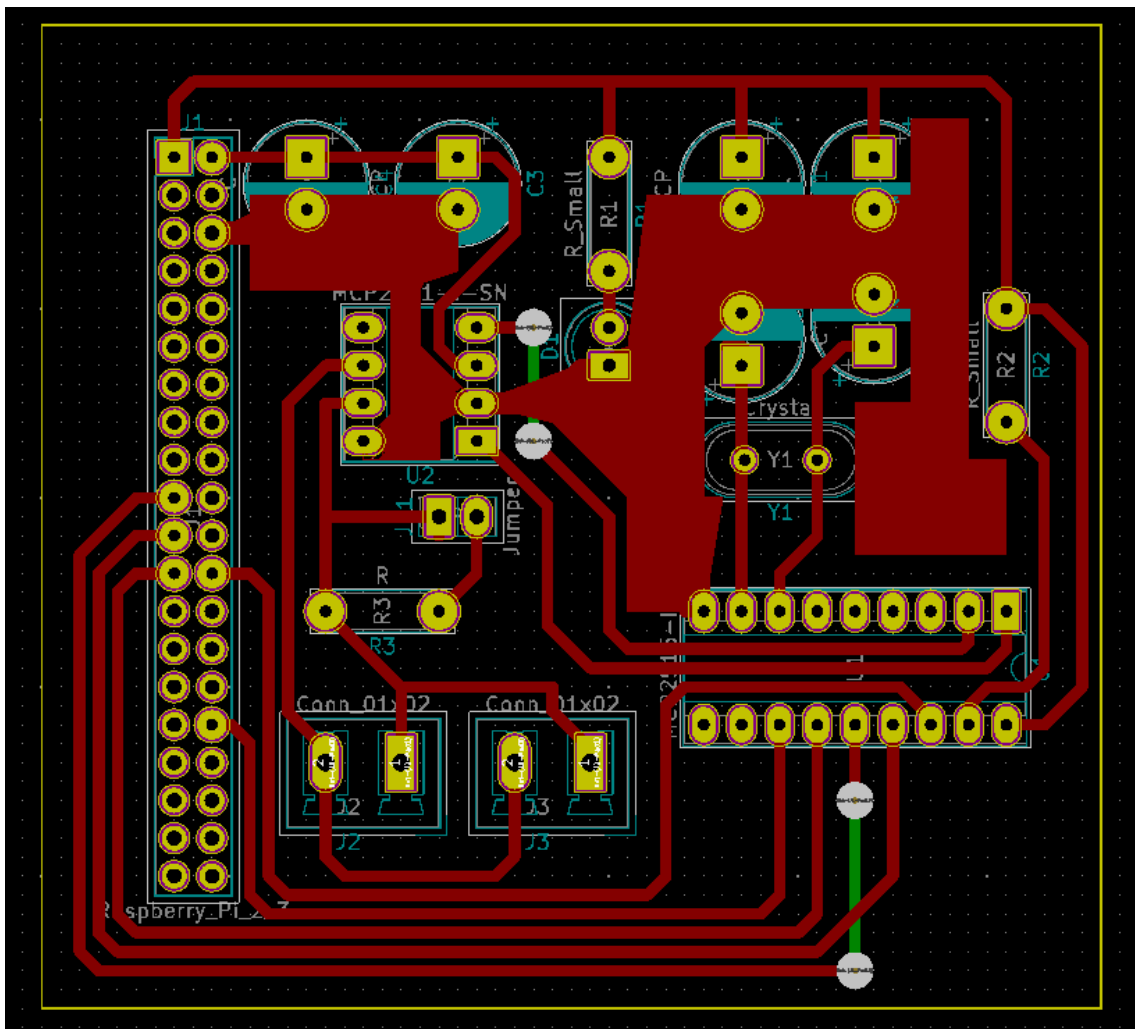


Ilustración 54 - Placa de circuito impreso

Hay programas que pueden realizar algunas de estas funciones de forma automática, pero en este caso se ha hecho todo de forma manual. En futuros prototipos sería recomendable eliminar los dos puentes que se han tenido que colocar. Por otro lado, también se recomienda la mejora del plano de masa que en este programa tampoco es sencillo de colocar. En la siguiente imagen se puede ver solo la capa de cobre.

