UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍAS INDUSTRIALES

INGENIERÍA TÉCNICA DE TELECOMUNICACIONES

ESPECIALIDAD: SISTEMAS ELECTRÓNICOS

PROYECTO FIN DE CARRERA

DESARROLLO DE UN INTERFACE USB-CAN

Autores:

Alvear Granja, Francisco Javier

Hernando Esteban, Juan Francisco

Tutor:

Pedro Luis Díez Muñoz **Dpto. Tecnología Electrónica**

JUNIO - 2012

AGRADECIMIENTOS

A nuestro tutor Pedro Luis que siempre cuando hemos acudido a él, desde el comienzo de este trabajo, nos ha mostrado como nadie su disposición e imprescindible ayuda, que nos ha permitido trabajar y adquirir muchos conocimientos durante todo este tiempo.

Agradecerle también el tiempo que nos ha dedicado, haciendo que hayamos recibido un trato personal y amistoso en la realización del mismo, sabiendo entendernos en todo momento.

A Raúl por su ayuda en la parte más complicada para nosotros y siempre ha estado dispuesto a perder de su tiempo para emplearlo con nosotros.

A Dani por su colaboración en la realización de los planos técnicos con Autocad.

Y a nuestras familias, que siempre han sido un gran apoyo en todo momento y que han padecido los éxitos y los sinsabores durante nuestra carrera.

ÍNDICE

1.-ENUNCIADO Y JUSTIFICACIÓN DEL PROYECTO

1.- ENUNCIADO Y JUSTIFICACIÓN DEL PROYECTO

Siendo requisito imprescindible para obtener la titulación de Ingeniero Técnico de Telecomunicaciones especialidad en Sistemas Electrónicos, Francisco Javier Alvear Granja y Juan Francisco Hernando Esteban nos disponemos a la realización del Proyecto Fin de Carrera titulado “Desarrollo de un interface USB-CAN” que consistirá en el desarrollo de una tarjeta para la transmisión de datos desde un BUS CAN a un PC a través del puerto USB del mismo y viceversa.

2.- DESCRIPCIÓN GENERAL DEL SISTEMA BUS-CAN

2.1.-INTRODUCCIÓN

Can-Bus es un protocolo de comunicación en serie desarrollado para el intercambio de información entre unidades de control electrónicas del automóvil.

Este sistema permite compartir una gran cantidad de información entre las unidades de control abonadas al sistema, lo que provoca una reducción importante tanto del número de sensores utilizados como de la cantidad de cables que componen la instalación eléctrica.

De esta forma aumentan considerablemente las funciones presentes en los sistemas del automóvil donde se emplea el Can-Bus sin aumentar los costes, además de que estas funciones pueden estar repartidas entre dichas unidades de control.

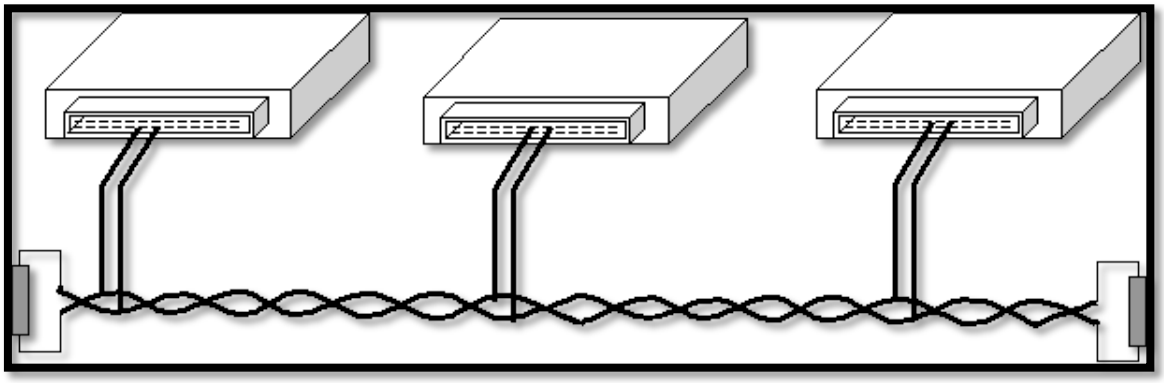


Figura 1- Cap.2.1: Disposición de un sistema BUS-CAN

2.2.- MOTIVACIÓN HISTÓRICA DEL BUS CAN

El sistema de bus serie CAN, siglas cuyo significado en castellano es "Control de Área de Red", nació en febrero de 1986, cuando el grupo "Robert Bosch GmbH" (más conocido como "Bosch") lo presentó en el congreso de la "Sociedad de Ingeniería de la Automoción". Desde entonces, CAN se ha convertido en uno de los protocolos líderes en la utilización del bus serie.

2.2.1.-RAZONES

A comienzos de los 80, los ingenieros de Bosch evaluaron el posible uso de los sistemas de bus serie existentes en los coches de pasajeros, pero ninguno de los protocolos de red disponibles satisfacía los requisitos de estos.

El ingeniero Uwe Kiencke fue quien inició el desarrollo del nuevo sistema en 1983.

Este protocolo de bus serie se ideó principalmente para aportar mayor funcionalidad, seguridad y fiabilidad, junto a una mayor eficiencia en el gasto del combustible, ya que la reducción del peso y la complejidad de los automóviles a través de la reducción del cableado iban a favorecer este hecho.

Ingenieros de "Mercedes-Benz" pronto se involucraron en las primeras fases de creación del nuevo sistema, y pronto lo hizo también "Intel" como potencial vendedor de semiconductores. El profesor Wolfhard Lawrenz de la universidad de ciencias aplicadas de Brunswick-Wolfenbüttel, Alemania, fue quién dio al nuevo protocolo de red el nombre de "Controller Area Network" (CAN).

A mediados de 1987, "Intel" presentó el primer chip de controlador CAN: el 82526. Poco tiempo después, "Semiconductores Philips" presentaría el 82C200. Estos dos antepasados primigenios de los controladores CAN actuales eran completamente distintos en cuanto a filtros de aceptación y control de mensajes.

Intel adoptó el concepto de FullCAN; este requería menos carga de la CPU del microcontrolador que la implementación BasicCAN elegida por Philips. Pero por otra parte, el dispositivo de FullCAN era limitado con respecto al número de los mensajes que podían ser recibidos. Además, el controlador de BasicCAN requería menos silicio, lo que abarataba aún más su coste. Actualmente, los términos 'BasicCAN' y 'FullCAN' han quedado obsoletos.

2.2.2.-ESTANDARIZACIÓN

La especificación CAN (versión 2.0) de "Bosch" fue sometida a la estandarización internacional a comienzos de los 90.

Concretamente en Noviembre de 1993, después de diversos conflictos políticos, se publicó el estándar ISO 11898, que definía además una capa física para velocidades de hasta 1 Mbit/s.

Paralelamente, un formato de CAN 'tolerante a fallos' se incluyó en la ISO 11519-2. En 1995, el estándar se amplió con la descripción del identificador CAN de 29 bits. Desafortunadamente, todas las especificaciones y estandarizaciones publicadas acerca de CAN contenían errores o estaban incompletas. Para evitar incompatibilidades, "Bosch" se cercioró, y sigue haciéndolo, de que todos los micros CAN cumplen con el modelo de referencia que ellos definieron.

De todas formas, las especificaciones CAN han sido revisadas y estandarizadas con el tiempo en diferentes secciones: la norma ISO 11898-1 describe la 'capa de transmisión de datos CAN'; la ISO 11898-2 la 'capa física CAN no tolerante a fallos'; y la ISO 11898-3 la 'capa física CAN tolerante a fallos'. Los estándares de ISO 11992 (referente a la interfaz para camiones y remolques) e ISO 11783 (referente a la maquinaria agrícola y forestal) definen los perfiles del uso de CAN basados en el US-protocol J1939.

2.2.3.-PIONEROS

A pesar de que CAN surgió con la idea de ser utilizado en coches de pasajeros, un gran número de las primeras aplicaciones llegarían desde otros segmentos de mercado distintos.

Entre otros, el fabricante finlandés de ascensores "Kone", fue uno de los primeros en aprovecharse de sus ventajas. La oficina de ingeniería Suiza "Kyaser" lo introdujo a los fabricantes de maquinaria textil del país, que acabaron fundando el 'Grupo de usuarios textiles CAN', bajo el liderazgo de 'Lars-Berno Fredriksson'.

En Holanda, "Sistemas médicos Philips" lo utilizó para las redes internas de sus máquinas de Rayos X. La 'especificación para mensajes de Philips' (PMS), desarrollada principalmente por Tom Suters, representó la primera capa de aplicación para redes CAN. También se sucedieron todo tipo de propuestas académicas, como por ejemplo, la creación a finales de los 80 de un sistema de bus basado en CAN, para vehículos agrícolas (LBS).

Ya en 1992, Mercedes-Benz implantó CAN en sus automóviles de clase alta. Y aunque en un principio el bus se limitaba a interconectar las unidades de control electrónico que cuidaban del control del motor, pronto conectarían además las unidades de control necesarias para los componentes electrónicos. Para ello, se implementaron dos sistemas de bus

CAN separados físicamente, y conectados a través de 'gateways'. Actualmente, muchos otros fabricantes de coches como "BMW", "Renault", "Fiat", "Saab", "Volkswagen" o "Volvo", han seguido su ejemplo y suelen implementar dos redes CAN en sus coches.

En Marzo de ese mismo año, usuarios internacionales y grupos de fabricantes fundaron oficialmente la organización 'CAN en la Automoción' (CiA).

La primera publicación técnica, que trataba acerca de la capa física, fue emitida solo unas semanas después: 'CiA' recomendaba utilizar solamente transceptores CAN que cumplieran la normativa ISO 11898. A día de hoy, son los transceptores RS485 los utilizados más comúnmente.

Otra de las primeras tareas de la CiA fue la especificación de la 'capa de aplicación CAN', que se desarrolló a partir del material existente de los "Sistemas médicos de Philips" y de "STZP".

2.2.4.-CAPAS DE APLICACIÓN

Desde 1993, dentro del alcance del proyecto 'ASPIC', un consorcio europeo liderado por "BOSCH", desarrolló un prototipo conocido como CANopen para las redes internas de las celdas de producción. En 1995, se publicó el perfil totalmente revisado de las comunicaciones de CANopen, que en el plazo de cinco años ya se había convertido en la red integrada estandarizada más importante de Europa, puesto que ofrecía una gran flexibilidad y un gran número de opciones de configuración, así como diversos perfiles de dispositivo, interfaz y uso.

Actualmente, CANopen se sigue utilizando especialmente en Europa: máquinas de moldeado por inyección en Italia, máquinas expendedoras en Inglaterra, grúas en Francia, control de maquinaria en Austria, o maquinaria de fabricación de relojes en Suiza... Mientras que en Estados Unidos CANopen se abre camino en el ámbito de los toros mecánicos y las máquinas clasificadoras.

Poco después de aquello, a comienzos de 1994, "Allen-Bradley" desarrolló la tecnología DeviceNet, enfocada esencialmente a la automatización de las fábricas. Pronto ganaría adeptos en Estados Unidos debido sobretodo a su funcionamiento "plug & play" (enchufar y listo).

Pero no fueron las únicas. Desde el momento de la creación de CAL, se fue sucediendo la creación de diferentes estándares que definían la capa de aplicación; algunos son muy específicos y están relacionados casi exclusivamente con sus campos de aplicación. Dejando de lado las 3 mencionadas anteriormente, entre los protocolos de alto nivel y las capas de aplicación más utilizadas cabe mencionar: SDS, OSEK y CANKingdom.

Por una parte, SDS (Smart Distributed System), fue creado por Honeywell durante la gestación de DeviceNet, y de hecho era muy similar a este, así que no entraremos en detalle. OSEK, que estaba basado además en TCP/IP y algoritmos DSP, rápidamente enfocaría su aplicación a una gran cantidad de usos en los turismos. Mientras que CANKingdom se enfocó a aplicaciones con una necesidad alta de aplicaciones en tiempo real, como pueda ser por ejemplo la electrónica marítima.

Ya por último, en el año 2000, ISO formó un grupo de trabajo que agrupaba empleados de "Bosch", académicos y expertos de la industria de los semiconductores, que definió un nuevo protocolo basado en CAN, el llamado TTCAN (Time-tiggered communication on CAN). Esta extensión permitía tanto la transmisión de mensajes simultáneos, como la implementación de un bucle cerrado para el control del bus. Y gracias a que el protocolo del bus no fue modificado, este tipo de mensajes podían seguir siendo enviados con el mismo sistema físico de bus.

2.3.-PRINCIPALES CARACTERÍSTICAS DEL BUS CAN

• Económico y sencillo: Dos de las razones que motivaron su desarrollo fueron precisamente la necesidad de economizar el coste monetario y el de minimizar la complejidad del cableado, por parte del sector automovilístico.

• Estandarizado: Se trata de un estándar definido en las normas ISO (Internacional Organization for Standardization), concretamente la ISO 11898, que se divide a su vez en varias partes, cada una de las cuales aborda diferentes aspectos de CAN.

• Medio de transmisión adaptable: El cableado, como ya hemos dicho, es muy reducido a comparación de otros sistemas. Además, a pesar de que por diversas razones el estándar de hardware de transmisión sea un par trenzado de cables, el sistema de bus CAN también es capaz de trabajar

con un solo cable. Esta particularidad es empleada en diversos tipos de enlaces, como los enlaces ópticos o los enlaces de radio.

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

• Programación sencilla.

• Número de nodos: Es posible conectar hasta 110 dispositivos en una sola red CAN.

• Garantía de tiempos de latencia: CAN aporta la seguridad de que se transmitirá cierta cantidad de datos en un tiempo concreto, es decir, que la latencia de extremo a extremo no excederá un nivel específico de tiempo. Además, la transmisión siempre será en tiempo real.

• Optimización del ancho de banda: Los métodos utilizados para distribuir los mensajes en la red, como el envío de estos según su prioridad, contribuyen a un mejor empleo del ancho de banda disponible.

• Desconexión autónoma de nodos defectuosos: Si un nodo de red cae, sea cual sea la causa, la red puede seguir funcionado, ya que es capaz de desconectarlo o aislarlo del resto. De forma contraria, también se pueden añadir nodos al bus sin afectar al resto del sistema, y sin necesidad de reprogramación.

• Velocidad flexible: ISO define dos tipos de redes CAN: una red de alta velocidad (de hasta 1 Mbps) definida por la ISO 11898-2, y una red de baja velocidad tolerante a fallos (menor o igual a 125 Kbps) definida por la ISO 11898-3.

• Relación velocidad-distancia: Al punto anterior habría que añadir que la velocidad también depende de la distancia hasta un máximo de 1000 metros (aunque podemos aumentar la distancia con bridges o repetidores), como podemos corroborar en la siguiente tabla comparativa:

Velocidad (Kbps)	Tiempo de bit (µs)	Longitud máxima de 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 1 – Cap 2.3: Relación entre velocidad y tiempo de bit

Además podemos hablar de otras características técnicas en las que profundizaremos más adelante:

• Orientado a mensajes: Se trata de un protocolo orientado a mensajes, y no a direcciones, es decir, la información que se va a intercambiar se descompone en mensajes, a los cuales se les asigna un identificador y son encapsulados en tramas para su transmisión. Cada mensaje tiene un identificador único dentro de la red, a partir del cual los nodos deciden aceptar o no dicho mensaje. Además están priorizados.

• Multidifusión (multicast): Permite que todos los nodos puedan acceder al bus de forma simultánea con sincronización de tiempos.

• Medio compartido (broadcasting): La información es enviada en la red a todos los destinos de forma simultánea. Así que los destinos habrán de saber si la información les concierne o deben rechazarla.

• Detección y señalización de errores: CAN posee una gran capacidad de detección de errores, tanto temporales, como permanentes, lograda a través de cinco mecanismos de detección, 3 al nivel de mensaje y 2 a nivel de bit. Los errores además pueden ser señalizados.

• Retransmisión automática de tramas erróneas: Junto a la detección y señalización de errores la retransmisión automática de tramas erróneas aporta la integridad de los datos. Además ambos procesos son transparentes al usuario.

• Jerarquía multimaestro: CAN es un sistema multimaestro en el cual puede haber más de un maestro (o master) al mismo tiempo y sobre la misma red, es decir, todos los nodos son capaces de transmitir, hecho que permite construir sistemas inteligentes y redundantes.

2.4.-TRANSMISION DE DATOS

2.4.1.-ARQUITECTURA DE CAPAS

Las implementaciones hardware de CAN cubren de forma estandarizada las capas físicas y de enlace del modelo de comunicaciones OSI (Open Systems Interconection), mientras diversas soluciones de software no estandarizadas cubren la capa de aplicación.

Las estandarizaciones ISO (International Standard Organization), a diferencia de las normas BOSCH, especifican también el medio de comunicación. Por lo tanto una implementación CAN a partir de las especificaciones de BOSCH no siempre será compatible con las normas ISO.