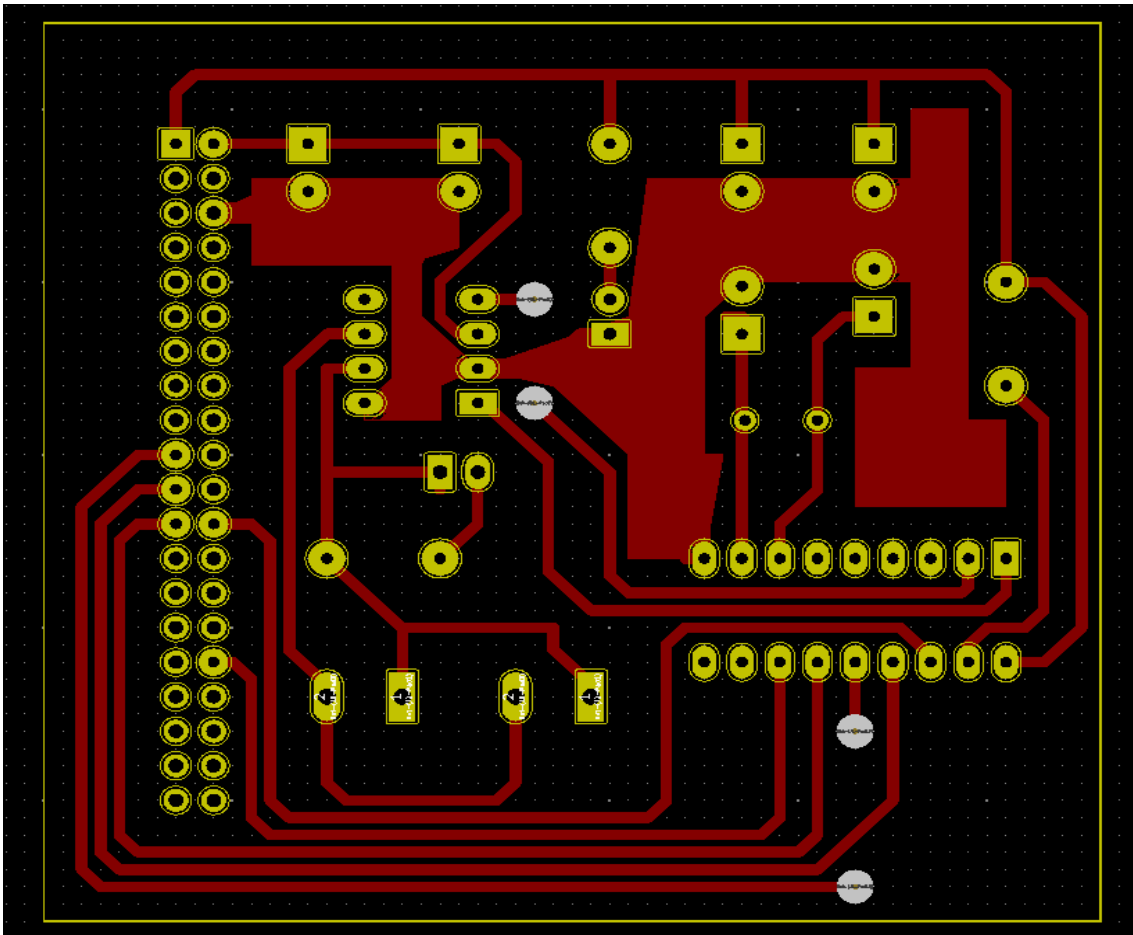


Ilustración 55 - Lado de pistas de PCB

En la siguiente en cambio se verá el lado de componentes, es verdad que los puentes quedan dibujados como si se estuviera trabajando con una placa de doble cara, pero cómo no se disponen de herramientas para poder fabricar en doble capa se sueldan dos puentes. También se ha incorporado la capa de silks, para poder ver mejor los componentes colocados en la placa.

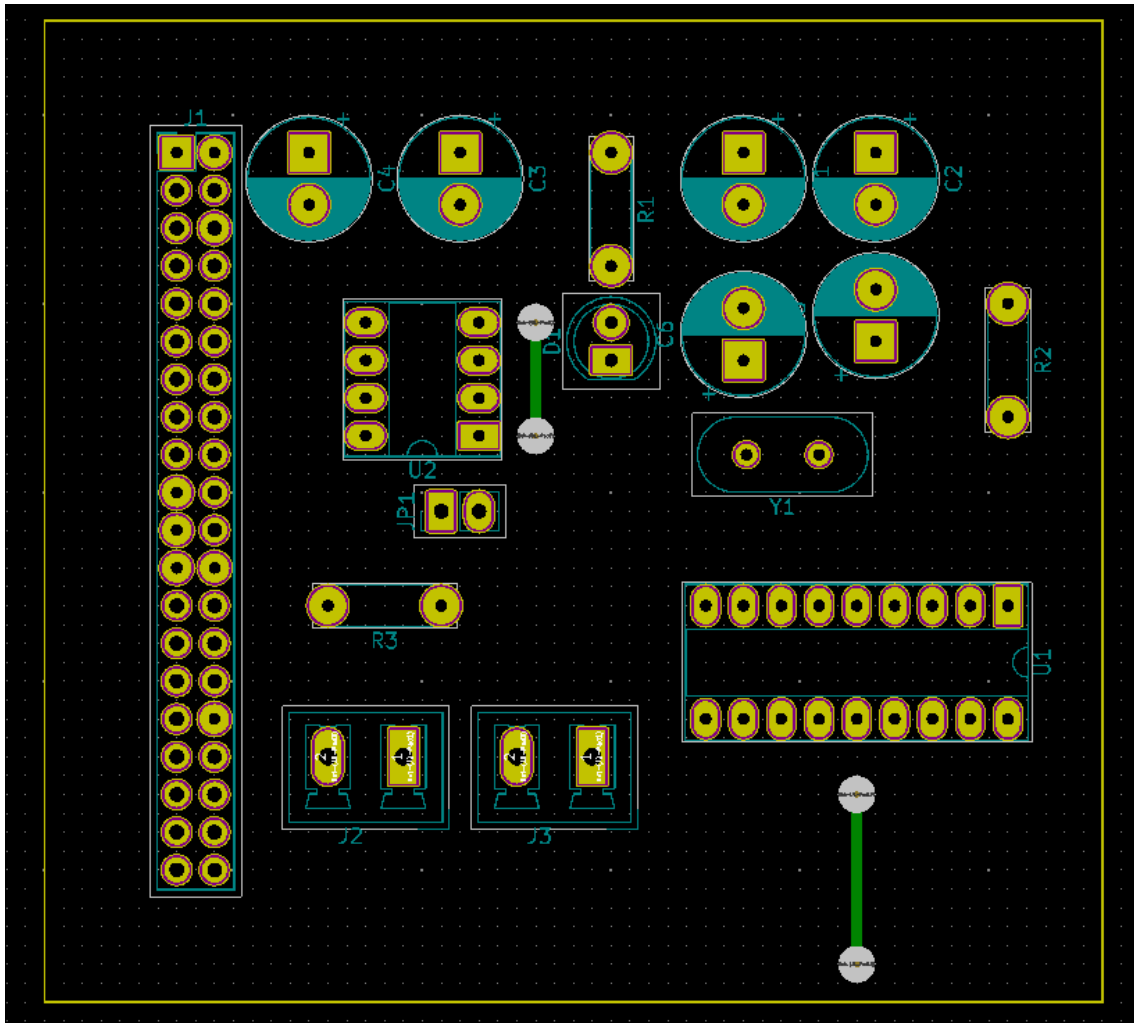


Ilustración 56 - Lado de componentes de PCB

5.2.5. Proceso de fabricación

La fabricación del prototipo es de gran dificultad, puesto que no se disponen de las herramientas más apropiadas y esto hace que se convierta en una difícil tarea. La placa se ha fabricado mediante el método del planchado. Para empezar lo primero que tenemos que hacer es imprimir la cara de cobre sobre un papel fotográfico, en concreto se ha utilizado Photo Paper de Kodak. La impresión de la capa de cobre se tiene que realizar mediante una impresora láser.

En el siguiente paso se coge la placa de cobre a la que se le quiere transferir el documento impreso, y se procede a la limpieza exhaustiva del cobre de la PCB. Esto se llevó a cabo mediante una lija de agua y una posterior limpieza con acetona, esto se realiza para eliminar cualquier rastro de suciedad que dificulte que el tóner se adhiera a la placa.

Inmediatamente a continuación que la placa este bien limpia, se recorta el papel donde este impreso el circuito y se coloca sobre la placa. La intención es que el tóner que esta adherido al papel se traslade a la capa de cobre de la placa. En esta ocasión se realizó un precalentamiento de la placa algo que dificulto la

colocación del papel, pero que parece que facilita el traslado del tóner del papel al cobre.

Inmediatamente a continuación de que el papel está bien colocado sobre el cobre, se procede a planchar el circuito con la plancha. El tiempo de planchado puede variar dependiendo de la potencia de la plancha. En este caso se ha realizado el planchado durante unos 10 minutos.

Justo después se procede a retirar el papel que quedará totalmente adherido al cobre. Esto se facilita mediante el uso de agua, es recomendable dejar de enfriar algo la placa y también es conveniente el uso de agua caliente. Si no se hiciera de esta forma se podrían generar burbujas en el papel que levantasen el tóner. En caso de los prototipos la calidad obtenida no fue demasiado buena lo que obliga a reparar las pistas con rotulador.

Inmediatamente después se procede al atacado de la placa, este proceso se realiza mediante la mezcla de agua fuerte con agua oxigenada de 110 volúmenes y es conveniente rebajar la mezcla con agua.

Justo a continuación se procede al cortado y taladrado de la placa. Por último, se sueldan todos los componentes a la misma. El resultado obtenido se puede ver en las siguientes imágenes.

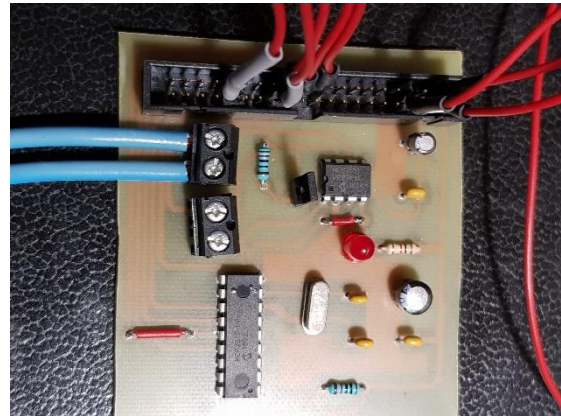
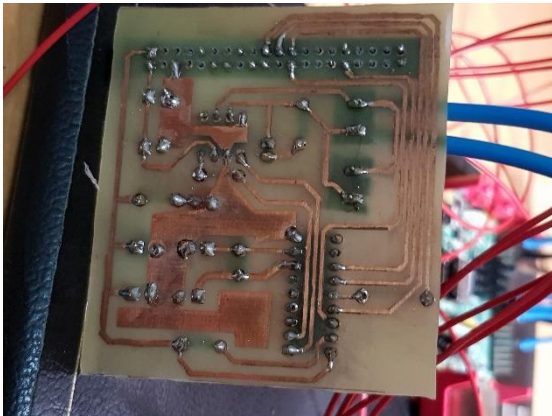


Ilustración 57 - Circuito impreso uno ya montado

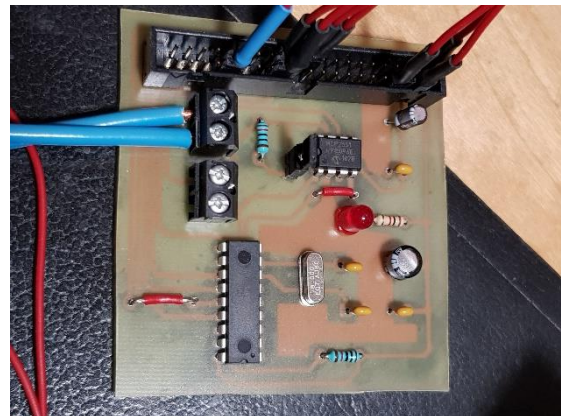
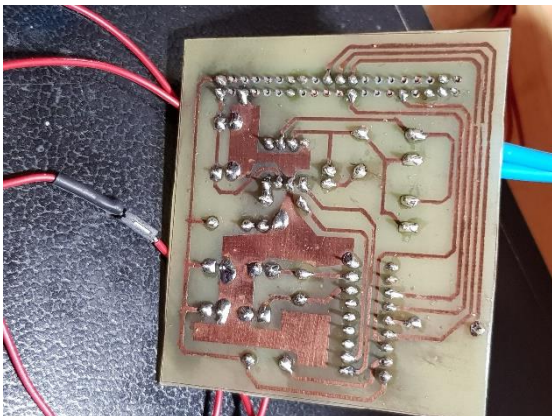


Ilustración 58 - Circuito impreso dos ya montado

Como se puede comprobar la calidad de las pistas no es tan buena como si se hubieran fabricado con métodos profesionales, pero son placas totalmente funcionales como se verá más adelante en el apartado de pruebas.