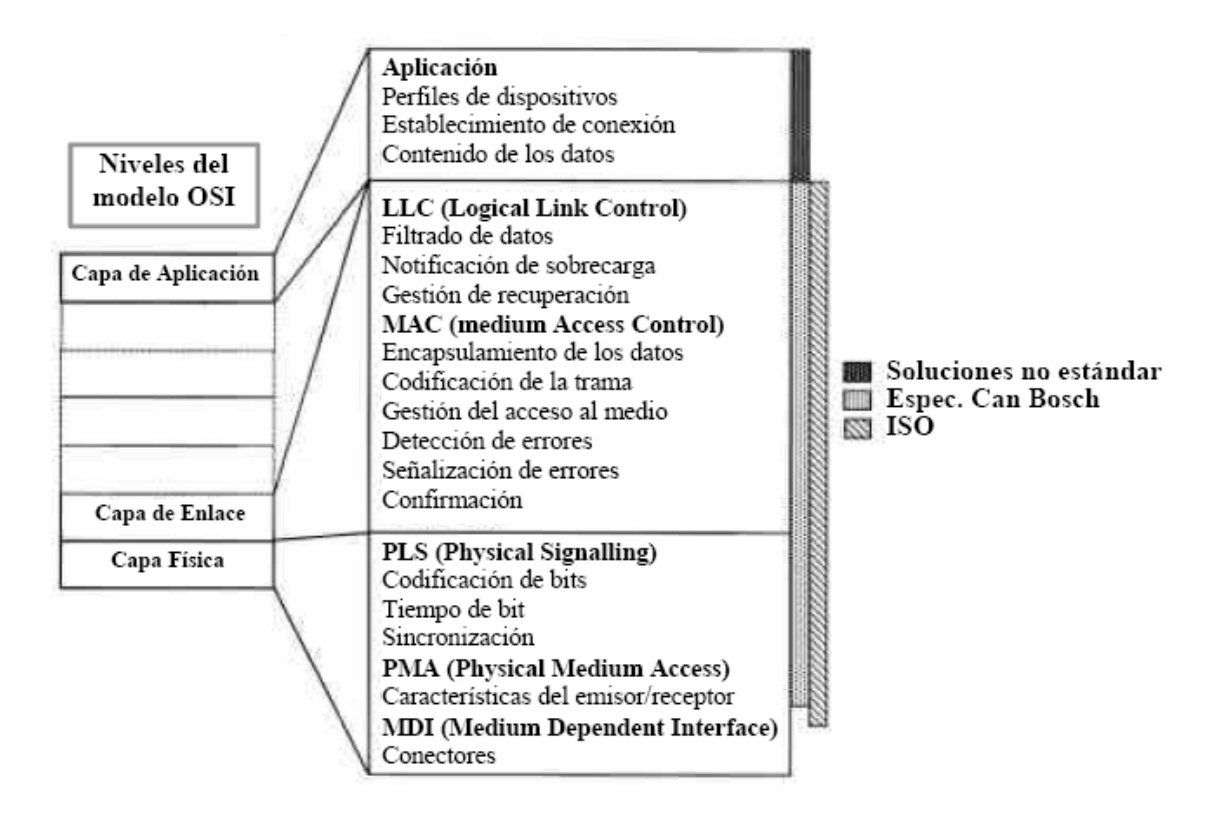


Figura 2- Cap. 2.4.1 : Capas del Protocolo CAN

2.4.1.1.-CAPA FÍSICA

La capa física de CAN, es responsable de la transferencia de bits entre los distintos nodos que componen la red. Define aspectos como niveles de señal, codificación, sincronización y tiempos en que los bits se transfieren al bus.

En la especificación original de CAN, la capa física no fue definida, permitiendo diferentes opciones para la elección del medio y niveles eléctricos de transmisión. Las características de las señales eléctricas en el bus, fueron establecidas más tarde por el ISO 11898 para las aplicaciones de alta velocidad y, por el estándar ISO 11519 para las aplicaciones de baja velocidad.

2.4.1.1.1.- ESTÁNDAR 11519

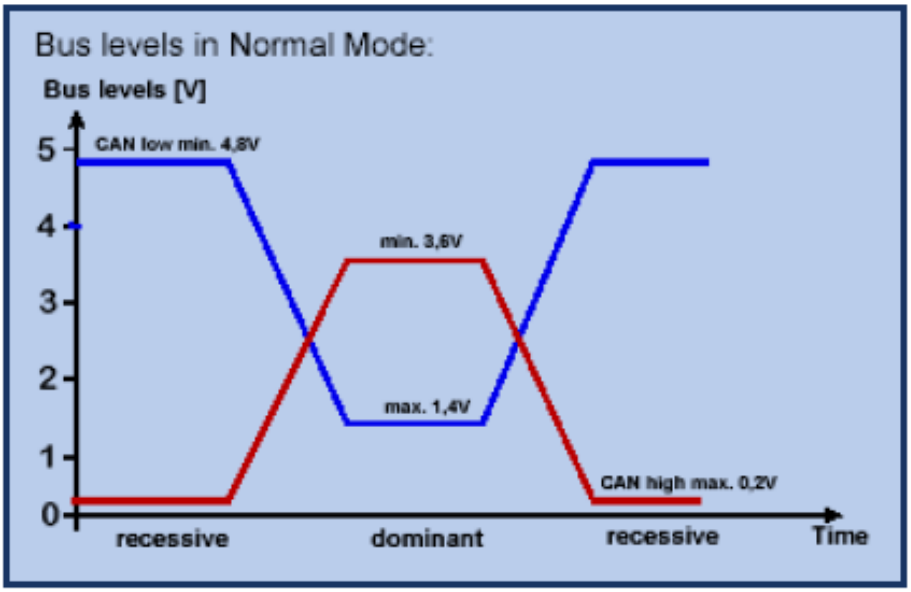


Figura 3 – Cap.2.4.1.1.1: Niveles del Bus en Modo Normal

Los nodos conectados en este bus interpretan dos niveles lógicos denominados: Dominante y Recesivo.

- Dominante: la tensión diferencial (CAN_H - CAN_L) es del orden de 2.0 V con CAN_H = 3.5V y CAN_L = 1.5V (nominales).

- Recesivo: la tensión diferencial (CAN_H - CAN_L) es del orden de 5V con CAN_H = 0V y CAN_L= 5V (nominales).

A diferencia del bus de alta velocidad, el bus de baja velocidad requiere dos resistencias en cada transceptor: RTH para la señal CAN_H y RTL para la señal CAN_L. Esta configuración permite al transceptor de bus de baja velocidad (fault-tolerant) detectar fallas en la red. La suma de todas las resistencias en paralelo, debe estar en el rango de 100-500.

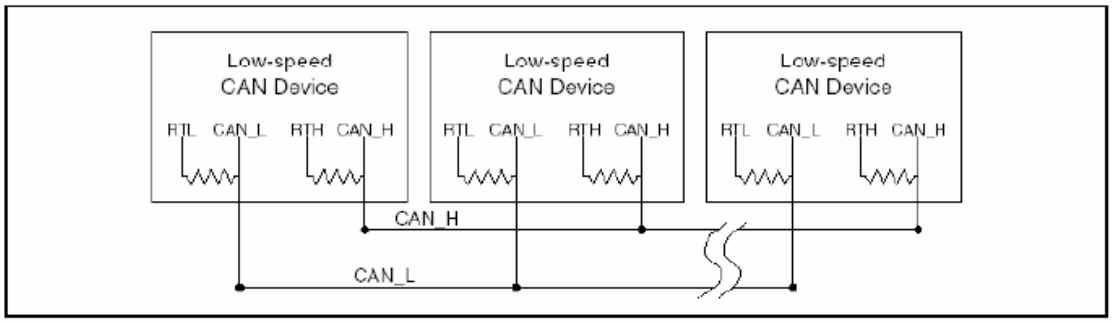


Figura 4 – Cap.2.4.1.1.1: Red Bus CAN de Baja Velocidad

2.4.1.1.2.- ESTÁNDAR 11898

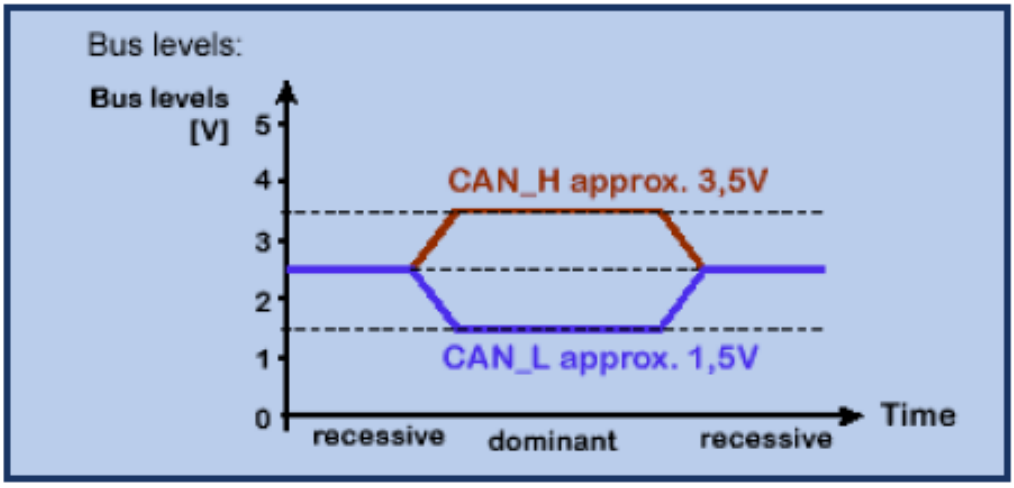


Figura 5 – Cap.2.4.1.1.2: Niveles del Bus

Los nodos conectados en este bus interpretan los siguientes niveles lógicos:

- *Dominante*: la tensión diferencial (CAN_H - CAN_L) es del orden de 2.0 V con CAN_H = 3.5V y CAN_L = 1.5V (nominales).

- *Recesivo*: la tensión diferencial (CAN_H - CAN_L) es del orden de 0V con CAN_H = CAN_L = 2.5V (nominales).

El par de cables trenzados (CAN_H y CAN_L) constituyen una transmisión de línea. Si dicha transmisión de línea no está configurada con

los valores correctos, cada trama transferida causa una reflexión que puede originar fallos de comunicación. Como la comunicación en el bus CAN fluye en ambos sentidos, ambos extremos de red deben de estar cerrados mediante una resistencia de 120. Ambas resistencias deberían poder disipar 0.25 w. de potencia.

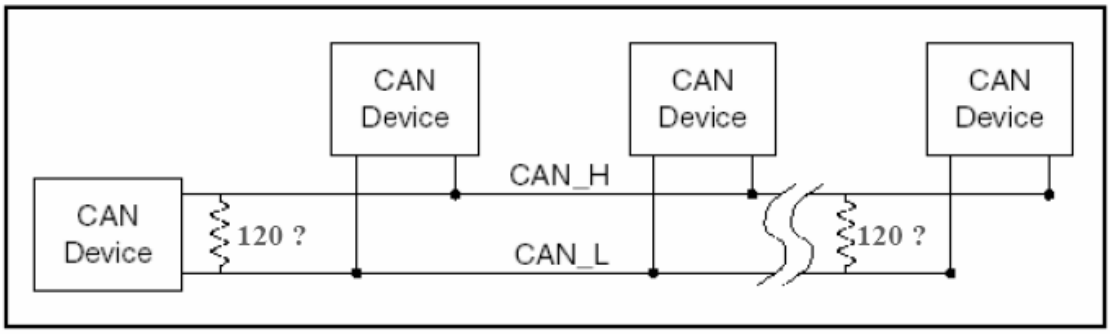


Figura 6 – Cap.2.4.1.1.2: Red Bus CAN de Alta Velocidad.

2.4.1.2.-CAPA DE ENLACE DE DATOS

La capa de enlace de datos es responsable del acceso al medio y el control lógico y está dividida a su vez en dos niveles.

- *El subnivel LLC (Logical Link Control):* Gestiona el filtrado de los mensajes, las notificaciones de sobrecarga y la administración de la recuperación.

- *El subnivel MAC (Medium Acces Control):* Es el núcleo del protocolo CAN y gestiona el tramado y desentramado de los mensajes, el arbitraje a la hora de acceder al bus y el reconocimiento de los mensajes, así como el chequeo de posibles errores y su señalización, el aislamiento de fallos en unidades de control y la identificación del estado libre del bus para iniciar una transmisión o recepción de un nuevo mensaje.

2.4.2.-ESTRUCTURA DE UN NODO CAN

Dentro de un nodo CAN, se pueden distinguir una serie de módulos interconectados entre ellos: un bus de direcciones, datos y un control (paralelo) enlazando el controlador central, la memoria de los datos y el programa (donde está almacenado el software de aplicación y el

controlador de red de alto nivel), los dispositivos de entrada y salida y, la interfaz de comunicación.

Desde el punto de vista del controlador, la interfaz de comunicación se puede ver como un conjunto de buzones, donde cada uno de estos sirve como registro lógico de interfaz entre el controlador local y los nodos remotos. Si un nodo quiere comunicarse, tiene que dar de alta los correspondientes buzones de recepción y transmisión antes de hacer ninguna operación.

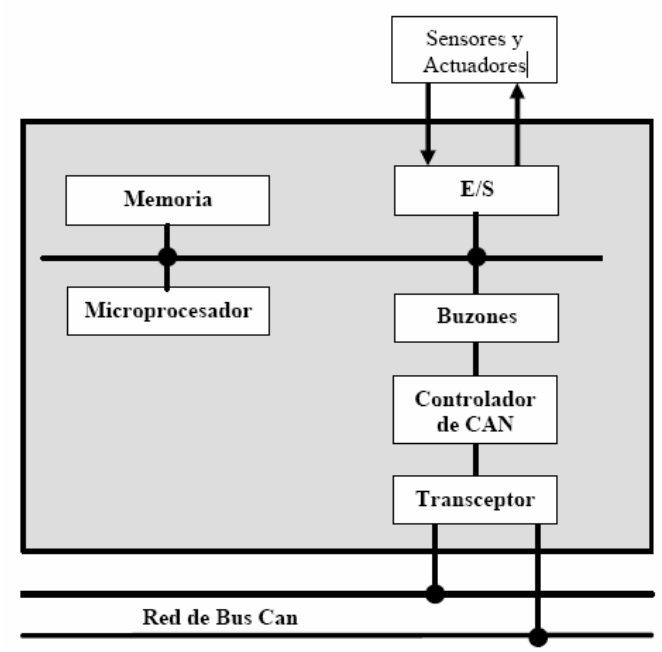


Figura 7 – Cap.2.4.2: Estructura de un nodo CAN

Teniendo en cuenta esta arquitectura, el funcionamiento sigue los siguientes pasos:

- Para inicializar, el programador especifica los parámetros de los registros de control de interfaz de comunicación, como las características del controlador de red o la velocidad de transmisión.

- A continuación, se actualizan todos los buzones. En cada buzón se especifica si es receptor o transmisor y, su estado inicial, inicializando su parte de datos del búfer.

- Posteriormente, para transmitir un mensaje es necesario poner los datos en el búfer de datos correspondiente al buzón de transmisión y activar el flanco de transmisión.

- Por último, la interfaz de red intenta comunicar los datos a través de la red. El estado de la transferencia se puede comprobar en el estatus de estado de cada buzón.

Una vez hecha la configuración inicial a partir del software de la capa de aplicación de esta manera, los nodos CAN funcionarán de forma autónoma.

2.4.3.-FORMATO DE CODIFICACIÓN Y SINCRONIZACIÓN DE DATOS

La codificación de bits se realiza por el método NRZ (Non-Return-to Zero) que se caracteriza por que el nivel de señal puede permanecer constante durante largos periodos de tiempo y habrá que tomar medidas para asegurarse de que el intervalo máximo permitido entre dos señales no es superado. Esto es importante para la sincronización (Bit Timing).

Este tipo de codificación requiere poco ancho de banda para transmitir, pero en cambio, no puede garantizar la sincronización de la trama transmitida. Para resolver esta falta de sincronismo se emplea la técnica del “bit stuffing”: cada 5 bits consecutivos con el mismo estado lógico en una trama (excepto del delimitador de final de trama y el espacio entre tramas), se inserta un bit de diferente polaridad, no perdiéndose así la sincronización. Por otro lado este bit extra debe ser eliminado por el receptor de la trama, que sólo lo utilizará para sincronizar la transmisión.

No hay flanco de subida ni de bajada para cada bit, durante el tiempo de bit hay bits dominantes (“0”) y recesivos (“1”) y disminuye la frecuencia de señal respecto a otras codificaciones.

2.4.4.-TRAMAS

2.4.4.1.-TIPO DE TRAMAS

El protocolo CAN está basado en mensajes, no en direcciones. El nodo emisor transmite el mensaje a todos los nodos de la red sin especificar un

destino y todos ellos escuchan el mensaje para luego filtrarlo según le interese o no.

Existen distintos tipos de tramas predefinidas por CAN para la gestión de la transferencia de mensajes:

- *Trama de datos*: Se utiliza normalmente para poner información en el bus y la pueden recibir algunos o todos los nodos.

- *Trama de información remota*: Puede ser utilizada por un nodo para solicitar la transmisión de una trama de datos con la información asociada a un identificador dado. El nodo que disponga de la información definida por el identificador la transmitirá en una trama de datos.

- *Trama de error*: Se generan cuando algún nodo detecta algún error definido.

- *Trama de sobrecarga*: Se generan cuando algún nodo necesita más tiempo para procesar los mensajes recibidos.

- *Espaciado entre tramas:* Las tramas de datos (y de interrogación remota) se separan entre sí por una secuencia predefinida que se denomina espaciado inter-trama.

- *Bus en reposo:* En los intervalos de inactividad se mantiene constantemente el nivel recesivo del bus.

En un bus CAN los nodos transmiten la información espontáneamente con tramas de datos, bien sea por un proceso cíclico o activado ante eventos en el nodo. La trama de interrogación remota sólo se suele utilizar para detección de presencia de nodos o para puesta al día de información en un nodo recién incorporado a la red. Los mensajes pueden entrar en colisión en el bus, el de identificador de mayor prioridad sobrevivirá y los demás son retransmitidos lo antes posible.

2.4.4.2.-TRAMA DE DATOS

Es la utilizada por un nodo normalmente para poner información en el bus. Puede incluir entre 0 y 8 bytes de información útil.

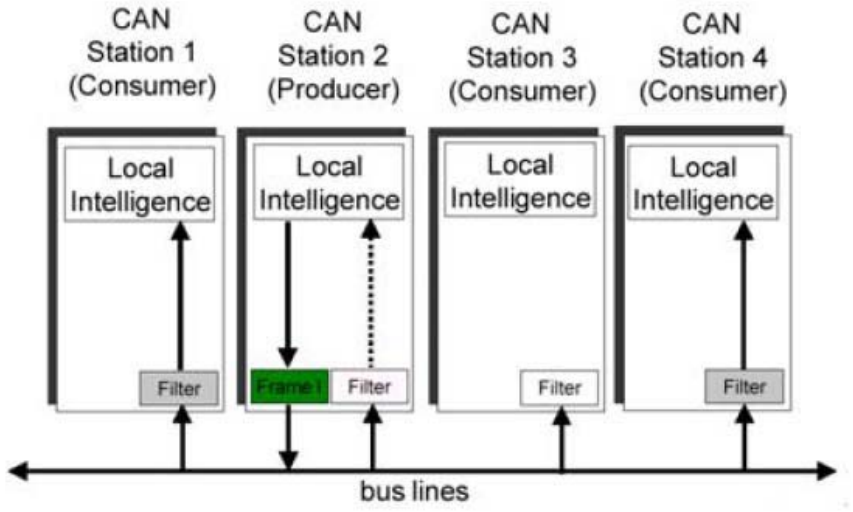


Figura 8 – Cap.2.4.4.2: Transmisión de trama de datos

Los mensajes de datos consisten en celdas que envían datos y añaden información definida por las especificaciones CAN:

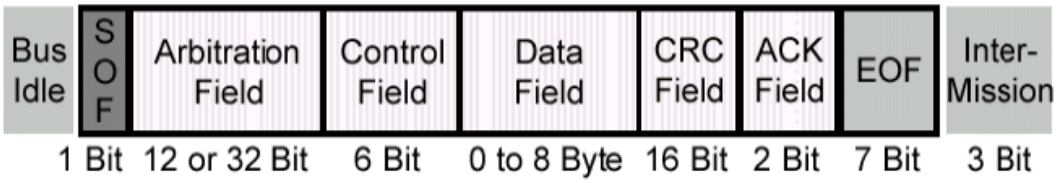


Figura 9 – Cap.2.4.4.2: Estructura de una trama de datos

- *Inicio de trama (SOF):* El inicio de trama es una celda de un sólo bit siempre dominante que indica el inicio del mensaje, sirve para la sincronización con otros nodos.

- *Celda de Arbitraje (Arbitration Field):* Es la celda que concede prioridad a unos mensajes o a otros.

En formato estándar tendrá 11 bits seguidos del bit RTR (Remote Transmisión Request) que en este caso será dominante.

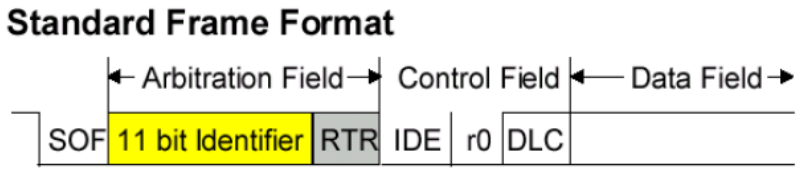


Figura 10 – Cap.2.4.4.2: Trama de datos formato estándar

En formato extendido serán 11 bits de identificador base y 18 de extendido. El bit SRR substituye al RTR y será recesivo.

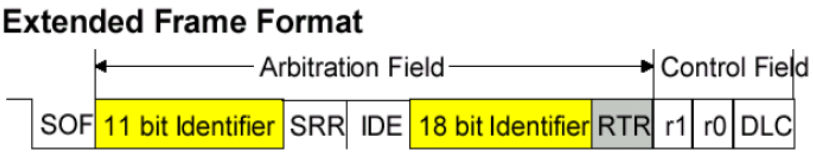


Figura 11 – Cap.2.4.4.2: Trama de datos formato extendido

NOTA: La trama en formato estándar prevalece sobre la extendida

- *Celda de control (Control Field):* El campo de control está formado por dos bits reservados para uso futuro y cuatro bits adicionales que indican el número de bytes de datos. En realidad el primero de estos bits (IDE) se utiliza para indicar si la trama es de CAN Estándar (IDE dominante) o Extendido (IDE recesivo). El segundo bit (RB0) es siempre recesivo. Los cuatro bits de código de longitud (DLC) indican en binario el número de bytes de datos en el mensaje (0 a 8)

- *Celda de Datos (Data Field):* Es el campo de datos de 0 a 8 bytes.

- *CRC:* Código de redundancia cíclica: Tras comprobar este código se podrá comprobar si se han producido errores.
- *Celda de reconocimiento (ACK):* es un campo de 2 bits que indica si el mensaje ha sido recibido correctamente. El nodo transmisor pone este bit como recesivo y cualquier nodo que reciba el mensaje lo pone como dominante para indicar que el mensaje ha sido recibido.

- *Fin de trama (EOF):* Consiste en 7 bits recesivos sucesivos e indica el final de la trama.

- *Espaciado* entre *tramas* (IFS): Consta de un mínimo de 3 bits recesivos.

2.4.4.3.-TRAMA REMOTA (Remote Frame)

Los nodos tienen habilidad para requerir información a otros nodos. Un nodo pide una información a los otros y el nodo que tiene dicha información envía una comunicación con la respuesta que puede ser recibida además por otros nodos si están interesados.

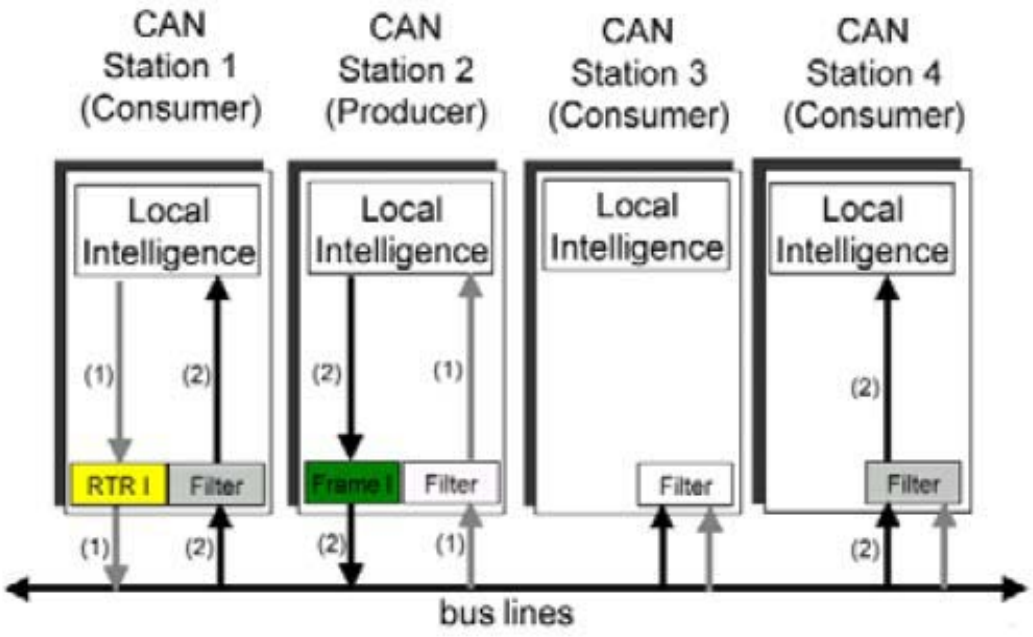


Figura 12 – Cap2.4.4.3: Transmisión de trama remota

Un mensaje de petición remota tiene la siguiente forma:

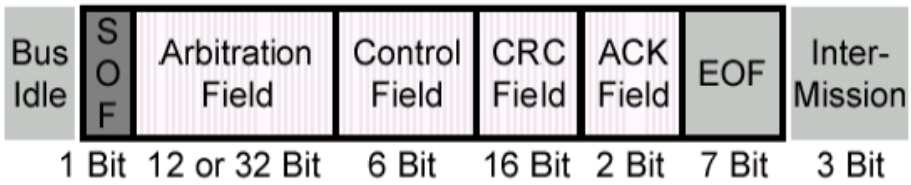


Figura 13 – Cap.2.4.4.3: Estructura de una trama remota

En este tipo de mensajes se envía una trama con el identificador del nodo requerido, a diferencia con los mensajes de datos, el bit RTR toma valor recesivo y no hay campo de datos.

En caso de que se envíe un mensaje de datos y de petición remota con el mismo identificador, el de datos ganará el acceso al bus puesto que el RTR lleva valor dominante.

2.4.4.4.-TRAMA DE ERROR

Las tramas de error son generadas por cualquier nodo que detecta un error. Consiste en dos campos: Indicador de error ("Error Flag") y Delimitador de error (“Error Delimeter”).

El *delimitador de error* consta de 8 bits recesivos consecutivos y permite a los nodos reiniciar la comunicación limpiamente tras el error.

El *Indicador de error* es distinto según el estado de error del nodo que detecta el error:

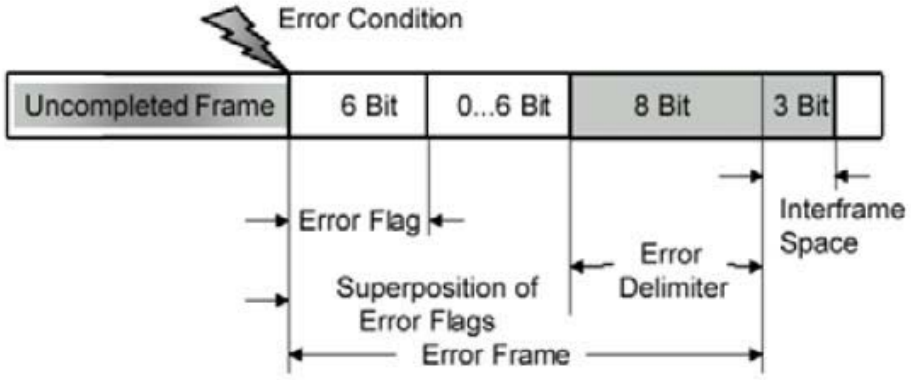


Figura 14 – Cap.2.4.4.4: Trama de error

Si un nodo en estado de error "Activo" detecta un error en el bus interrumpe la comunicación del mensaje en proceso generando un "Indicador de error activo" que consiste en una secuencia de 6 bits dominantes sucesivos. Esta secuencia rompe la regla de relleno de bits y provocará la generación de tramas de error en otros nodos. Por tanto el indicador de error puede extenderse entre 6 y 12 bits dominantes sucesivos. Finalmente se recibe el campo de delimitación de error formado por los 8 bits recesivos. Entonces la comunicación se reinicia y el nodo que había sido interrumpido reintenta la transmisión del mensaje.

Si un nodo en estado de error "Pasivo" detecta un error, el nodo transmite un "Indicador de error pasivo" seguido, de nuevo, por el campo delimitador de error. El indicador de error de tipo pasivo consiste en 6 bits recesivos seguidos y, por tanto, la trama de error para un nodo pasivo es

una secuencia de 14 bits recesivos. De aquí se deduce que la transmisión de una trama de error de tipo pasivo no afectará a ningún nodo en la red, excepto cuando el error es detectado por el propio nodo que está transmitiendo. En ese caso los demás nodos detectarán una violación de las reglas de relleno y transmitirán a su vez tramas de error.

Tras señalar un error por medio de la trama de error apropiada cada nodo transmite bits recesivos hasta que recibe un bit también recesivo, luego transmite 7 bits recesivos consecutivos antes de finalizar el tratamiento de error.

La regla de relleno de bits que aparece líneas superiores, consiste en que cada cinco bits de igual valor se introduce uno de valor inverso tal y como se ve en la figura siguiente:

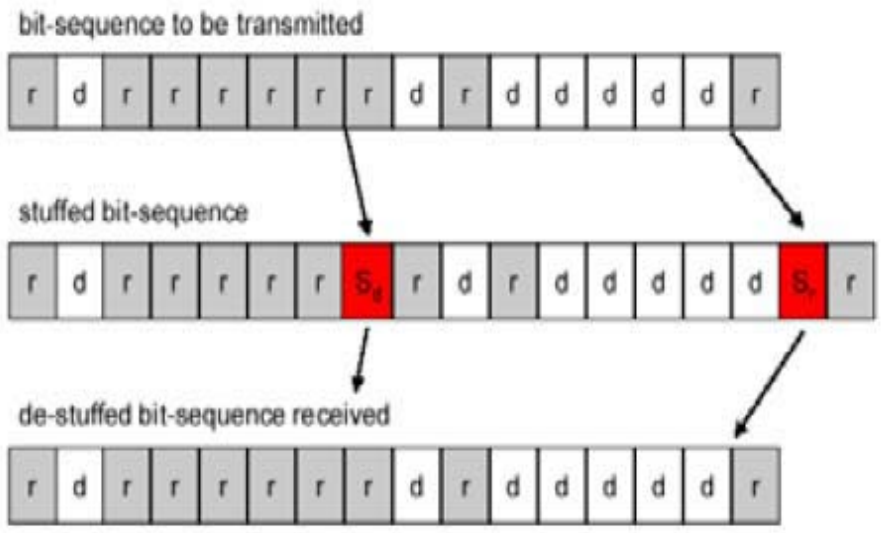


Figura 15 – Cap.2.4.4.4: Regla de relleno de bits

Otro método para la detección de errores es el chequeo de la trama**:**

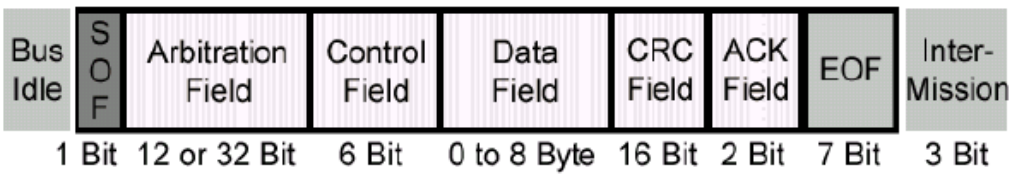


Figura 16 – Cap.2.4.4.4: Chequeo de trama

El campo CRC contiene información adicional a la trama, éste se calcula con un polinomio generador de igual manera en el receptor y en el emisor. Esto permite detectar errores aleatorios en hasta 5 bits o una secuencia seguida de 15 bits corruptos.

El campo ACK, en el caso del emisor será recesivo y el receptor deberá sobrescribirlo como dominante y el primero comprobará, mediante la monitorización, que el mensaje ha sido escuchado. Si no sucede así, la trama se considerará corrupta.

2.4.4.5.-TRAMA DE SOBRECARGA

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

Una trama de sobrecarga puede ser generada por cualquier nodo que debido a sus condiciones internas no está en condiciones de iniciar la recepción de un nuevo mensaje. De esta forma retrasa el inicio de transmisión de un nuevo mensaje. Un nodo puede generar como máximo 2 tramas de sobrecarga consecutivas para retrasar un mensaje.

Otra razón para iniciar la transmisión de una trama de sobrecarga es la detección por cualquier nodo de un bit dominante en los 3 bits de "intermission". Por todo ello una trama de sobrecarga de 5 generada por un nodo dará normalmente lugar a la generación de tramas de sobrecarga por los demás nodos dando lugar, como se ha indicado, a un máximo de 12 bits dominantes de indicador de sobrecarga.

2.4.4.6.-ESPACIO ENTRE TRAMAS

El espacio entre tramas separa una trama (de cualquier tipo) de la siguiente trama de datos o interrogación remota. El espacio entre tramas ha de constar de, al menos, 3 bits recesivos. Esta secuencia de bits se

denomina "íntermission". Una vez transcurrida esta secuencia un nodo en estado de error activo puede iniciar una nueva transmisión o el bus permanecerá en reposo.

Para un nodo en estado error pasivo la situación es diferente, deberá espera una secuencia adicional de 8 bits recesivos antes de poder iniciar una transmisión. De esta forma se asegura una ventaja en inicio de transmisión a los nodos en estado activo frente a los nodos en estado pasivo.

2.4.5.-ACCESO MÚLTIPLE Y ARBITRAJE DE ACCESO AL BUS

Unas de las características que distingue a CAN con respecto a otras normas, es su técnica de acceso al medio denominada como CSMA/CD+CR o "Carrier Sense, Multiple Access/Collision Detection + Collision Resolution" (Acceso Múltiple con detección de portadora, detección de colisión más Resolución de colisión). Cada nodo debe vigilar el bus en un periodo sin actividad antes de enviar un mensaje (Carrier Sense) y además, una vez que ocurre el periodo sin actividad cada nodo tiene la misma oportunidad de enviar un mensaje (Multiple Access). En caso de que dos nodos comiencen a transmitir al unísono se detectará la colisión.

El método de acceso al medio utilizado en bus CAN añade una característica adicional: la resolución de colisión. En la técnica CSMA/CD utilizada en redes Ethernet ante colisión de varias tramas, todas se pierden. CAN resuelve la colisión con la supervivencia de una de las tramas que chocan en el bus. Además la trama superviviente es aquella a la que se ha identificado como de mayor prioridad.

La resolución de colisión se basa en una topología eléctrica que aplica una función lógica determinista a cada bit, que se resuelve con la prioridad del nivel definido como bit de tipo dominante. Definiendo el bit *dominante* como equivalente al valor lógico '0' y bit *recesivo al* nivel lógico '1' se trata de una función AND de todos los bits transmitidos simultáneamente. Cada transmisor escucha continuamente el valor presente en el bus, y se retira cuando ese valor no coincide con el que dicho transmisor ha forzado. Mientras hay coincidencia la transmisión continua, finalmente el mensaje con identificador de máxima prioridad sobrevive. Los demás nodos reintentarán la transmisión lo antes posible.

Se ha de tener en cuenta que la especificación CAN de Bosh no establece cómo se ha de traducir cada nivel de bit (dominante o recesivo) a

variable física. Cuando se utiliza par trenzado según ISO 11898 el nivel dominante es una tensión diferencial positiva en el bus, el nivel recesivo es ausencia de tensión, o cierto valor negativo, (los transceptores no generan corriente sobre las resistencias de carga del bus). Esta técnica aporta la combinación de dos factores muy deseados en aplicaciones industriales distribuidas: la posibilidad de fijar con determinismo la latencia en la transmisión de mensajes entre nodos y el funcionamiento en modo multi maestro sin necesidad de gestión del arbitraje, es decir control de acceso al medio, desde las capas de software de protocolo. La prioridad queda así determinada por el contenido del mensaje, en CAN es un campo determinado, el identificador de mensaje, el que determina la prioridad.

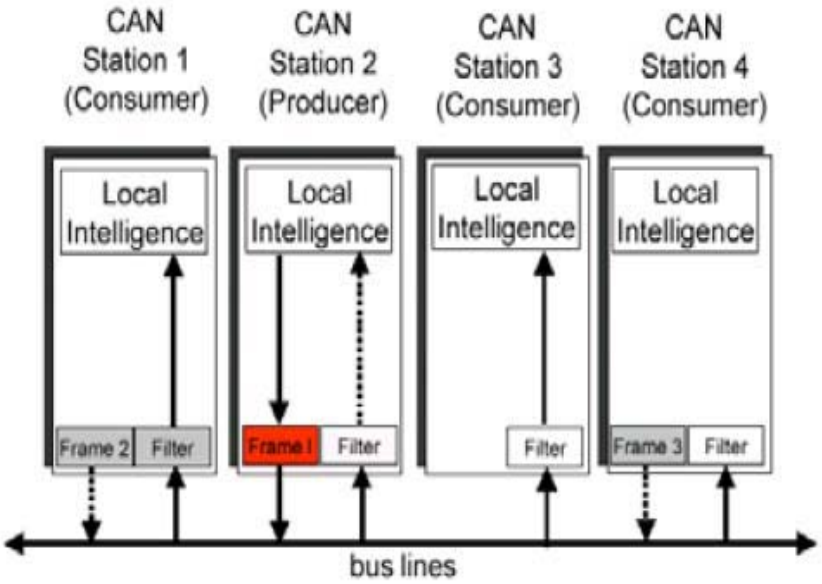


Figura 17 – Cap.2.4.5: Arbitraje de acceso al BUS

En un bus único, un identificador de mensaje ha de ser asignado a un solo nodo concreto, es decir, se ha de evitar que dos nodos puedan iniciar la transmisión simultánea de mensajes con el mismo identificador y datos diferentes. El protocolo CAN establece que cada mensaje es único en el sistema, de manera que por ejemplo, si en un automóvil existe la variable "presión de aceite", esta variable ha de ser transmitida por un nodo concreto, con un identificador concreto, con una longitud fija concreta y coherente con la codificación de la información en el campo de datos.