5.2.6. Configuración del Software

Una vez que el Sistema Operativo está instalado debemos configurarlo para que se pueda establecer la comunicación entre las dos Raspberrys. Lo primero que se debe de hacer es asegurarse que se disponen de las últimas versiones de los paquetes, esto se comprobara mediante las siguientes líneas.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo reboot
```

Después de realizar la actualización del sistema debemos de activar la interfaz SPI.

```
cd /usr/bin
```

```
sudo ./raspi-config
```

Estas líneas abren un menú en el que hay que buscar la interfaz SPI y activarla. Después de estos pasos es conveniente reiniciar la Raspberry. Una vez arrancada de nuevo tenemos que configurar la interfaz que se utilizará para establecer la comunicación CAN.

```
sudo nano /boot/config.txt
```

Se añaden justo después de la línea **dtoverlay=spi=on** las siguientes líneas:

```
dtoverlay=mcp2515-can0,oscillator=8000000,interrupt=12  
dtoverlay=spi-bcm2835-overlay
```

El dato del oscilador depende del reloj que se coloca en las placas, que en este caso es de 8MHz mientras que para la interrupción utilizamos la 12, que depende también de la entrada/salida que se utilice en la Raspberry.

```
sudo apt-get install can-utils
```

Una vez realizada esta configuración se puede reiniciar. Inmediatamente después del reinicio configuramos la interfaz CAN.