En la siguiente figura se ve un ejemplo de arbitraje en un bus CAN

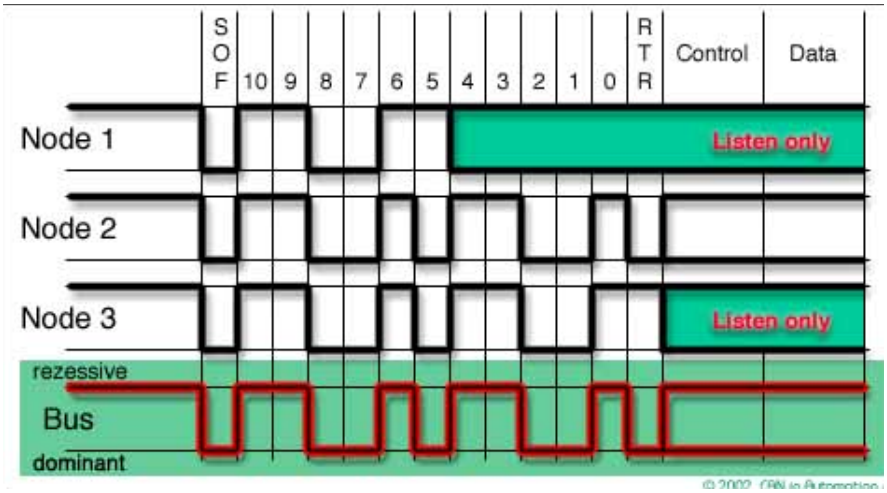


Figura 18 – Cap.2.4.5: Ejemplo de arbitraje en CAN

2.4.6.-DETECCIÓN DE ERRORES

Una de las características más importantes y útiles de CAN es su alta fiabilidad, incluso en entornos de ruido extremos, el protocolo CAN proporciona una gran variedad de mecanismos para detectar errores en las tramas. Esta detección de error es utilizada para retransmitir la trama hasta que sea recibida con éxito.

Otro tipo de error es el de aislamiento, y se produce cuando un dispositivo no funciona correctamente y un alto por ciento de sus tramas son erróneas. Este error de aislamiento impide que el mal funcionamiento de un dispositivo condicione el funcionamiento del resto de nodos implicados en la red.

2.4.6.1.-DETECCIÓN DE ERRORES

En el momento en que un dispositivo detecta un error en una trama, este dispositivo transmite una secuencia especial de bits, “el error flag”. Cuando el dispositivo que ha transmitido la trama errónea detecta el error flan, transmite la trama de nuevo. Los dispositivos CAN detectan los errores siguientes:

- *Error de bit:* Durante la transmisión de una trama, el nodo que transmite monitoriza el bus. Cualquier bit que reciba con polaridad inversa a la que ha transmitido se considera un error de bit, excepto

cuando se recibe durante el campo de arbitraje o en el bit de reconocimiento. Además, no se considera error de bit la detección de bit dominante por un nodo en estado de error pasivo que retransmite una trama de error pasivo.

- *Error de relleno (Stuff Error):* Se considera error de relleno la detección de 6 bits consecutivos del mismo signo, en cualquier campo que siga la técnica de relleno de bits, donde por cada 5 bits iguales se añade uno diferente.

- *Error de CRC:* Se produce cuando el cálculo de CRC realizado por un receptor no coincide con el recibido en la trama. El campo CRC (Cyclic Redundant Code) contiene 15 bits y una distancia de Hamming de 6, lo que asegura la detección de 5 bits erróneos por mensaje. Estas medidas sirven para detectar errores de transmisión debido a posibles incidencias en el medio físico como por ejemplo el ruido.

- *Error de forma (Form Error):* Se produce cuando un campo de formato fijo se recibe alterado como bit.

- *Error de reconocimiento (Acknowledgement Error):* Se produce cuando ningún nodo cambia a dominante el bit de reconocimiento.

Si un nodo advierte alguno de los fallos mencionados, iniciará la transmisión de una trama de error. El protocolo CAN especifica diversos fallos en la línea física de comunicación, como por ejemplo línea desconectada, problemas con la terminación de los cables o líneas cortocircuitadas. Sin embargo, no especificará cómo reaccionar en caso de que se produzca alguno de estos errores.

2.4.6.2.-AISLAMIENTO DE NODOS DEFECTUOSOS

Para evitar que un nodo en problemas condicione el funcionamiento del resto de la red, se han incorporado a la especificación CAN medidas de aislamiento de nodos defectuosos que son gestionadas por los controladores. Un nodo pude encontrarse en uno de los tres estados siguientes en relación a la gestión de errores:

- *Error Activo (Error Active):* Es el estado normal de un nodo. Participa en la comunicación y en caso de detección de error envía una trama de error activa.

- *Error pasivo (Error Passive):* Un nodo en estado de error pasivo participa en la comunicación, sin embargo ha de esperar una secuencia adicional de bits recesivos antes de transmitir y sólo puede señalar errores con una trama de error pasiva.

- *Anulado (Bus Off):* En este estado, deshabilitará su transceptor y no participará en la comunicación.

La evolución entre estos estados se basa en dos contadores incluidos en el controlador de comunicaciones. Contador de errores de transmisión (TEC) y Contador de errores de recepción (REC).

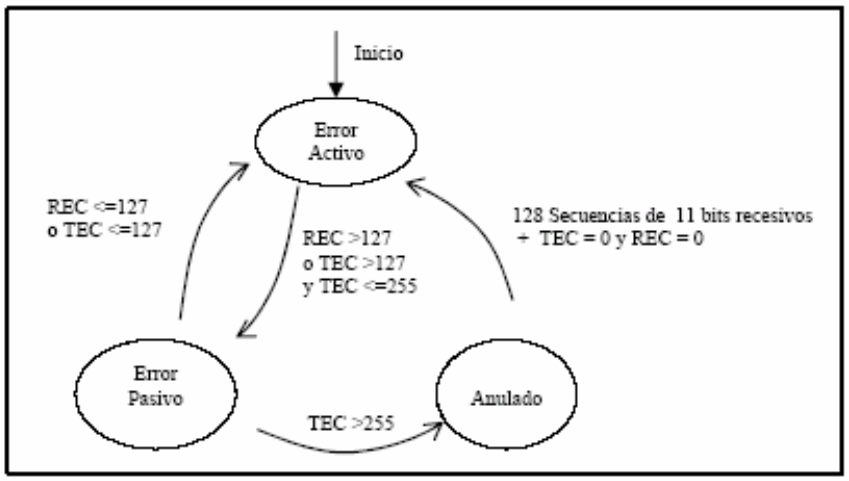


Figura 19 – Cap.2.4.6.2: Evolución entre estados de error

2.4.7.-ESPECIFICACIONES

CAN tiene dos formatos diferentes para transmitir datos: el formato de trama estándar (Standard Frame), según la especificación CAN 2.0A y, el formato de trama extendida (Extended Frame) según la especificación CAN 2.0B. La principal diferencia entre ambos es la longitud del identificador del mensaje (ID), que en el caso de la trama estándar es de 11 bits (2032 identificadores, ya que los 16 bits identificadores de menor prioridad están reservados) y en el caso de la extendida es de 29 bits (más de 536 millones de identificadores):

- *Controladores 2.0A:* únicamente transmiten y reciben mensajes en formato estándar. Si el formato es extendido se producirá un error.

- *Controladores 2.0B pasivos:* únicamente transmiten y reciben mensajes en formato estándar, pero admiten la recepción de mensajes en formato extendido, aunque los ignora posteriormente.

- *Controladores 2.0B activos:* transmiten y reciben mensajes en ambos formatos.

2.4.8.-IMPLEMENTACIONES

Existen tres tipos de arquitecturas en microcontroladores: Stand-Alone CAN controller, Integrated CAN Controller y Single-Chip CAN Node.

2.4.8.1.- *STAND-ALONE CAN CONTROLLER*

Es la arquitectura más simple, para llevar a cabo una comunicación en una red
CAN será necesario:

1. *Un microcontrolador* como puede ser el 16F877, con el que se ha venido trabajando a lo largo de todo este proyecto.

2. *Un controlador CAN* que introduzca el protocolo CAN, filtro de mensajes,… y todo el interface necesario para las comunicaciones. Un ejemplo de controlador CAN es el MCP2510 cuya DATA SHEET se puede encontrar en la página web de microchip.

3. *Un transceptor CAN*, esto es, un transmisor/receptor que puede ser por ejemplo el MCP2551 desarrollado por microchip.

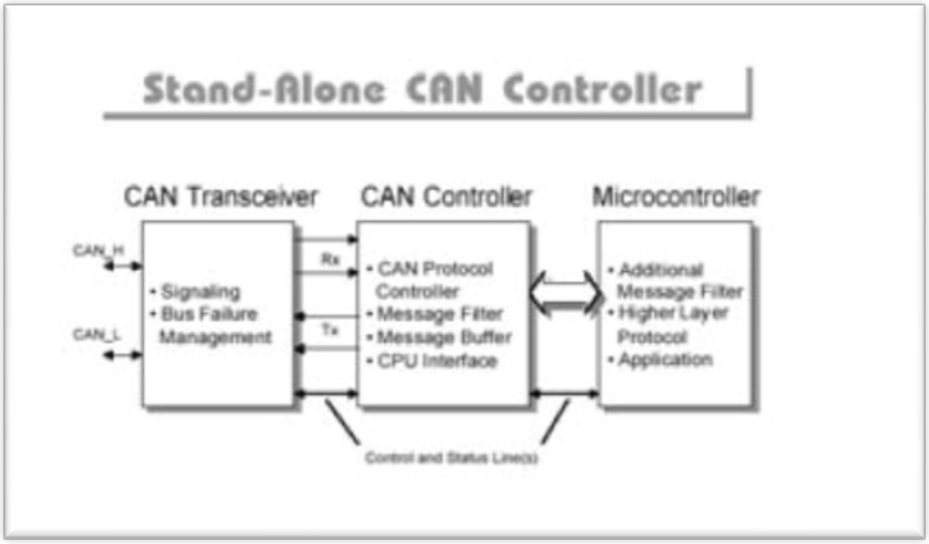


Figura 20 – Cap.2.4.8.2: Controlador Stand-Alone CAN

Así pues, si de alguna manera se pretende trabajar en una red CAN con una placa de pruebas, en un principio no sería factible a no ser que de alguna manera se le acoplasen el controlador y el transceptor CAN correspondiente.

Pero Microchip ha creado una placa de pruebas específica para este tipo de comunicaciones, que desarrolla además este tipo de arquitectura y que se puede conseguir también a través de la página web de microchip.

2.4.8.2.-*INTEGRATED CAN CONTROLLER*

Este tipo de arquitectura consiste en un microcontrolador que incluya, no sólo sus características propias sino además un módulo CAN con las características de un microcontrolador CAN. El transceptor se sitúa de manera separada. Este es el caso de nuestro microcontrolador dSPIC30F4013. El módulo del transceptor actúa como un controlador de línea, balanceando la señal, esto es, adecuando los niveles de tensión de esta al exterior.

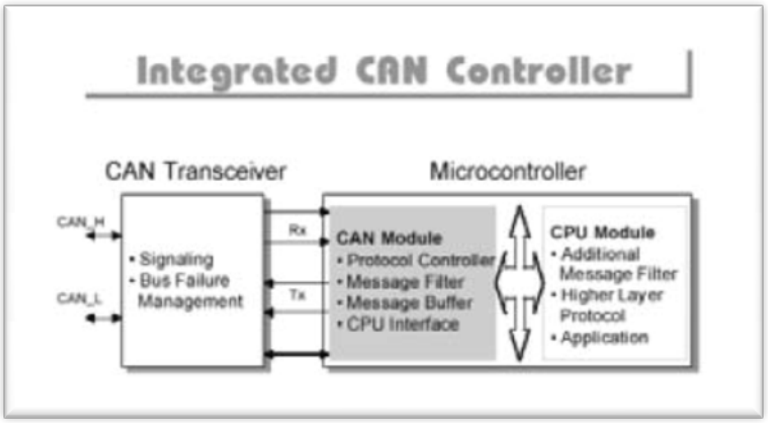


Figura 21 – Cap.2.4.8.3: Integrated CAN Controller

2.4.8.3.- ***SINGLE-CHIP CAN NODE***

Se trata de un chip que incluye en su interior los tres elementos necesarios para llevar a cabo las comunicaciones en un entorno CAN.

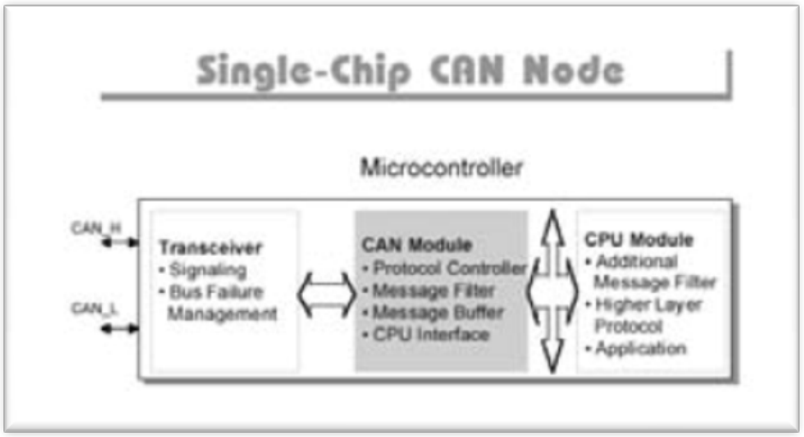


Figura 22 – Cap.2.4.8.3: Single-Chip CAN Node

2.5.-COMPETENCIA Y FUTURO

Realmente CAN se encuentra en una posición privilegiada en el mercado. El siguiente gráfico traza una comparativa entre diferentes tecnologías respecto al coste por nodo.

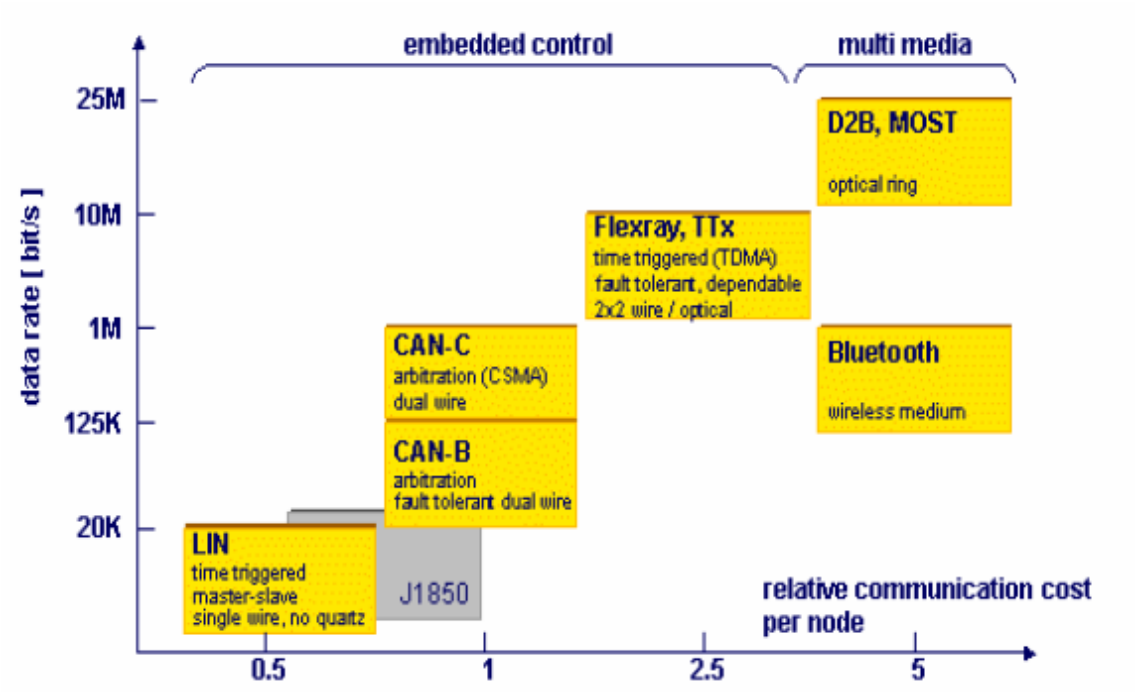


Figura 23 – Cap.2.5: Costes por nodo de diferentes tecnologías

Vemos que J1850, pero sobretodo LIN, tiene una mayor presencia en la industria, son opciones más económicas.

En la siguiente tabla, se exponen las diferentes características de LIN y CAN (y de cómo extra el I2C de “Philips”), para detallar la similitud entre ellas:

	CAN	LIN	I2C
Compañía que lo desarrolló	Bosch	Open source	Philips
Velocidad	1Mb/s	20Kb/s	0.1Mb/s / 0.4Mb/s
Tamaño de datos	64bits	8bits	8bits
Prioridad de mensajes	Si	No	No
Garantía de latencia	Si	***	No
Flexibilidad en la configuración	Si	***	Si
Sistema Multimaestro	Si	No	Si
Detección y señalización de errores	Si	Si	Si
Retransmisión de tramas	Automática	No	Programable

Tabla 2 – Cap.2.5: Características LIN y CAN

Se puede llegar a la conclusión fácilmente, de que cada una va a ser útil en un ámbito de manufactura distinto, ya que sus características son distintas. La diferencia principal, además de la económica, va a resultar ser la velocidad.

Dentro de la industria automovilística por ejemplo, CAN es empleado como bus de comunicaciones para los elementos más importantes de este, que van a requerir una seguridad y velocidad mayores. Para otro tipo de vicisitudes, como los elevalunas eléctricos por ejemplo, que tienen mucha menos importancia que la que puedan tener los frenos o el estado del motor, se emplea el bus LIN.

Uso general del bus CAN en los coches:

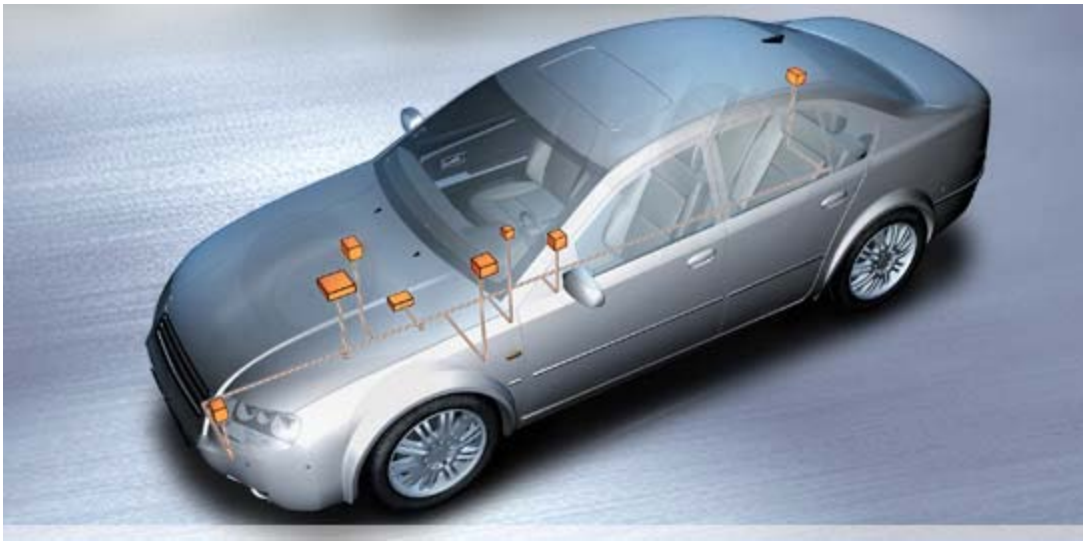


Figura 24 – Cap.2.5: Uso del bus CAN en coches

Uso general del bus LIN en los coches:

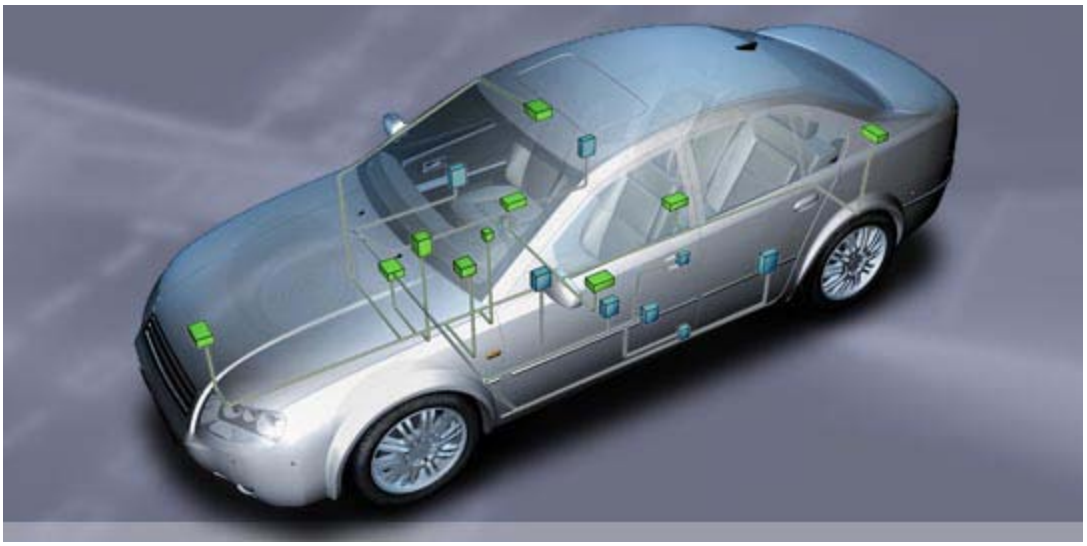


Figura 25 – Cap2.5: Uso del bus LIN en coches

A continuación vamos a nombrar algunas de las diferencias más importantes que hay entre estos, aunque no sean competidores directos:

CAN – Ethernet

Aún cuando las similitudes entre CAN y Ethernet son obvias, CAN posee algún beneficio clave cuando se compara con el protocolo Ethernet. El esquema de arbitraje que usa el protocolo CAN es de bit inteligente no destructivo. Esto significa que se comparan los mensajes con cada bit en un momento determinado, pero el mensaje con la prioridad más alta no se destruye y se retransmite; sólo el mensaje que no gana el arbitraje de bus se detiene y se retransmite. Éste es un punto importante que ayuda a minimizar el tiempo de fuera de servicio del bus y aumentan al máximo uso eficaz del ancho de banda disponible.

Otra ventaja importante del arbitraje del bit inteligente no destructivo, que usa CAN, es el hecho que esto da al bus características muy predecibles. Con Ethernet, ambos transmisores se detienen cuando se detecta una colisión y una cantidad de tiempo aleatorio se permite pasar antes de que ambos prueben de la retransmisión. Con este elemento aleatorio de tiempo eliminado de la función del bus, es posible lograr casi el 100% de eficacia en términos de utilización del ancho de banda.

Futuro

A día de hoy, el futuro de CAN se prevé esperanzador, ya que incluso las estimaciones más conservadoras coinciden en que la presencia de CAN en el mercado y en diversos campos de la industria, va a seguir en aumento durante los próximos diez o quince años.

2.6.-CAMPOS DE APLICACIÓN

El 80% de las aplicaciones modernas del bus CAN se pueden encontrar en la ingeniería del automóvil y otro tipo de vehículos como autobuses, trenes y aviones. En el caso de los automóviles por ejemplo, CAN es el encargado de la comunicación y automatización del sistema de freno, los faros, el ABS, o el ordenador de abordo, del cual vemos su esquema en el siguiente gráfico:

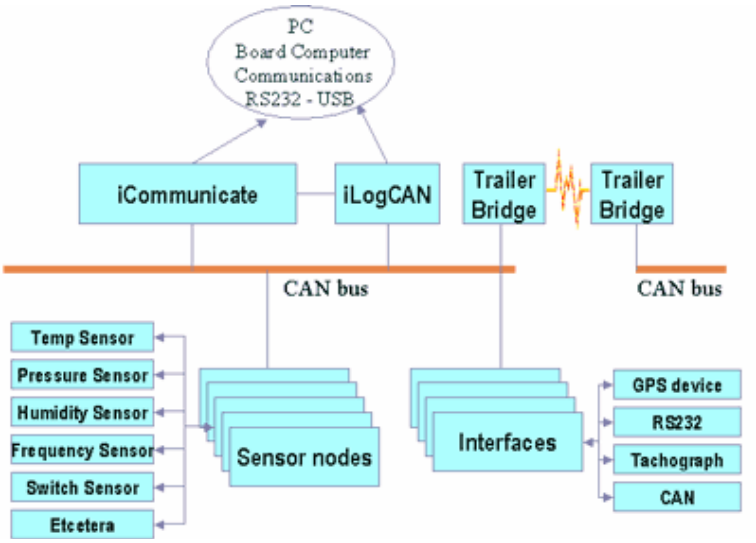


Figura 26 – Cap.2.6: Comunicación CAN en un automóvil

Sin embargo, también se puede encontrar el bus CAN en aplicaciones de diversa índole debido a su naturaleza, que le aporta robustez, economía y un altísimo grado de seguridad y fiabilidad; entre las más comunes:

Control y automatización industrial:

- Redes entre diversas máquinas y elementos de las mismas.
- Redes de supervisión.
- Redes de seguridad.

Control y automatización de edificios:

- Control de ascensores, puertas mecánicas, aspersores y diversos elementos mecánicos.
- Control de iluminación.

Aplicaciones específicas:

- Control de máquinas expendedoras (en Inglaterra está muy extendido su uso).
- Control de equipamiento médico.
- Control de sistemas automáticos de almacenaje.
- Control de electrodomésticos.

3.- DESCRIPCIÓN GENERAL DEL SISTEMA USB

3.1.- INTRODUCCIÓN

Para que podamos almacenar, transmitir o recibir nuestras tareas ordinarias, que cada vez suelen ser de mayor tamaño, aparecen en el mercado periféricos y componentes de mayor capacidad, discos duros con más Gigas, tarjetas de video más potentes, procesadores más rápidos, etc. Pero también necesitamos que la "vía" por la que se van a transmitir esos datos sea más fiable y más rápida, de forma que se ajuste a los avances producidos.

Por esta razón, aparecen nuevos medios físicos más rápidos, en un principio teníamos la interfaz serie y paralelo, pero era necesario unificar todos los conectores creando uno más sencillo y de mayores prestaciones. Así nació el **USB (Universal Serial Bus)** una de las mayores revoluciones en la computación, una tecnología que dejó completamente en el olvido la forma de interconectar periféricos a las computadoras, la expansibilidad, la sencillez de configuración y uso del hardware.

La documentación técnica relacionada con USB es realmente abundante y de gran profundidad, no solamente informática, sino también eléctrica, electrónica y mecánica. Por esta razón es altamente recomendable analizarla detalladamente, desde sus orígenes y la motivación para su actual existencia.

El Bus Universal en Serie (USB) consiste en una norma para bus periférico desarrollada por las industrias de computadoras y telecomunicaciones (Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC y Philips). Un computador con USB permite adjuntar dispositivos periféricos rápidamente sin necesidad de reiniciar ni de volver a configurar el sistema. Los dispositivos con USB se configuran automáticamente una vez que han sido conectados físicamente. La computadora cuenta normalmente con dos conectores USB, pero se pueden unir más dispositivos con USB a la computadora mediante adaptadores.

3.2.- ANTECEDENTES

Este capítulo presenta una breve descripción de los antecedentes del Bus Serial Universal (USB), incluyendo metas de diseño, herramientas del bus y tecnologías existentes.

3.2.1.- OBJETIVOS DEL USB

El USB está especificado para ser una extensión estándar para la industria de arquitectura PC con un enfoque en periféricos para PC que

habilita aplicaciones para consumidores y negocios. Los siguientes criterios se aplicaron en la definición de la arquitectura del USB:

- Fácil de usar por los periféricos del PC.
- Solución de bajo costo que soporte rangos de transferencia de 480 Mb/s.
- Full soporte para datos de voz, audio y video en tiempo real.
- Protocolo flexible para modo mixto de transferencia isocrónica y mensajería asincrónica.
- Comprensión de varias configuraciones PC y factores de forma.
- Provisión de una interface estándar.
- Habilitación de nuevas clases de dispositivos que aumenten la capacidad de los PC.
- iFull compatibilidad de retroalimentación de los dispositivos USB con las versiones previas de la especificación.

3.2.2. TAXONOMÍA

La siguiente figura describe una taxonomía para el rango de carga de trabajo para el tráfico de datos que puede ser servido a través del USB. Como puede verse, un bus de 480 Mb/s comprende los rangos de datos de velocidad alta, full velocidad y baja velocidad. Típicamente, los tipos de datos de alta velocidad y full velocidad pueden ser isocrónicos, mientras los datos de baja velocidad vienen de herramientas interactivas. El USB es principalmente un bus de PC pero puede ser fácilmente aplicado a otras herramientas computacionales como Host-centric. La arquitectura del software para la futura extensión del USB provee soporte para múltiples Controladores Host USB.

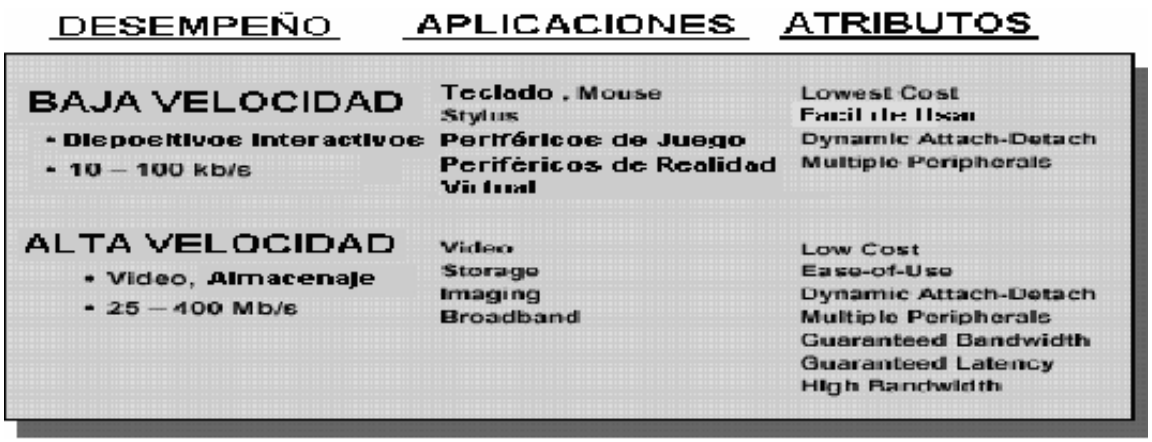


Figura 27 – Cap.3.3.2: Taxonomía del Espacio de Aplicación.

3.2.3. LISTAS DE HERRAMIENTAS

La especificación USB provee una selección de atributos que puede conseguir múltiples puntos de integración precio/desempeño y puede habilitar funciones que permiten diferenciación del nivel de sistema y componente. Las herramientas son categorizadas por los siguientes beneficios:

Fácil de usar por el usuario final

- Modelo simple de cables y conectores.
- Detalles eléctricos aislados del usuario final.
- Periféricos auto-identificables, mapeo automático de la configuración y la función del controlador.

Rango ancho de carga de trabajo y aplicaciones

- Apropiado para dispositivos con rangos de ancho de banda de unos Kb/s a varios miles Mb/s.
- Soporta tipos de transferencia isocrónicas tan bien como asincrónicas con los mismos conductores.
- Soporta operación concurrente de muchos dispositivos (conexiones múltiples).
- Soporta hasta 127 dispositivos físicos.
- Soporta transferencia de múltiple flujo de datos y mensajes entre el host y los dispositivos.
- Permite dispositivos compuestos (periféricos compuestos de muchas funciones).
- Disminuye el protocolo aéreo, resultando en alta utilización del bus.

Ancho de Banda Isocrónica

- Ancho de banda garantizado y bajo estado latente apropiado para telefonía, audio, vídeo, etc.

Flexibilidad

- Soporta un ancho rango de tamaños de paquete, que permite un rango de opciones de herramientas de memoria.
- Permite un ancho rango de herramientas promedio de datos por paquetes.
- Control de flujo por encabezamiento de memoria dentro del protocolo.

Robustez

- El mecanismo de recuperación de error de encabezamiento/falla está dentro del protocolo.
- La inserción y remoción dinámica de dispositivos es identificada en tiempo real.
- Soporte de identificación de dispositivos fallados.

Sinergia con la industria PC

- El protocolo es simple de implementar e integrar.
- Consistente con la arquitectura plug and play del PC.
- Ventajas con las interfaces de sistemas operativos existentes.

Bajo costo de implementación

- Bajo costo del subcanal a 1.5 Mb/s.
- Optimizado para integración entre periféricos y el hardware host.
- Utilizable por desarrolladores de periféricos de bajo costo.

- Uso cómodo de tecnologías.

Trayectoria de mejoramiento

- Arquitectura mejorada para soporte múltiple de controladores host USB en un sistema.

3.3.- ARQUITECTURA DEL USB

Este capítulo presenta un resumen de la arquitectura del Bus Serial Universal y conceptos claves. El USB es un cable bus que soporta el intercambio de datos entre un computador host y periféricos de banda ancha accesibles simultáneamente. Los periféricos adjuntados parten la banda ancha del USB a través de la programación del host, protocolo de prueba-base. El bus permite que los periféricos sean adjuntados, configurados, usados y detectados mientras el host y otros periféricos están en operación.

En los siguientes ítems se describen varios componentes del USB en gran detalle.

3.3.1.- DESCRIPCIÓN DEL SISTEMA USB

Un sistema USB es descrito por tres áreas definidas:

- Interconexión USB.
- Dispositivos USB.
- Host USB.

La interconexión USB es la manera en la que los dispositivos USB son conectados para una comunicación con el host. Esta incluye lo siguiente:

- Topología del Bus: Modelo de conexión entre los dispositivos USB y el host.

- Conexiones Inter-capas: En términos de capacidad de pila, las pilas USB que son hechas en cada capa en el sistema.

- Modelos de flujo de datos: La manera en que cada dato se mueve dentro del sistema entre los productores y los consumidores.

- Programación USB: El USB provee una interconexión partida. El acceso a la interconexión es programado en orden para soportar transferencia isocrónica de datos y para eliminar el arbitraje aéreo.

3.4.- TOPOLOGIA Y PROTOCOLOS

La estratificación del sistema USB está compuesto por tres áreas claramente demarcadas: (1) el host USB, (2) los dispositivos USB y, (3) toda la interconexión USB. La interconexión USB es la manera en la cual los dispositivos USB se conectan y comunican con el host, esto incluye: la topología del bus o el modelo de conexión entre los dispositivos USB y el host; los modelos de flujo de datos, es decir la forma en la que la información se mueve en el sistema entre los diversos elementos del mismo; la planificación USB que define la secuencia en la cual los dispositivos accederán al bus; finalmente, las relaciones entre capas del modelo, y las funciones de cada capa.

3.4.1.- TOPOLOGIA DEL BUS.DISPOSITIVOS USB

Los dispositivos USB pueden ser hubs que provean puntos de conexión adicionales a los existentes en el host, o bien diferentes dispositivos típicos periféricos. Es evidente que todos estos dispositivos deben tener la capacidad de soportar la especificación USB en cuanto a protocolos de comunicación se refiere, operaciones USB, configuración y reseteo USB.

La topología del bus USB adopta forma de estrella y se organiza por niveles. En un bus USB existen dos tipos de elementos: **Anfitrión** (“host”) y dispositivos; a su vez los dispositivos pueden ser de dos tipos: **Concentradores** y **Funciones.**

El **anfitrión** o “host”: A diferencia de los dispositivos y los hubs, existe tan solo un host dentro del sistema USB, que es el ordenador mismo, particularmente una porción del mismo denominado Controlador USB del Host. Este tiene la misión de hacer de interfaz entre el micro mismo y los diferentes dispositivos. Existen algunas particularidades respecto a este controlador. Su implementación es una combinación de hardware y software todo en uno, es decir firmware. Puede proveer de uno o dos puntos de conexión iniciales, denominados Hub Raíz, a partir de los cuales y de forma ramificada irán conectándose los periféricos.

El Host USB trabaja con los diferentes dispositivos valiéndose del Controlador de Host compuesto por una parte de hardware y otra de software, de esta forma conjunta el host es responsable al nivel de hardware, de los siguientes aspectos dentro del sistema UBS:

1. Detectar tanto la conexión de nuevos dispositivos USB al sistema como la atención de aquellos ya conectados, y por supuesto, configurarlos y ponerlos a disposición del usuario, tarea que involucra acciones por software.

2. Administrar y controlar el flujo de datos entre el host y los dispositivos USB, es decir el movimiento de información generada por el usuario mismo.