```
sudo ip link set can0 up type can bitrate 125000
```

[37]

5.2.7. Prueba de funcionamiento

Una vez montadas las placas y configurado el software, pasamos a conectar todas las placas para iniciar las pruebas de la comunicación. Una vez conectados todos los elementos nos percatamos que una de las Raspberry Pi no arranca, después de varias comprobaciones se llega a la conclusión de que la fuente de alimentación elegida no genera la intensidad suficiente para alimentar la Raspberry Pi y la placa CAN.

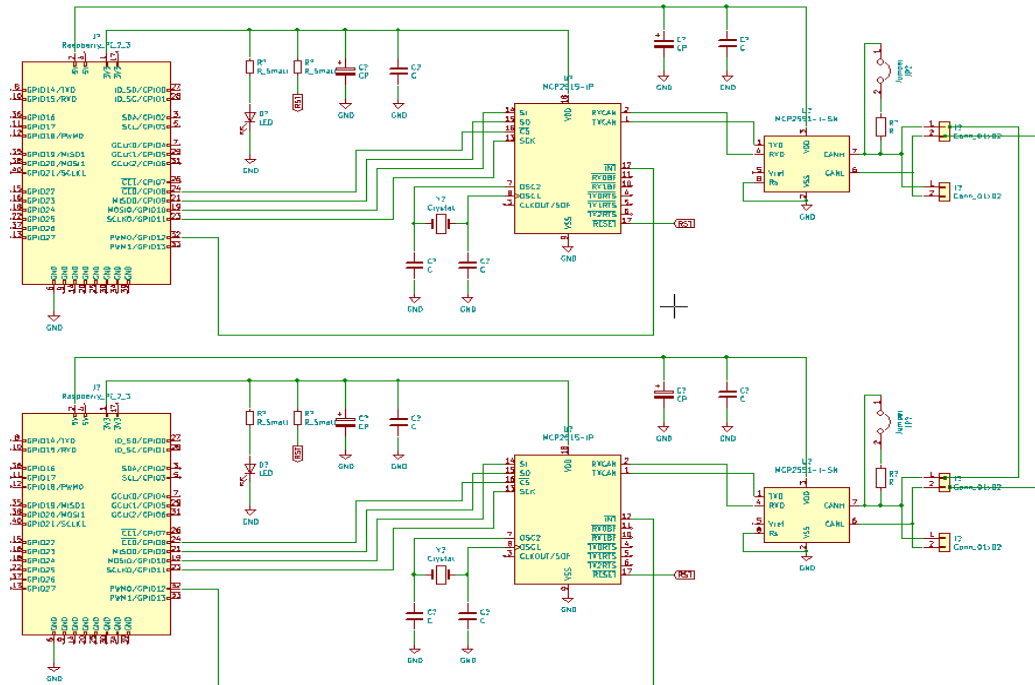


Ilustración 59 - Esquema de conexiones de placas CAN

Debido a esto se ha tenido que incorporar una fuente de alimentación externa para alimentar la placa de CAN. Una vez resuelto el problema de alimentación configuramos una placa para enviar datos y la otra en modo de recepción.

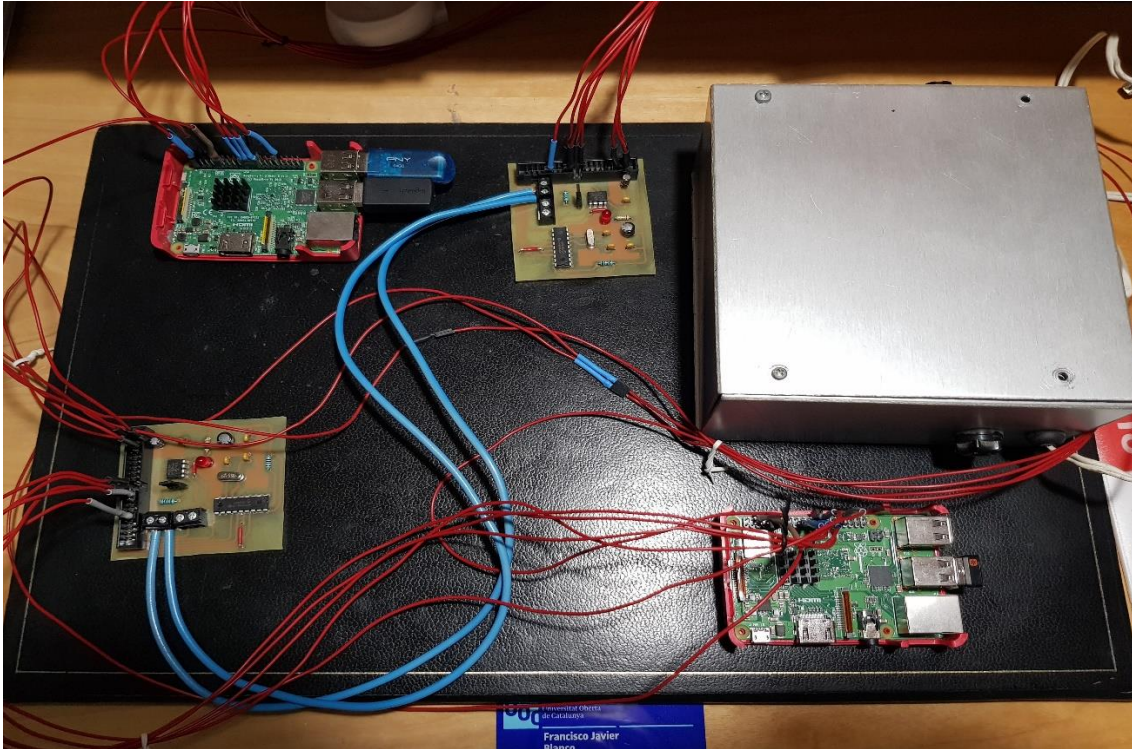


Ilustración 60 - Resultado de la conexión de todas las placas

En la imagen se puede ver como se envían tramas desde la Raspberry Pi mediante cansend.

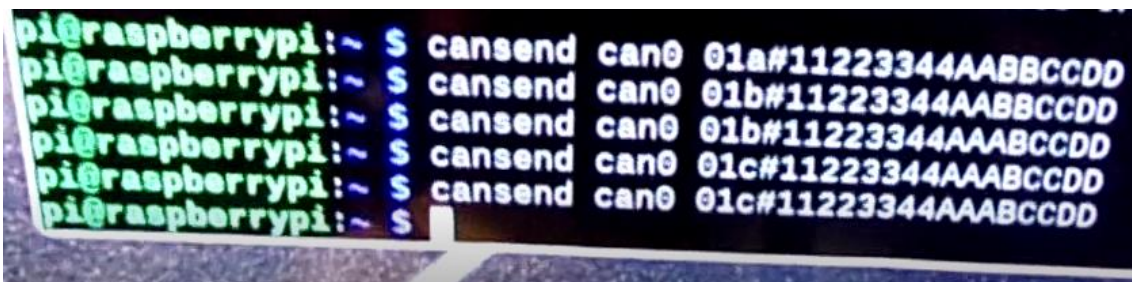


Ilustración 61 - Envío de tramas en una Raspberry Pi

Mientras en la otra Raspberry Pi se pone en modo de recepción mediante candump, en la imagen se puede ver la recepción de las tramas. Hay algunos símbolos en la trama 3 y 4, esto es debido a pulsaciones de teclado inadecuadas ya que se trabaja con una única pantalla y dos teclados.

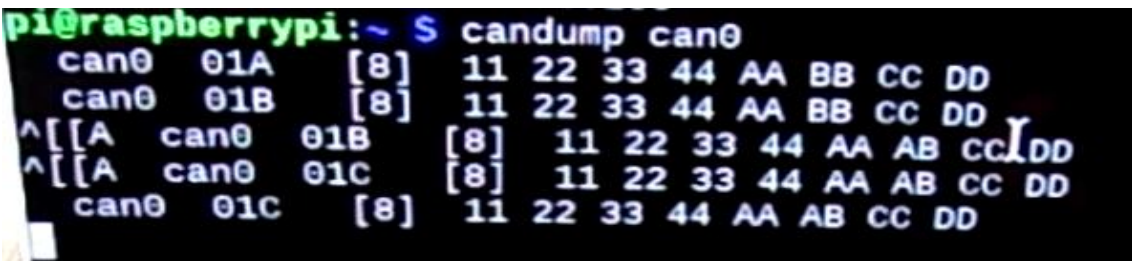


Ilustración 62 - Recepción de las tramas en la Raspberry Pi

6. Presupuesto

En la tabla que se puede ver a continuación se incorporan el precio de todos los componentes necesarios para la implementación de los dos módulos CAN necesarios para establecer la comunicación entre las dos Raspberry Pi. En el presupuesto desarrollado no se incorporan los costes derivados del diseño de la PCB, ni de la mano de obra necesaria para el montaje de esta.

Descripción	Cantidad	Importe unitario	Importe
CAN Controller SPI data filtering DIP18. MCP2515-I/P	2	1,81 €	3,62 €
High Speed CAN Transceiver, MCP2551-I/P	2	1,02 €	2,04 €
Resistencia de carbón, 1K Ω , 0,25W, 5%	2	0,13 €	0,26 €
LR Metal Axial Film Resistor 10K Ω , 0.6W	2	0,05 €	0,10 €
LR Metal Axial Resistor 120 Ω , 0,6W	2	0,06 €	0,12 €
Condensador electro Al radial KS, 10 μ F 50V	2	0,21 €	0,42 €
Condensador Radial PW series 15 μ F 16V	2	0,23 €	0,26 €
Condensador cerámico 22pF	4	0,30 €	1,20 €
Condensador cerámico 0,1 μ F 50V	4	0,23 €	0,46 €
Conector, ribbon cable, straight 40 way	2	2,55 €	5,10 €
LED difuso rojo, L-53HD 25mA 5mm	2	0,23 €	0,46 €
Crystal 8 MHz HC49S	2	1,57 €	3,14 €
Puente bipolar negro, extracción manual	2	0,39 €	0,78 €
Raw epoxy FR4, 8/10 single 100x160mm	2	3,97 €	7,94 €
Terminal para PCB negro 2 vías, 5,08mm	4	0,52 €	1,04 €
Total			26,94 €
IVA 21%			5,66 €
Total con IVA			32,6 €

Tabla 15 - Presupuesto

El precio de los costos producidos por las Raspberrys Pi y sus fuentes de alimentación se han llevado a cabo en una tabla independiente.

Descripción	Cantidad	Importe unitario	Importe
Raspberry Pi	2	29,47 €	58,94 €
Cargador Duracell Modelo: DRACUSB2-EU	2	11,70 €	23,4 €
Total			82,34 €
IVA 21%			17,30 €
Total con IVA			99,64 €

Tabla 16 – Precio de las Raspberrys Pi y los cargadores

7. Conclusiones

La realización de este proyecto me ha permitido adentrarme en el mundo de los buses de campo y de sus posibilidades, en particular sobre el bus de comunicaciones CAN. El bus de comunicaciones CAN se encuentra en un nivel de madurez e implantación extraordinario, hoy en día se utiliza en cientos de miles de aplicaciones, los fabricantes de componentes electrónicos están incorporando controladores CAN de forma generalizada a sus componentes. Los controladores CAN se pueden incorporar en ASICs y en dispositivos de lógica programable (FPGAs). Podemos decir que a tenor de lo aprendido en el estudio CAN resulta una opción muy para tener en cuenta en sistemas distribuidos de tiempo real.