3. Administrar y regular los flujos de control entre el host y los dispositivos USB, es decir la información que se mueve con el objeto de mantener el orden dentro de los elementos del sistema.

4. Recolectar y resumir estadísticas de actividad y estados de los elementos del sistema.

5. Proveer de una cantidad limitada de energía eléctrica para aquellos dispositivos que pueden abastecerse con tan solo la energía eléctrica proveniente desde el ordenador (el teclado y el ratón son dos ejemplos claros).

Por otra parte, al nivel de software las funciones del Controlador de Host se incrementan y complican:

1. Enumeración y configuración de los dispositivos conectados al sistema.

2. Administración y control de transferencias isocrónicas de información.

3. Administración y control de transferencias asincrónicas.

4. Administración avanzada de suministro eléctrico a los diferentes dispositivos.

5. Administración de la información del bus y los dispositivos USB.

Los **concentradores** (“Hubs”): Los hubs son elementos claves dentro de la arquitectura Conectar & Operar de USB. Adicionalmente, simplifican de gran manera la sencillez de la interconexión de dispositivos al computador. Son el centro de una estrella, y sirven para conectar con el sistema anfitrión, con otro hub o con una función. Cada hub puede proporcionar 500 mA de energía y de alimentación (+5 V.cc + 0.25V.Las Figuras muestran hubs USB disponibles en el mercado.

Bajo una óptica eléctrica y teleinformática, los hubs son concentradores cableados que permiten múltiples conexiones simultáneas. Su aspecto más interesante es la concatenación, función por la que a un hub se puede conectar otro y otro, ampliando la cantidad de puertos disponibles para periféricos.

El hub USB tiene la capacidad de detectar si un periférico ha sido conectado a uno de sus puertos, notificando de inmediato al Controlador de Host en el computador, proceso que inicia la configuración del equipo nuevo; adicionalmente, los hubs también son capaces de detectar la desconexión de un dispositivo, notificando al Controlador de Host que debe remover las estructuras de datos y programas de administración (drivers) del dispositivo retirado.

Otra de las funciones importantes de los hubs es la de aislar a los puertos de baja velocidad de las transferencias a alta velocidad, proceso sin el cual todos los dispositivos de baja velocidad conectados al bus entrarían en colapso. La protección de los dispositivos lentos de los rápidos ha sido siempre un problema serio dentro de las redes mixtas, como es USB.

El hub está compuesto por dos partes importantes: El Controlador del Hub y el Repetidor del Hub. El Repetidor del Hub tiene la función de analizar, corregir y retransmitir la información que llega al hub, hacia los puertos del mismo. Mantiene una memoria consistente en varios registros de interfaz que le permiten sostener diálogos con el host y llevar adelante algunas funciones administrativas además de las meramente operativas; mientras que el Controlador de Hub puede asemejarse a una pequeña CPU de supervisión de las múltiples funciones que deben desempeñar un hub.

Una **función** es, dentro de la terminología USB, todos los dispositivos que pueden ser conectados al bus USB, a excepción de los hubs. Es un dispositivo capaz de transmitir o recibir datos de información de control en un bus USB, suele conectarse como un dispositivo independiente enlazado por un cable de por lo menos 5 metros, a un puerto

del hub o directamente al sistema anfitrión.. El común denominador a todas las funciones USB es su cable y el conector del mismo, diseñado y fabricado atendiendo a las especificaciones del bus, por lo que no hay que preocuparse por la compatibilidad entre equipos de diferentes fabricantes. Son funciones típicas el ratón, el monitor, módem, etc. La Figura las ilustra adecuadamente.

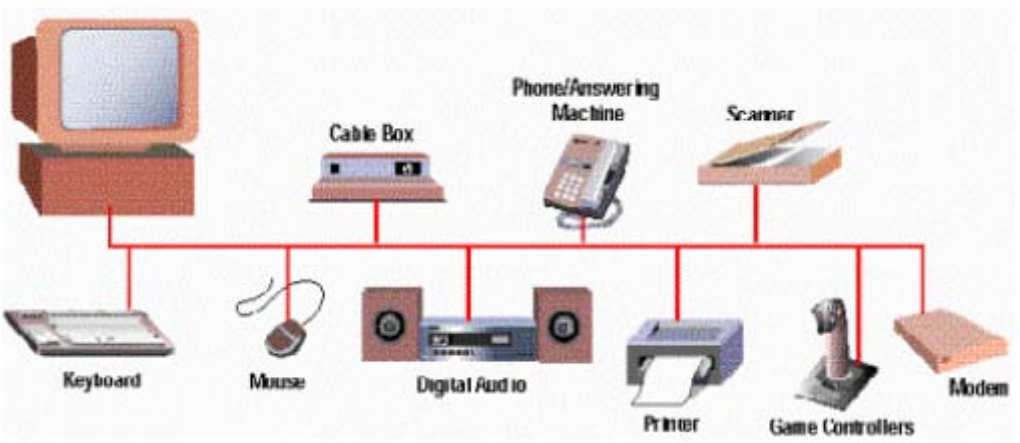


Figura 28 – Cap.3.4.1: Típicas funciones de un sistema USB

Un aspecto interesante de las funciones, es que pueden ser a su vez nuevos hubs. De hecho, la figura siguiente muestra un esquema en el que el PC tiene tres puertos, el monitor cuatro, el teclado tres y adicionalmente un hub propiamente, provee 4 puertos más. En un esquema tan sencillo, existen 14 puertos disponibles para todo tipo de periféricos, entre los que podemos citar: ratón, tablilla digitalizadora, lápiz óptico, teclado, impresora, un teléfono ISDN, etc.

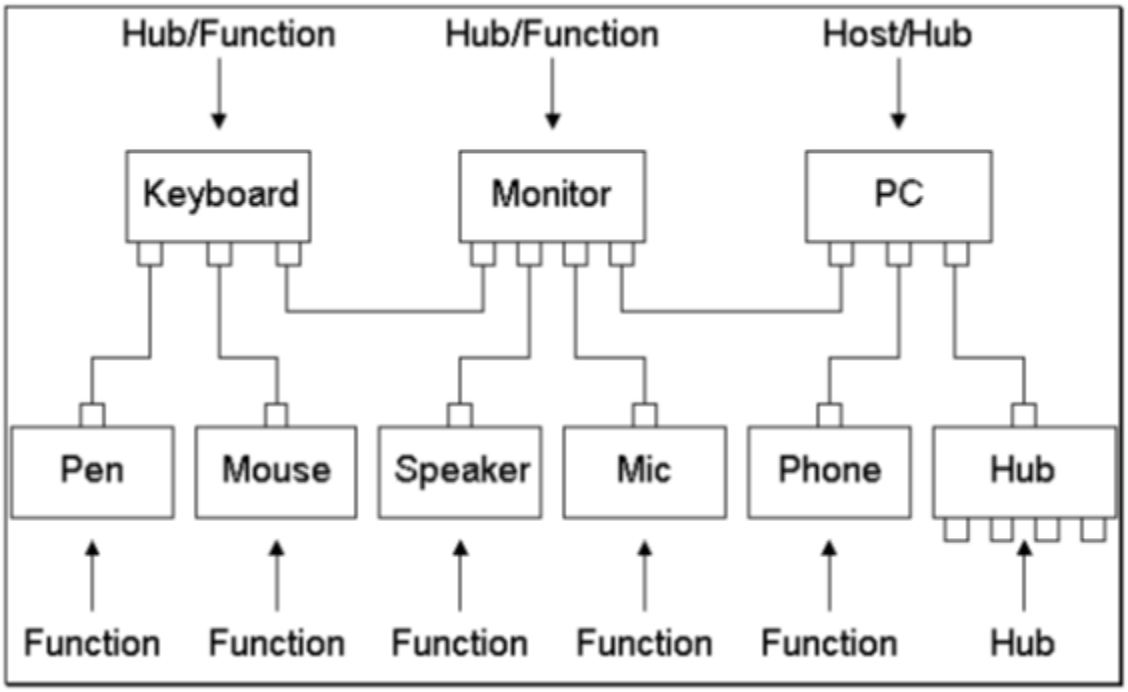


Figura 29 – Cap.3.4.1: Esquema interconexión USB

En general cada segmento del bus representa una conexión punto a punto de alguno de los tipos siguientes:

Sistema anfitrión↔ Función.

Sistema anfitrión↔ Concentrador.

Concentrador ↔ Concentrador.

Concentrador ↔ Función.

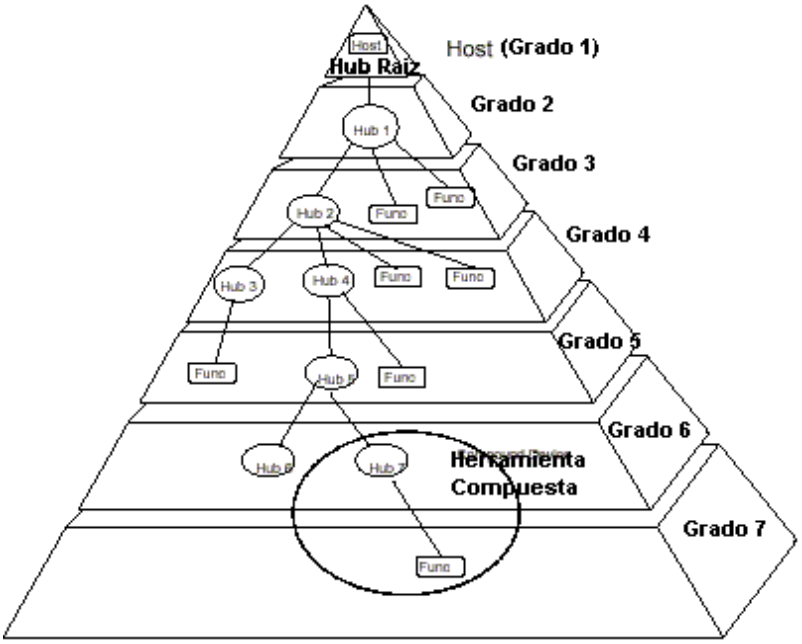


Figura 30 – Cap.3.4.1: Topología estrella del bus USB.

Hasta ahora todos estamos acostumbrados a ver en la parte posterior de nuestras computadoras dos puertos seriales, un puerto paralelo y quien sabe un puerto PS/2. La introducción del bus USB no marca un final para todos los dispositivos que poseen muchos usuarios con conectores RS-232 de 9 ó 25 pines.

Los PCs que estarán disponibles en el mercado seguirán manteniendo estos puertos con sus características habituales, sin embargo se plantea que a la larga desaparezcan poco a poco con el transcurrir de los años. De todas formas ya existen conversores tanto seriales y paralelos a USB.

3.4.1.1.- MODELO LOGICO FUNCIONAL

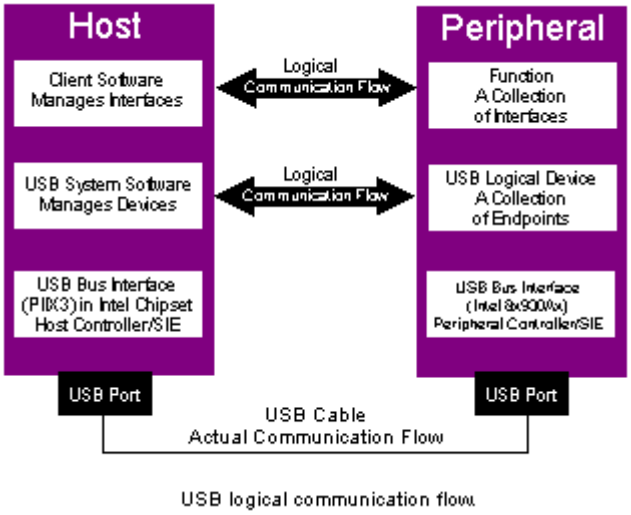


Figura 31 – Cap.3.4.1.1: Flujo de datos USB

El diagrama de la Figura ilustra el flujo de datos USB a partir de tres niveles lógicos: entre el Software Cliente y la Función, el Controlador USB y el dispositivo, y finalmente la capa física, donde la transmisión realmente sucede. Es importante entender que este modelo es muy parecido al OSI, el estándar de redes, y su comprensión radica en el hecho de que si bien existe un solo canal físico, pero los datos son manejados en cada punto por unidades homólogas, tal como si estuviesen sosteniendo una comunicación directa. Por esta razón se las denomina Capas Lógicas.

El nivel superior lógico es el agente de transporte de datos que mueve la información entre el Software Cliente y el dispositivo. Existe un Software Cliente en el host, y un Software De Atención al mismo en cada una de las funciones o periféricos USB. A este nivel, el host se comunica con cada uno de los periféricos en alguna de las varias formas posibles de transmisión que soporta USB. El Software Cliente solicita a los dispositivos diversas tareas y recibe respuestas de ellos a través de esta capa.

La capa lógica intermedia es administrada por el Software de Sistema USB, y tiene la función de facilitarles las tareas particulares de comunicación a la capa superior, cabe decir, administra la parte del periférico con la que la capa superior desea comunicarse, maneja la información de control y comando del dispositivo, etc. Su objetivo es permitir a la capa superior concentrarse en las tareas específicas tendientes

a satisfacer las necesidades del usuario, adicionalmente gestiona el control interno de los periféricos.

El acceso al bus es bajo la modalidad de Ficha o Token, lo que involucra siempre complejidad de protocolos, especialmente si agregamos dos velocidades posibles: 12Mbps ó 1.5Mbps. Todos estos algoritmos y procesos son administrados por el Host USB, reduciendo la complejidad del periférico, y lo más importante, el costo final de los dispositivos USB.

La capa física del modelo lógico USB comprende los puertos físicos, el cable, los voltajes y señales, el hardware y funcionamiento del hardware. Esta capa tiene el objetivo de liberar a las capas superiores de todos los problemas relacionados a la modulación, voltajes de transmisión, saltos de fase, frecuencias y características netamente físicas de la transmisión. Así que dejemos este punto a las empresas que fabrican los diferentes dispositivos de hardware USB.

3.4.2.- PROTOCOLO

3.4.2.1.- TRAMAS Y MICROTRAMAS

USB 1.x dividía el tiempo en tramas de 1 milisegundo. Adicionalmente, USB 2.0 define un tiempo de microtrama de 125 microsegundos.

Al igual que en USB 1.x se reservan ciertos porcentajes del tiempo de trama, para dar servicio a las distintas transacciones de Control, Interrupción e Isócronas full/low-speed, en USB 2.0 también se reservan ciertos porcentajes del tiempo de microtrama, para dar servicio a los distintos tipos de transacciones de alta velocidad.

3.4.2.2.- ENDPOINTS HIGH-SPEED Y HIGH BANDWIDTH

En USB 1.x, el tiempo máximo reservado en cada trama a cada endpoint Isócrono o de Interrupción es el de una transacción por trama.
USB 2.0 soporta dos tipos de endpoints Isócronos y de Interrupción:

- Endpoints de ancho de banda normal, que precisan hasta 1,024 bytes por microtrama (una transacción).

- Endpoints de alto ancho de banda (High Bandwidth), que precisan más de 1.024 bytes por microtrama, hasta un máximo de 3.072 bytes (3 transacciones).

3.4.2.3.- PROTOCOLO I: PAQUETES Y TRANSACCIONES

USB 2.0 mantiene la arquitectura centralizada definida en USB 1.x, en la que el host es el iniciador de todas las transferencias que se producen en el bus. Los dispositivos deben mantenerse a la espera hasta recibir del host un paquete especial (token) dirigido a él, indicando el tipo de transferencia a realizar. USB 2.0 también mantiene la estructura de la transacción, formada en base a la secuencia de paquetes Token-Dato-Validación (Handshake), definida en USB 1.x. Básicamente USB 2.0 añade algunos tipos de paquetes nuevos, para implementar las nuevas funciones y protocolos que se han incorporado.

En el grupo de paquetes de Token, siguen existiendo los definidos en USB 1.x (IN (direcciones de dispositivo transmisor de los datos), OUT (direcciones de dispositivo destinatario de los datos), SOF (indicador de comienzo y numeración de trama) y SETUP (direcciones de dispositivo destinatario de comandos de control)), aunque USB 2.0 reutiliza el tipo SOF:

- USB 1.x utiliza el token SOF para indicar el Principio de Trama (una vez cada milisegundo). USB 2.0 usa este mismo token para indicar el Principio de Microtrama (una vez cada 125 microsegundos).

En el grupo de paquetes de Datos, aparte de los dos tipos definidos en USB 1.x (DATA0 (paquete de datos par) y DATA1 (paquete de datos impar)), USB 2.0 define dos nuevos tipos:

- DATA2, empleado en transacciones isócronas de alta velocidad y alto ancho de banda.

- MDATA, empleado en transacciones isócronas de alta velocidad y alto ancho de banda y en transacciones Split (las transacciones se describen más adelante).

En el grupo de paquetes de Validación (Handshake), aparte de los tres tipos definidos en USB 1.x (ACK (recepción de paquete de datos libre de errores), NAK (receptor no puede aceptar datos o el transmisor no puede

enviar) y STALL (los dispositivos TX/RX detenidos o no pueden soportar un comando)), USB 2.0 define un nuevo paquete:

- NYET, significa “Not YET” (todavía no), y se emplea en los protocolos Split y de control de flujo PING.

Por último, en el grupo de paquetes Especial, USB 2.0 define tres nuevos tipos de paquetes, reutilizando uno de ellos (ERR) el código asignado por USB 1.x al paquete especial PRE. Se recuerda aquí la descripción del paquete PRE, para mostrar que no hay posibilidad de error en cuanto a que un mismo código identifique a dos paquetes distintos:

- PRE (Preámbulo): Definido en USB 1.x para indicar que a continuación se va a transmitir en modo baja velocidad. Sólo lo puede transmitir un host USB 1.x (en modo full-speed) inmediatamente antes del campo de token de la transacción low-speed.

- ERR (Error): Se usa para indicar errores en transacciones Split high-speed. Reutiliza el código del tipo de paquete PRE. Sólo lo puede transmitir un concentrador USB 2.0 (en modo high-speed) en el campo de validación (handshake) de una transacción Split.

- SPLIT: Lo transmite el host en el campo de token de una transacción Split.

- PING: Lo transmite el host en el campo de token de una transacción de control de flujo PING.

3.4.2.3.1.-PAQUETE DE TOKEN SOF Y NÚMERO DE MICROTRAMA

En USB 1.x, el número de la trama es un campo de 11 bits que el host incrementa una vez por trama, y que envía sólo en el paquete SOF (Start-Of-Frame) al principio de cada trama. USB 2.0 utiliza este mismo campo para indicar el número de microtrama. En este caso, el host no envía todos los bits del número de microtrama (14 bits), sino solamente los 11 bits más altos. Como cada trama (1 ms) contiene 8 microtramas (125 us), el efecto de lo anterior es que el host USB 2.0 envía durante 8 microtramas consecutivas un paquete SOF con el mismo número de trama. Los

dispositivos full-speed reciben un paquete SOF cada milisegundo con un número de trama secuencial. Los dispositivos high-speed reciben un paquete SOF cada 125 microsegundos, pero con el mismo número de trama en cada grupo de 8 consecutivos.

Los dispositivos high-speed que necesiten llevar un control del número de microtrama, pueden detectar cuándo un paquete SOF tiene un número de trama distinto del anterior, e iniciar un contador interno de microtrama (de 3 bits) a 0. Cada uno de los siguientes 7 paquetes SOF con el mismo número de trama serán los correspondientes a las microtramas 1 a 7.

3.4.2.3.2.- PAQUETES DE DATOS

USB 2.0 sigue haciendo un uso de los paquetes DATA0 y DATA1, como parte del mecanismo de detección de errores Data Toggle en los casos definidos en USB 1.x; adicionalmente USB 2.0 define los nuevos paquetes DATA2 y MDATA, de nuevo con la función de detección de errores en los nuevos casos definidos en USB 2.0.

Uso de los paquetes de datos en los nuevos casos definidos en USB 2.0 es:

- Las transacciones de Interrupción high-speed high bandwidth realizadas en la misma microtrama, hacen uso del mecanismo Data Toggle (uso alternativo de paquetes DATA0 y DATA1).

- En transacciones Isócronas-IN high-speed high-bandwidth, el dispositivo envía los datos utilizando las siguientes secuencias de paquetes:

  - En el caso de 1 transacción por microtrama, se usa el paquete DATA0.

  - En el caso de 2 transacciones por microtrama, se usa la secuencia de paquetes DATA1-DATA0.

  - En el caso de 3 transacciones por microtrama, se usa la secuencia de paquetes DATA2-DATA1-DATA0.

- En transacciones Isócronas-OUT high-speed high-bandwidth, el host envía los datos utilizando las siguientes secuencias de paquetes:

- En el caso de 1 transacción por microtrama, se usa el paquete DATA0.

- En el caso de 2 transacciones por microtrama, se usa la secuencia de paquetes MDATADATA1.

- En el caso de 3 transacciones por microtrama, se usa la secuencia de paquetes MDATAMDATA-DATA2.

- En transacciones Start-Split de Interrupción-OUT, el host alterna los paquetes DATA0 y DATA1 en la manera habitual (data toggle).

- En transacciones Complete-Split de Interrupción-IN, el concentrador también alterna los paquetes DATA0 y DATA1, siempre que la transacción en el bus full/low-speed se haya completado durante una microtrama. Cuando se da el caso de que la transacción en el bus full/low-speed abarca dos microtramas, la respuesta del concentrador en la primera transacción Complete-Split usa el paquete MDATA, para indicar al host que faltan datos por enviar, y que debe ejecutar otra transacción Complete-Split en la siguiente microtrama, para obtener el resto de los datos (que se transmitirán mediante un paquete DATA0 o DATA1, según corresponda en la secuencia Data Toggle).

- En transacciones Start-Split Isócronas-OUT, el host siempre usa el paquete DATA0.

- En transacciones Complete-Split Isócronas-IN, el concentrador usa el paquete MDATA para todos los paquetes excepto para el último, en que usa el paquete DATA0 (la transacción full-speed puede abarcar varias microtramas, por lo que los paquetes MDATA informan al host que debe seguir enviando transacciones Complete-Split en cada microtrama, hasta que reciba un paquete DATA0).

La máxima cantidad de información útil que se puede transferir en un paquete de datos depende de la velocidad del dispositivo y del tipo de transferencia.

3.4.2.3.3.-PAQUETE DE VALIDACIÓN (HANDSHAKE) NYET

Se utiliza sólo en modo high-speed, y se puede transmitir en dos circunstancias:

- Lo puede transmitir un endpoint high-speed tipo Bulk-OUT o de Control, como respuesta durante el protocolo de control de flujo PING.

- También lo puede transmitir un concentrador high-speed como respuesta a una transacción Split, cuando el concentrador todavía no ha concluido la transacción full/low-speed con el dispositivo, o bien cuando el concentrador no puede atender la transacción en ese momento.

3.4.2.3.4.- PAQUETE ESPECIAL DE HANDSHAKE ERR

Se utiliza sólo en modo high-speed, y lo puede transmitir un concentrador high speed en el campo de handshake de una transacción Complete-Split, para informar de un error en la transacción full/low-speed.

3.4.2.3.5.- PAQUETES ESPECIALES DE TOKEN PARA TRANSACCIONES SPLIT

Aparte de las transacciones definidas en USB 1.x (Control, Bulk, Interrupción e Isócronas), USB 2.0 define las transacciones Split. Sólo los controladores de host y los concentradores USB 2.0 deben soportar este nuevo tipo de transacción, ya que es invisible a los dispositivos.

Este tipo de transacción es el que permite la comunicación con dispositivos full/low speed conectados a concentradores que funcionan en modo high-speed. El host comienza una transacción Split cuando envía al concentrador, en modo high-speed, toda la información necesaria para que el concentrador ejecute ahora una transacción full/low-speed con el dispositivo. Toda la información queda almacenada en el concentrador, en el TT correspondiente al puerto al que está conectado el dispositivo (o en el único TT disponible si se trata de un concentrador mono-TT). Mientras el concentrador ejecuta esta transacción con el dispositivo full/lowspeed, el bus queda libre para ejecutar nuevas transacciones high-speed con otros dispositivos.

La respuesta del dispositivo full/low-speed queda almacenada a su vez en el TT del concentrador, disponible para cuando el host posteriormente indique al concentrador que la envíe, en modo high-speed.

USB 2.0 define el token especial SPLIT para llevar a cabo las transacciones Split. Este es el único paquete de token de 4 bytes, a diferencia de los paquetes de token normales de 3 bytes.

USB 2.0 define dos transacciones Split que hacen uso del token especial SPLIT:

- Start Split: La utiliza el host para enviar al concentrador, en modo high-speed, toda la información necesaria para que el concentrador ejecute ahora una transacción full/low-speed con el dispositivo.

- Complete Split: La utiliza el host para leer del concentrador, en modo high- speed, la respuesta del dispositivo full/low-speed.

Un campo en el propio paquete SPLIT identifica el tipo de transacción.

3.4.2.3.6.- PAQUETE ESPECIAL DE TOKEN PING PARA CONTROL DE FLUJO

Un caso bastante frecuente y que produce una gran pérdida de ancho de banda útil en un bus USB 1.x, se da cuando los dispositivos full/low-speed contienen endpoints tipo Bulk-OUT y de Control que necesitan un tiempo para procesar los datos recibidos, de forma que no pueden momentáneamente recibir nuevos datos hasta que se desocupe el buffer de recepción.

En USB 1.x, esta situación la controla el dispositivo devolviendo una validación (handshake) NAK en la transacción OUT en la que el host ha transmitido un nuevo paquete de datos. Esta validación indica que el dispositivo no ha podido recibir los datos porque no tiene espacio suficiente en su buffer de recepción, y el host debe reintentar la transmisión posteriormente. Desafortunadamente, para cuando el dispositivo informa que no tiene espacio, la mayor parte del tiempo de la transacción ya se ha consumido, ya que el paquete de datos se ha transferido íntegramente. Esto produce una pobre utilización del bus cuando se suceden múltiples transacciones OUT con respuesta negativa (NAK) por parte del dispositivo.

USB 2.0 define un nuevo protocolo de control de flujo más eficiente, denominado PING, que debe ser utilizado por las transferencias de tipo Bulk- OUT y de Control. Las transferencias de Control deben soportar este protocolo en las fases de Datos y de Estado, pero no en la fase de Setup. Este nuevo protocolo evita la transmisión de paquetes de datos hasta que el host sabe que el dispositivo puede aceptarlos.

El host pregunta al dispositivo high-speed mediante el paquete especial de token PING, y el dispositivo puede contestar con una validación ACK,

para indicar que tiene espacio para recibir un nuevo paquete de datos, o con un handshake NAK, para indicar que no tiene espacio. El host pregunta periódicamente mediante paquetes PING hasta que recibe un handshake ACK, en cuyo caso procede a la transmisión del paquete de datos.

Una vez que se produce la recepción de un paquete de datos, el dispositivo puede contestar con una validación ACK, para indicar la correcta recepción del paquete y que tiene espacio para el siguiente paquete, o con una validación NYET, para indicar la correcta recepción del paquete y que no tiene espacio para el siguiente paquete.

El host puede seguir transmitiendo paquetes de datos en tanto que el dispositivo siga contestando con ACK.

En el momento en que el dispositivo conteste con NYET, el host debe volver al proceso de preguntar al dispositivo mediante paquetes PING antes de enviar un nuevo paquete de datos.

3.4.2.4.- PROTOCOLO II: PIPES Y TRANSFERENCIAS

USB 1.x define las vías de comunicación entre las aplicaciones que se ejecutan en el host (clientes) y los distintas endpoints en los dispositivos USB (servidores), y las denomina “pipes”. Cuando un dispositivo USB se conecta a un sistema, y el sistema lo reconoce y lo configura, el dispositivo queda organizado como un cierto conjunto de endpoints de distintos tipos (existen 4 tipos de endpoints). Entonces el sistema establece todas las vias de comunicación (pipes) necesarias entre el sistema y cada uno de los endpoints disponibles en dicha configuración. El dispositivo puede implementar varias posibles configuraciones, con distintos conjuntos de de distintos tipos transferencias en cada una de ellas. El sistema elige una

cierta configuración en función de la funcionalidad particular que se precise del dispositivo.

Existen 4 tipos de endpoints (Bulk, Control, Interrupción e Isócrono) y 2 tipos de pipes (Control o Mensaje y Stream). Las posibles combinaciones son:

- Pipe de Control o Mensaje. Es una vía de comunicación bidireccional entre el host y dos endpoints de Control en un dispositivo USB. Un endpoints es de Salida y el otro es de Entrada, de forma que se pueda establecer la comunicación bidireccional. Todos los dispositivos USB disponen de dos endpoints de Control en la dirección 0, uno de entrada y uno de salida, de manera que el sistema siempre puede establecer una pipe de Control con el dispositivo, incluso antes de configurarlo (se denomina Pipe de Control por Defecto, y es la única pipe que se puede establecer antes de configurar al dispositivo). A través de esta pipe, el sistema puede leer del dispositivo toda la información descriptiva necesaria para enterarse del tipo de dispositivo, posibles configuraciones, protocolos que soporta, número y tipos de endpoints que soporta en cada posible configuración, etc. Esta información descriptiva son los Descriptores.

- Pipe Stream. Es una vía de comunicación unidireccional entre el host y un endpoints de los tipos Bulk, Interrupción o Isócrono. Si un dispositivo necesita transferencias bidireccionales de un tipo de endpoint concreto, el sistema debe establecer dos pipes, una de salida (con un endpoint de salida) y otra de entrada (con un endpoint de entrada).

Las diferencias básicas entre las transferencias USB 1.x y USB 2.0 son:

- Tamaños máximos de los paquetes de datos en cada tipo de transferencia.

- Reserva de tiempo de microtrama para las transferencias de Control, de Interrupción e Isócronas.

3.4.2.4.1.-TRANSFERENCIAS DE CONTROL

Las transferencias de Control proporcionan control de flujo y una entrega de datos garantizada y libre de errores.

Todos los dispositivos full, high y low-speed pueden incorporar endpoints de Control, y por lo tanto pueden hacer uso de las transferencias de Control. Todos implementan, al menos, un endpoint de salida y uno de entrada en la dirección 0, para poder establecer la Pipe de Control por Defecto.

Las transferencias de Control se componen de 3 transacciones denominadas Setup-Dato-Estado. Los tamaños máximos del paquete de datos durante la transacción de datos son:

- Full-speed: 8, 16, 32, ó 64 bytes.
- High-speed: 64 bytes.
- Low-speed: 8 bytes.

USB hace una gestión “best effort” para ir dando curso a las distintas transferencias de Control pendientes en cada momento en todas las pipes de Control establecidas con todos los dispositivos. Para ello se hace la siguiente reserva del tiempo de trama o microtrama:

- En un bus full/low-speed, la reserva es del 10% del tiempo de trama.
- En un bus high-speed, la reserva es del 20% del tiempo de microtrama.

Las reglas definidas por USB para el envío de las transferencias pendientes son:

- Si el tiempo de trama o microtrama utilizado por las transferencias de Control pendientes es inferior al reservado, el tiempo restante puede utilizarse para transferencias Bulk.
- Si hay más transferencias de Control pendientes que tiempo reservado, pero hay tiempo adicional en la trama o microtrama no consumido por transferencias de Interrupción o Isócronas, entonces el host puede utilizar dicho tiempo adicional para enviar nuevas transferencias de Control.

- Si hay más transferencias de Control pendientes que tiempo disponible en una trama o microtrama, el host selecciona cuáles se procesan, quedando el resto pendientes para una próxima trama o microtrama.

Los endpoints de Control high-speed soportan el protocolo de control de flujo PING en las transacciones de Dato y Estado de salida.

3.4.2.4.2.-TRANSFERENCIAS ISÓCRONAS

Las transferencias Isócronas están diseñadas para soportar aquellos dispositivos que precisan una entrega de datos a velocidad constante, y en la que no importa la pérdida eventual de información. Esto es necesario para aplicaciones en que la información de tiempo va implícita en la propia velocidad de transmisión/recepción de datos (isocronismo).

Para ello, las transferencias Isócronas proporcionan:

- Ancho de banda garantizado.
- Latencia limitada.
- Velocidad de transferencia de datos constante garantizada a través de la pipe.
- En caso de error en la entrega, no se reintenta la transmisión.
- Sin control de flujo.

Sólo los dispositivos high y full-speed pueden incorporar endpoints Isócronos.

Las transferencias Isócronas se componen sólo de transacciones de datos. Las frecuencias y los tamaños de los paquetes de datos son:

- Full-speed: 1 transacción por trama de hasta 1,023 bytes.
- High-speed: 1 transacción por microtrama de hasta 1,024 bytes.

- High-speed high-bandwidth: 2 ó 3 transacciones por microtrama de hasta 1,024 bytes cada una.

La gestión que hace USB para garantizar las transferencias es la de establecer o no la pipe en función de que haya suficiente tiempo libre de trama o microtrama para realizarlas. Para ello, los endpoints Isócronos indican qué cantidad de información como máximo debe transferir la pipe en cada trama o microtrama, de forma que el sistema USB puede calcular si hay suficiente tiempo o no para acomodar la pipe, y en función de eso la establece o no.

La reserva de tiempo de trama o microtrama para acomodar transferencias Isócronas y de Interrupción es como máximo el tiempo no reservado para transferencias de Control. El sistema USB puede ir estableciendo pipes Isócronas y de Interrupción con distintos dispositivos hasta agotar dicha reserva:

- Full-speed: Hasta un 90% del tiempo de trama.

- High-Speed: Hasta un 80 % del tiempo de microtrama.

3.4.2.4.3.-TRANSFERENCIAS DE INTERRUPCIÓN

Las transferencias de Interrupción están diseñadas para soportar aquellos dispositivos que precisan enviar o recibir datos de manera no frecuente, pero con ciertos límites de latencia.

Para ello, las transferencias de Interrupción proporcionan:

- Tiempo máximo de servicio (latencia) garantizado.

- Reintento de transferencia en el siguiente periodo, en caso de eventual fallo en la entrega.

Todos los dispositivos high, full y low-speed pueden incorporar endpoints de Interrupción.

Las transferencias de Interrupción se componen sólo de transacciones de datos. Los tamaños de los paquetes de datos son:

- Low-speed: hasta 8 bytes.

- Full-speed: hasta 64 bytes.
- High-speed: hasta 1,024 bytes.
- High-speed high-bandwidth: 2 ó 3 transacciones por microtrama de hasta 1,024 bytes cada una.

La gestión que hace USB para garantizar las transferencias es la de establecer o no la pipe en función de que haya suficiente tiempo libre de trama o microtrama para realizarlas. Para ello, Las transferencias de Interrupción indican qué cantidad de información como máximo debe transferir la pipe en cada transacción, así como el tiempo máximo entre transacciones, de forma que el sistema USB puede calcular si hay suficiente tiempo o no para acomodar la pipe, y en función de eso la establece o no.

El tiempo máximo entre transacciones (tiempo de latencia máximo) especificado por cada dispositivo puede ser:

- Low-speed: de 10 a 255 ms.
- Full-speed: de 1 a 255 ms.
- High-speed: de 125 us a 4´096 seg.

La reserva de tiempo de trama o microtrama para acomodar transferencias Isócronas y de Interrupción es como máximo el tiempo no reservado para transferencias de Control. El sistema USB puede ir estableciendo pipes Isócronas y de Interrupción con distintos dispositivos hasta agotar dicha reserva:

- Full y Low-speed: Hasta un 90% del tiempo de trama.
- High-Speed: Hasta un 80 % del tiempo de microtrama.

3.4.2.4.4.-TRANSFERENCIAS BULK

Las transferencias Bulk están diseñadas para soportar aquellos dispositivos que precisan enviar o recibir grandes cantidades de datos, con latencias que pueden tener amplias variaciones, y en que las transacciones

pueden utilizar cualquier ancho de banda disponible. Para ello, las transferencias Bulk proporcionan:

- Acceso al bus en función del ancho de banda disponible.
- Reintento de transferencias en caso de errores de entrega.
- Entrega garantizada de datos, pero sin garantía de latencia máxima ni de ancho de banda.

Las transferencias Bulk se realizan relati vamente rápidas si el bus dispone de mucho ancho de banda libre, pero en un bus USB con mucho ancho de banda reservado, pueden alargarse durante periodos de tiempo relativamente grandes.

Sólo los dispositivos high y full -speed pueden incorporar endpoints Bulk. Las transferencias Bulk se componen sólo de transacciones de datos. Los tamaños de los paquetes de datos son:

- Full-speed: 8,16, 32 y 64 bytes.
- High-speed: 512 bytes.

USB hace una gestión “good effort” para ir dando curso a las distintas transferencias pendientes en cada momento en todas las pipes Bulk establecidas con todos los dispositivos. Las transferencias de Control tienen preferencia sobre las Bulk, por lo que las transferencias Bulk se realizan siempre que no haya otro tipo de transferencias que hacer en una trama o microtrama.

Las endpoints Bulk-OUT high-speed soportan el protocolo de control de flujo PING.