Con respecto a los objetivos marcados al principio del proyecto es posible que no se hayan podido cumplir en su totalidad, esto se debe principalmente a la ingente cantidad de documentación encontrada con respecto al bus CAN. Mucha de la cual es a veces contradictoria lo que lleva a generar confusión. Aun así, espero que el estudio realizado ayude a mejorar los conocimientos de los lectores sobre el bus estudiado en el trabajo, además de promocionar su uso que era otro de los objetivos buscados.

La planificación seguida ha sufrido algunas modificaciones a lo largo de la realización del trabajo, estos se han producido para garantizar el éxito del trabajo. El principal cambio en la planificación es el montaje de los nodos de ejemplo, cuya misión es facilitar la comprensión del funcionamiento de los nodos de CAN Bus.

Como líneas de trabajo futuro se podría realizar un estudio más amplio de los buses de campo, pudiendo de esta manera comparar el Bus CAN con otros buses de campo de tiempo real. Además, a partir del año 2011 Bosch en cooperación con otras empresas inició el desarrollo de la nueva generación del bus CAN, esta nueva generación se denomina protocolo CAN FD. Por lo que se recomienda realizar un estudio sobre esta evolución de CAN que se encuentra estandarizada en la norma ISO/DIS 11898-1.

Con respecto al prototipo se recomienda realizar mejoras en la alimentación, conexionado, masa, etc.

8. Glosario

ABS: Antilock Brake System

ACK: Acknowledgment

CAL: Application Layer

CAN: Controller Area Network

CD: Collision Detection

CENELE: Comité Europeo de normalización Electrotécnica

CiA: CAN in Automation

CR: Collision Resolution

CRC: Cyclic Redundancy Check

CSI: Camera Serial Interface

CSMA: Carrier Sense Multiple Access

DB: Diffusion Box

DLC: Data Length

DS: Disconnectable Subscriber

DSI: Display Serial Interface

EDA: Electronic Design Automation

EN: Normas Europeas

EOF: End Of Frame

FMA: Fieldbus Management

GmbH: Gesellschaft mit beschränkter Haftung (sociedad con responsabilidad limitada)

GPIO: General Purpose Input/Output

HAT: Hardware Attached on Top

IDE: Identifier Extension

IEC: International Electrotechnical Commission

IFS: Inter-Frame Space

I2C: Inter-Integrated Circuit

I2S: Inter-IC Sound

JB: Junction Box

LED: Light-Emitting Diode

LLC: Logic Link Control

LSDU: Link Service Data Unit

MAC: Medium Access Channel

microSD: micro Secure Digital

NDS: Non-locally Disconnectable Subscriber
NRZ: Non Return to Zero

OSI: Open System Interconnection

PCB: Printed Circuit Board

PLC: Programmable Logic Controller)

PMS: Philips Medical System

PROFIBUS-DP: Distributed Peripheral

PROFIBUS-FMS: Fieldbus Message Spectification

PROFIBUS-PA: Process Automation

PWM: Pulse Width Modulation

QoS: Quality os Service

REP: Repeater

RTR: Remote Transmisión Request

RZ: Return to Zero

SD: Secure Digital

SDRAM: Double Data Rate Synchronous

SJW: Synchronization Jump Width

SoC: System on Chip

SOF: Start of Frame

SPI: Serial Pheripheral Interface

TQ: Time Quantum

UART: Universal Asynchronous Receiver-Transmitter

WBS: Work Breakdown Structure

WorldFIP: Factory Instrumentation Protocol

9. Bibliografía

[1] Red controladora de área (CAN): Protocolo de comunicación serial distribuida. [En línea]. Disponible en:

<https://www.uaeh.edu.mx/docencia/Tesis/icbi/licenciatura/documentos/Red%20controladora%20de%20area%20CAN.pdf>

[Accedido: 22-sep-2018]

[2] Estudio del Protocolo de Comunicación serial BUS CAN y la aplicación en la industria de vehículos de transporte BUSSCAR DE COLOMBIA. [En línea]. Disponible en:

<http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/9251/T004.62%20B932.pdf?sequence=1&isAllowed=y>

[Accedido: 22-sep-2018]

[3] Modelo OSI. [En línea]. Disponible en:

https://es.wikipedia.org/wiki/Modelo_OSI

[Accedido: 23-sep-2018]

[4] El modelo OSI. [En línea]. Disponible en:

<https://www.monografias.com/trabajos107/elmodelo-osi/elmodelo-osi.shtml>

[Accedido: 23-sep-2018]

[5] Modos de transmisión. [En línea]. Disponible en:

<http://sistemredes-najomat.blogspot.com/p/modos-de-transmision.html>

[Accedido: 23-sep-2018]

[6] Análisis y simulación de un protocolo de comunicaciones para una red de sensores: Aplicación en dispositivos de vuelo. [En línea]. Disponible en:

http://sedici.unlp.edu.ar/bitstream/handle/10915/27353/Documento_completo.pdf?sequence=1

[Accedido: 23-sep-2018]

[7] Topología de red – Que es, tipos y características. [En línea]. Disponible en:

<http://247tecno.com/topologia-de-red-tipos-caracteristicas/>

[Accedido: 24-sep-2018]

[8] Redes industriales. [En línea]. Disponible en:

vip.ucaldas.edu.co/vector/downloads/Vector7_3.pdf

[Accedido: 25-sep-2018]

[9] Implementación de las comunicaciones pc-autómata-robot mediante interfaz ethernet industrial. [En línea]. Disponible en:

http://egela.oteitzalp.org/pluginfile.php/8806/mod_resource/content/0/infoPLC_n et_PFC_Miguel_Alvaro_Yunta_COM_S7-300_RobotABB_.pdf

[Accedido: 26-sep-2018]

[10] Chapter 2: The WorldFIP Protocol Standard and Specifications. [En línea]. Disponible en:

<http://people.cs.pitt.edu/~mhanna/Master/ch2.pdf>

[Accedido 27-sep-2018]

[11] Capítulo II: Fundamentos teóricos. [En línea]. Disponible en:
<http://bibing.us.es/proyectos/abreproy/12219/fichero/Capítulo+2+-+Fundamentos+teóricos.pdf>

[Accedido: 29-sep-2018]

[12] ANÁLISIS DE LOS ACCIONADORES DEL ROBOT HUMANOIDE RH-2. [En línea]. Disponible en:

<https://e-archivo.uc3m.es/handle/10016/6975#preview>

[Accedido: 3-oct-2018]

[13] REDES Y BUSES DE CAMPO. [En línea]. Disponible en:

<http://biblioteca.utb.edu.co/notas/tesis/0030756.pdf>

[Accedido: 4-oct-2018]

[14] Sistema Naval de Adquisición de Datos. [En línea]. Disponible en:

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/63725/6/vgarridosTFM0617memoria.pdf>

[Accedido: 6-oct-2018]

[15] Bus CAN: estado de buses industriales y aplicaciones. [En línea]. Disponible en:

http://instrumentacionycontrol.net/Descargas/Documentacion/Comunicaciones/lyCnet_Busses%20de%20campo.pdf

[Accedido: 8-oct-2018]

[16] Diseño de una red CAN bus con Arduino. [En línea]. Disponible en:

<https://academica-e.unavarra.es/bitstream/handle/2454/19115/TFG%20Diseño%20de%20una%20Red%20Can%20bus%20-%20Alejandro%20García%20Osés.pdf?sequence=1&isAllowed=y>

[Accedido: 10-oct-2018]

[17] Bus CAN. [En línea]. Disponible en:

https://es.wikipedia.org/wiki/Bus_CAN

[Accedido: 12-oct-2018]

[18] Conector M12 5P. [En línea]. Disponible en:

<https://www.calimport.cl/productos/1/8/917-conector-m12-5p-cable-5-mts-recto>

[Accedido: 15-oct-2018]

[19] Automatización de la estación de ensayos y caracterización de abonadoras y sembradoras. Proyecto final de carrera de Francisco Javier Blanco Curieses.

[Accedido: 22-oct-2018]

[20] Raspberry Pi. [En línea]. Disponible en:

https://es.wikipedia.org/wiki/Raspberry_Pi

[Accedido: 24-oct-2018]

[21] Desarrollo de un sistema de navegación para un robot móvil. [En línea]. Disponible en:

<https://uvadoc.uva.es/bitstream/10324/12993/1/TFG-P-216.pdf>

[Accedido: 2-nov-2018]

[22] Raspberry Pi: tabla técnica completa. [En línea]. Disponible en:

<https://rasberryparatorpes.net/raspberry-pi-tabla-tecnica-completa/>

[Accedido: 4-nov-2018]

[23] LAN7500/LAN7500I. [En línea]. Disponible en:

<http://ww1.microchip.com/downloads/en/DeviceDoc/00001734B.pdf>

[Accedido: 10-nov-2018]

[24] LAN9514/LAN9514I. [En línea]. Disponible en:

<http://ww1.microchip.com/downloads/en/DeviceDoc/00002306A.pdf>

[Accedido: 6-nov-2018]

[25] Entradas y salidas digitales. [En línea]. Disponible en:

<https://franciscomoya.gitbooks.io/taller-de-raspberry-pi/content/es/elems/gpio.html>

[Accedido: 13-nov-2018]

[26] ¿Qué es GPIO?. [En línea]. Disponible en:

<https://www.programoergosum.com/cursos-online/raspberry-pi/238-control-de-gpio-con-python-en-raspberry-pi/que-es-gpio>

[Accedido: 13-nov-2018]

[27] ¡Asignación de pines! La guía detallada sobre la asignación de pines GPIO para Raspberry Pi. [En línea]. Disponible en:

<https://es.pinout.xyz/#>

[Accedido: 14-nov-2018]

[28] Raspberry Pi LCD DSI Display Connector. [En línea]. Disponible en:

https://www.petervis.com/Raspberry_Pi/Raspberry_Pi_LCD/Raspberry_Pi_LCD_DSI_Display_Connector.html

[Accedido: 15-nov-2018]

[29] Raspberry Pi CSI Connector Pinout. [En línea]. Disponible en:

https://www.petervis.com/Raspberry_Pi/Raspberry_Pi_CSI/raspberry-pi-csi-interface-connector-pinout.html

[Accedido: 16-nov-2018]

[30] Micro B USB Jack. [En línea]. Disponible en:

<https://components101.com/connectors/micro-usb-pinout-datasheet>

[Accedido: 17-nov-2018]

[31] UB connector Micro USB. [En línea]. Disponible en:

www.farnell.com/datasheets/1794424.pdf

[Accedido: 17-nov-2018]

[32] ¿Tu Raspberry no funciona? Que no cunda el pánico. [En línea]. Disponible en:

<https://www.raspberrypizaragoza.es/tu-raspberry-no-funciona-que-no-cunda-el-panico/>

[Accedido: 19-nov-2018]

[33] archivo bootcode.bin. [En línea]. Disponible en:

<https://github.com/raspberrypi/firmware/blob/234c19de7cbaaf4997671d61df20a05759066295/boot/bootcode.bin>

[Accedido: 20-nov-2018]

[34] Control de un Robot Mediante Raspberry Pi. [En línea]. Disponible en:

<https://upcommons.upc.edu/bitstream/handle/2099.1/24012/ALBERTO%20MONTON%20CUARTERO.pdf?sequence=1&isAllowed=y>

[Accedido: 20-nov-2018]

[35] Diseño Hardware del Sistema. [En línea]. Disponible en:

<http://bibing.us.es/proyectos/abreproy/40048/fichero/VOLUMEN+1.+MEMORIA%252F11.+Diseño+Hardware+del+Sistema.pdf>

[Accedido: 23-nov-2018]

[36] KiCad. [En línea]. Disponible en:

<https://es.wikipedia.org/wiki/KiCad>

[Accedido: 24-nov-2018]

[37] CAN bus on Raspberry Pi with MCP2515. [En línea]. Disponible en:

<https://www.raspberrypi.org/forums/viewtopic.php?t=141052>

[Accedido: 25-nov-2018]

[38]MCP2515. [En línea]. Disponible en:

<http://ww1.microchip.com/downloads/en/devicedoc/21801e.pdf>

[Accedido: 27-nov-2018]

[39]MCP2551. [En línea]. Disponible en:

<http://ww1.microchip.com/downloads/en/devicedoc/21667e.pdf>

[Accedido: 29-nov-2018]

[40] Introducción a CAN bus: Descripción, ejemplos y aplicaciones de tiempo real. [En línea]. Disponible en:

http://oa.upm.es/48061/10/TFM_ADRIAN_MARTINEZ_REQUENA.pdf

[Accedido: 30-nov-2018]

[41] Organización Internacional de Normalización. [En línea]. Disponible en:

https://es.wikipedia.org/wiki/Organización_Internacional_de_Normalización

[Accedido: 5-dic-2018]

[42] Aplicación de gestión de bases de datos para la colección de muestras de hematología. [En línea]. Disponible en:

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/63505/3/allabaITFG0617memoria.pdf>

[Accedido: 6-dic-2018]

[43] Diseño e implementación de una estación meteorológica con Raspberry Pi. [En línea]. Disponible en:

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/52761/11/atobajasqTFM0616memoria.pdf>

[Accedido: 10-dic-2018]

[44] Asistente de aparcamiento con Arduino. [En línea]. Disponible en:

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/64425/7/egallifamTFG0617memoria.pdf>

[Accedido: 12-dic-2018]

[45] Estudio de los sensores para la detección de obstáculos aplicables a robots móviles. [En línea]. Disponible en:

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/80846/6/jacobarTFM0618memoria.pdf>

[Accedido: 15-dic-2018]

[46] Introducción a ROS en Raspberry Pi. [En línea]. Disponible en:

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/63705/3/jgutierrezperTFM0617memoria.pdf>

[Accedido: 17-dic-2018]

[47] Arduino+módulo GSM/GPRS: monitorización, automatización y gestión remota en un viñedo. [En línea]. Disponible en:

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/65345/7/radriandTFG0617anexos.pdf>

[Accedido: 20-dic-2018]

[48] Control domótico de temperatura mediante Arduino UNO. [En línea]. Disponible en:

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/32261/7/rcabellocTFG0614presentacion.pdf>

[Accedido: 26-dic-2018]