4.-DESCRIPCIÓN GENERAL DEL PIC32MX575F512H

4.1.- INTRODUCCIÓN

Para que nuestro dispositivo sirviese para comunicar un bus CAN con un PC a través del USB del mismo, pensamos en usar un microprocesador que tuviese la capacidad de realizar las dos funciones (CAN y USB simultáneamente). El microprocesador escogido es el PIC32MX575F512H de Microchip que además de estas dos funciones tiene una amplia memoria y otras muchas características que a continuación detallamos más ampliamente.

4.2.- CARACTERÍSTICAS PRINCIPALES

- Alto rendimiento de 32-bit RISC CPU:

  a) Frecuencia máxima: 80 MHz.
  b) Módulo de Caché de Precarga para acelerar la ejecución desde la memoria Flash.

- Características del microcontrolador:

  a) Rango de voltaje de 2.3V a 3.6V.
  b) Memoria flash de 64K a 512K, además de un plus adicional de 12 KB de Flash de arranque.
  c) De 16K a 128K de memoria SRAM.
  d) Compatibilidad con la mayoría de dispositivos DSC PIC24/dsPIC.

- Características de periféricos:

  a) Hasta 8 canales de hardware DMA con detección automática de tamaño de datos.
  b) USB 2.0 de máxima velocidad del dispositivo y de controlador On-The-Go (OTG).
     - Canales DMA dedicados.
  c) Módulo Can 2.0B con canales DMA dedicados.
  d) Cristal de 3 MHz a 25 MHz.
  e) Osciladores internos de 8 MHz y 32 KHz.
  f) PLLs separados para los relojes de la CPU y el USB.
  g) Pines I/O de alta velocidad, capaces de conmutar a velocidades de hasta 80 MHz.

- Características de depuración (debug):

  a) Dos interfaces de programación y depuración:
     - 2-hilos interfaz con acceso no invasivo e intercambio de datos en tiempo real con la aplicación.
     - 4-hilos MIPS con interfaz JTAG estándar mejorado.

- Funciones analógicas:

  a) Hasta 16 canales, con convertidor analógico-digital de 10 bits.
  b) 2 comparadores analógicos.

Dispositivo	Pins	Memoria de Programa (Kb)	Memoria de Datos (Kb)	USB	CAN	Canales DMA (Programabl	10-bit 1 Msps ADC	Comparador	JTAG	Encapsulado
PIC32MX575 F512H	64	512+12[(1)]	64	1	1	8/4	16	2	Sí	TQFP/QFN

Tabla 3 – Cap.4.2: Caracterísitcas del PIC32MX575F512H

4.3.- DIAGRAMA DE PINES

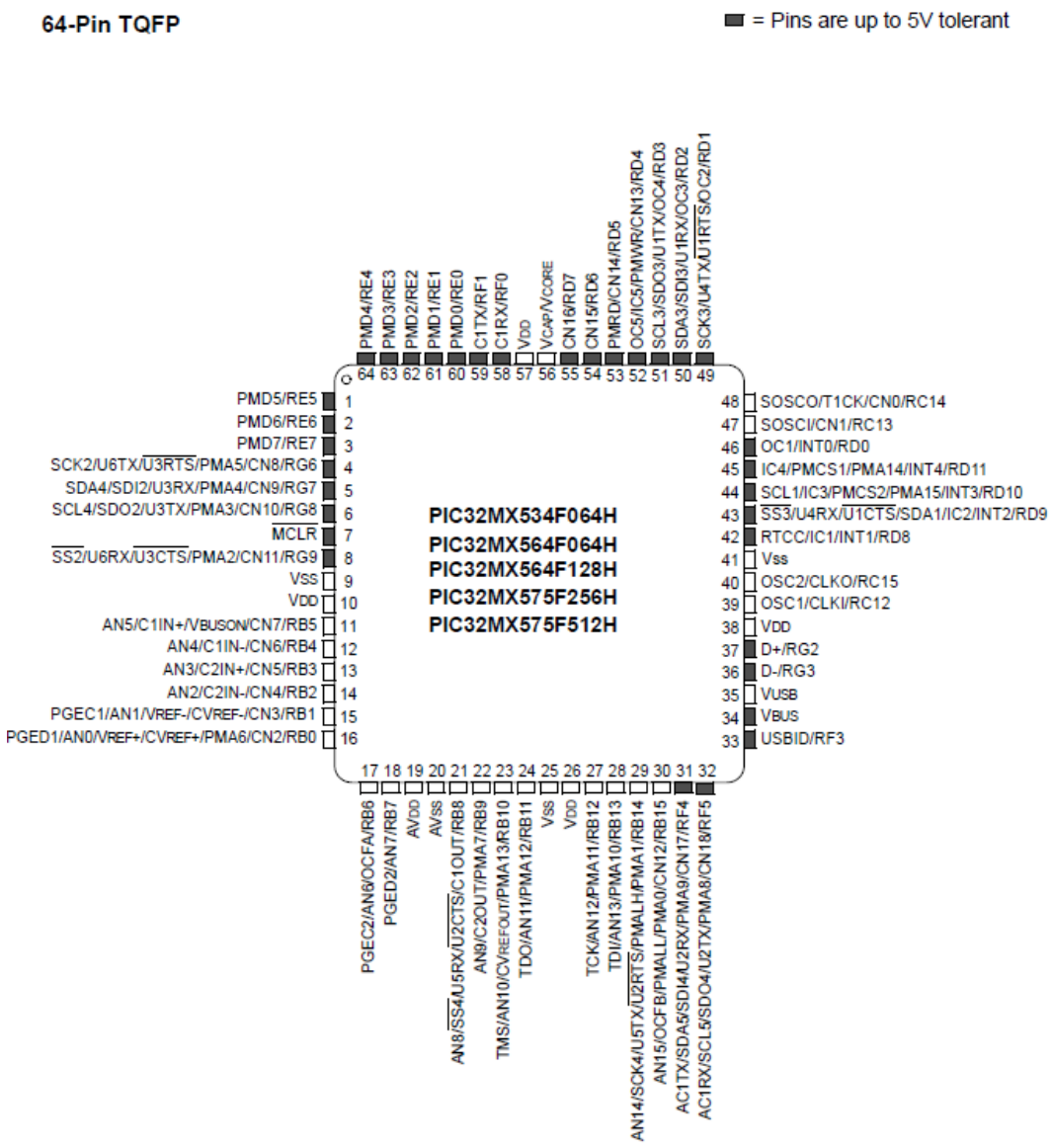


Figura 32 – Cap.4.3: Diagrama de pines del PIC 32MX575F512H

4.4.- SISTEMA USB DEL PIC32MX575F512H

El módulo USB contiene componentes analógicos y digitales para proporcionar un host encajado de velocidad completa (full-speed) USB 2.0 y un host encajado de poca velocidad (low-speed) USB 2.0, un dispositivo de velocidad completa o una implementación OTG (On The Go) con un mínimo de componentes externos. Este módulo en modo Host está previsto para usarse como host encajado y por lo tanto no implementa un controlador UHCI o OHCI.

El módulo USB consiste en el reloj generador, los comparadores de voltaje USB, el transceptor, el interface serie motor (Serial Interface Engine, SIE), un controlador dedicado USB DMA, resistores pull-up y pull-down, y la interface de registro.

El reloj generador proporciona 48MHz requeridos para la comunicación a velocidad completa y poca velocidad USB. Los comparadores de voltaje visualizan el voltaje en el pin VBUS para determinar el estado del bus. El transceptor proporciona traducción analógica entre el bus USB y la lógica digital. El SIE (Serial Interface Engine) es una máquina de estado que transfiere datos desde y hasta el buffer final y genera el protocolo hardware para transferir datos. El controlador USB DMA transfiere datos entre los buffers de datos en la RAM y en el SIE. Los resistores integrados pull-up y pull-down eliminan la necesidad de componentes externos de señal. El registro interface permite a la CPU configurar y comunicarse con el módulo.

El módulo PIC 32 USB incluye las siguientes características:

- Soporte de velocidad completa para host y dispositivo.
- Soporte de poca velocidad para host.
- Soporte USB OTG (On The Go).
- Resistores de señalización integrados.
- Comparadores analógicos integrados para el VBUS de supervisión.
- Transceptor integrado USB.
- Formalización de transacción realizada por hardware.

- Buffering de punto final en cualquier parte del sistema RAM.
- DMA integrado para acceder al sistema RAM y Flash.

4.4.1.-MODOS DE FUNCIONAMIENTO

4.4.1.1.-MODO HOST

El host es el maestro en un sistema USB y es responsable de la identificación de todos los dispositivos conectados a él, iniciar todas las transferencias, asignación del ancho de banda del bus y suministrar energía a cualquier dispositivo USB conectado directamente a él.

- Host USB Standard: En modo Host USB Standard, las siguientes características y requerimientos son relevantes:

  - Soporta gran variedad de dispositivos.
  - Soporta todo tipo de transferencias USB.
  - Soporta hubs USB (permite la conexión de múltiples dispositivos simultáneamente)
  - Los drivers de los dispositivos pueden ser actualizados para soportar nuevos dispositivos.
  - El receptáculo tipo A se usa para cada puerto.
  - Cada puerto debe poder entregar un mínimo de 100mA para un dispositivo configurado o sin configurar, y opcionalmente, hasta 500mA para un dispositivo configurado.
  - Los protocolos de velocidad completa y baja velocidad deben ser soportados.

- Host encajado (Embedded Host): El modo Host encajado tiene las siguientes características:

  - Solamente soporta una lista específica de dispositivos, designados en una lista (Targeted Peripheral List, TPL).

- Solamente se requiere soportar esos tipos de transferencia que son requeridos por los dispositivos en la TPL.
- El soporte USB hub es opcional.
- No se requiere los drivers de los dispositivos para ser actualizados.
- El receptáculo tipo A es usado por cada puerto.
- Solamente esas velocidades requeridas por los dispositivos en la TPL deben ser soportadas.
- Cada puerto debe poder entregar un mínimo de 100mA para un dispositivo configurado o sin confirmar y opcionalmente hasta 500mA para un dispositivo configurado.

4.4.1.2.-MODO DISPOSITIVO (Device Mode)

Los dispositivos USB aceptan comandos y datos desde el host y responde a la petición de datos. Los dispositivos USB realizan funciones periféricas.

Las siguientes características describen generalmente un dispositivo USB:

- La funcionalidad debe ser clase-específica (class-specific) o vendedor-específico (vendor-specific).
- Consigue 100mA o menos desde el bus antes de la configuración.
- Puede conseguir hasta 500mA desde el bus después de una negociación satisfactoria con el host.
- Puede soportar protocolo de baja-velocidad, velocidad-completa o alta velocidad.
- Soporta transferencia de control y datos que se requieren para la implementación.
- Soporta opcionalmente el protocolo de petición de la sesión (Session Request Protocol, SRP).

- Puede ser accionado por bus (bus-powered) o por uno mismo (self-powered).

4.4.1.3.-MODO OTG DUAL ROLE

Un dispositivo OTG Dual Role soporta host y dispositivo USB funcionalmente. Los dispositivos OTG Dual Role usan un receptáculo micro-AB. Esto permite a un enchufe micro-A o un micro-B ser conectado. Ambos enchufes, micro-A y micro-B tienen un pin adicional, el pin ID, que indica el tipo de enchufe que fue conectado. El tipo de enchufe conectado al receptáculo determina el papel por defecto del host o dispositivo. Un dispositivo OTG realizará el papel de un host cuando un enchufe micro-A es detectado. Cuando un dispositivo OTG es conectado directamente a otro dispositivo OTG usando un cable OTG (micro-A a micro-B), puede usarse el protocolo de negociación de host (Host Negotiation Protocol, HNP) para intercambiar los papeles de host y dispositivo entre los dos sin desconectar y reconectar el cable. Para diferenciar entre los dos dispositivos OTG, el término "dispositivo-A" se refiere al dispositivo conectado al enchufe micro-A y el "dispositivo-B" se refiere al dispositivo conectado al enchufe micro-B.

- <u>Dispositivo-A, el Host por defecto</u>

  En OTG Dual Role, operando como un host, las siguientes características y requerimientos describen un dispositivo-A:

  - Soporta los dispositivos de la TPL.
  - Requiere soportar esos tipos de transacciones que requieren los dispositivos de la TPL.
  - El soporte hub es opcional.
  - No se requieren los drivers de los dispositivos para ser actualizados.
  - Se usa un simple receptáculo micro-AB.
  - Debe ser soportado el protocolo de velocidad completa (full-speed)
  - El puerto USB debe poder suministrar un mínimo de 8mA para un dispositivo configurado o sin configurar y

opcionalmente hasta 500mA para un dispositivo configurado.
- Soporta HNP; el host puede intercambiar los papeles para llegar a ser un dispositivo.
- Soporta por lo menos una forma de SRP.
- El dispositivo-A recibe alimentación de VBUS cuando se alimenta el bus, aunque los papeles se intercambien usando HNP.

- Dispositivo-B, el dispositivo por defecto

  En el modo OTG Dual Role, operando como dispositivos USB, las siguientes características y requerimientos describen un dispositivo-B:

  - Funcionalidad específica clase o vendedor.
  - Consigue 8mA o menos antes de la configuración.
  - Está autoalimentado, debido a los bajos requerimientos de corriente, pero puede conseguir hasta 500mA después de una exitosa negociación con el host.
  - Se usa un simple receptáculo micro-AB.
  - Debe soportar el protocolo de velocidad completa (full-speed).
  - Soporta transferencia de control y soporta transferencias de datos que se requieren para la implementación.
  - Soportan ambas formas de SRP.
  - Soporta HNP.
  - El dispositivo-B no recibe alimentación de VBUS, aunque los papeles se intercambian usando HNP.

4.4.2.-PROTOCOLO

La comunicación USB requiere el uso de protocolos específicos. Las siguientes secciones proporcionan una descripción de la comunicación vía USB.

4.4.2.1.-BUS DE TRANSFERENCIAS

La comunicación en el bus ocurre a través de transferencias entre un host y un dispositivo. Cada tipo de transferencia tiene características únicas. Un host encajado o un host OTG pueden implementar solo el control y la transferencia de datos que se usarán.

Los siguientes cuatro tipos de transferencias son posibles en el bus:

- Transferencia de Control: La transferencia de control se usa para identificar un dispositivo durante la enumeración y para controlarlo durante la operación. Un porcentaje del ancho de banda del USB se asegura para estar disponible para las transferencias de control. Los datos son verificados por un CRC y la recepción por la tarjeta es verificada.

- Transferencia de Interrupción: La interrupción de transferencia es una transferencia programada de datos en la que el host asigna ranuras de tiempo para las transferencias que se requieren por la configuración del dispositivo. Esta asignación de ranura de tiempo se obtiene en el dispositivo siendo votado de una forma periódica. El dato es verificado por un CRC y la recepción por la tarjeta es reconocida.

- Transferencia Isócrona: La transferencia isócrona es una transferencia programada de datos en la que el host asigna ranuras de tiempo para las transacciones que se requieren para la configuración de dispositivos. La recepción no es reconocida, pero la integridad de los datos es verificada por el dispositivo usando un CRC. Este tipo de transferencia es típicamente usado para audio y video.

- Transferencia Bulk: La transferencia bulk es usada para mover gran cantidad de datos donde el tiempo de transacción no está asegurado. El tiempo para este tipo de transferencia está

asignado desde el tiempo que no ha sido asignado, hasta los otros tres tipos de transferencias. El dato es verificado por un CRC y la recepción es reconocida.

4.4.2.2.-ASIGNACIÓN DE ANCHO DE BANDA

La transferencia de control o transacciones, están garantizadas para estar por lo menos en un 10% del ancho de banda disponible dentro de un marco dado. El resto está disponible para la asignación a transferencias de interrupción y transferencias isócronas. Las transferencias bulk están asignadas para cualquier ancho de banda no asignado a transferencias de interrupción, de control o isócrona. Las transferencias bulk no tienen asegurado el ancho de banda.

4.4.2.3.-PUNTOS FINALES Y DESCRIPTORES USB

Todo dato transferido en el bus es enviado o recibido a través de puntos finales (endpoints). USB soporta dispositivos de hasta 16 puntos finales. Cada punto final puede tener funcionalidad para transmitir (Tx) y/o recibir (Rx). Cada punto final usa un tipo de transacción. El punto final ‘0’ es el punto final por defecto de la transferencia de control.

4.5.- SISTEMA CAN DEL PIC32MX575F512H

4.5.1.-CARACTERÍSTICAS BÁSICAS

Se trata de una interfaz serie, utilizada para la comunicación con otros módulos CAN u otros dispositivos del propio microcontrolador.

El modulo está formado por un ´motor de protocolo´ (protocol engine) y por un control y almacenamiento de los mensajes. El motor de protocolo CAN maneja todas las funciones encargadas de recibir y transmitir mensajes a través del bus. El estado y los errores de la transmisión pueden consultarse leyendo en los registros apropiados. Además, como ya sabemos, cualquier mensaje detectado en el bus es sometido a un control de errores y después es filtrado para comprobar si

debería ser recibido y almacenado en uno de los registros de recepción, o debería por lo contrario ser rechazado.

Básicamente las características técnicas del módulo son las siguientes:

- Conformidad estándar
    - Conformidad con el protocolo FULL CAN 2.0B
    - Velocidad de transferencia de datos programable hasta 1Mbit/segundo.
- Recepción y transmisión de mensajes
    - 32 mensajes FIFO
    - Cada FIFO puede tener hasta 32 mensajes para un total de 1024 mensajes.
    - FIFO puede ser un mensaje de transmisión FIFO o un mensaje de recepción FIFO.
    - Los niveles de prioridad definidos por el usuario para mensajes FIFO se usa para la transmisión.
    - 32 filtros de aceptación para la filtración del mensaje.
    - 4 registros de máscara de filtro de aceptación para la filtración del mensaje.
    - Respuesta automática a la Petición de Transmisión Remota.
- Características adicionales
    - Modos de auto prueba Loopback, de escucha de todos los mensajes y de solo escucha, sistemas de diagnósticos y bus de supervisión.
    - Modos de operación de baja potencia.
    - El módulo CAN es un bus maestro en el sistema bus en el PIC32MX.
    - No se requiere el uso de DMA.
    - Dispone de un controlador de tiempo dedicado.
    - Modo exclusivo para la recepción de mensajes de datos.

4.5.2.-MODOS DE FUNCIONAMIENTO

El módulo puede trabajar en uno de entre varios modos de funcionamiento, previamente escogido por el usuario entre:

- Modo de configuración (Configuration mode).
- Modo de funcionamiento normal (Normal Operation mode).
- Modo de solo escucha (Listen Only mode).
- Modo de escucha de todos los mensajes (Listen All Messages mode).
- Modo en bucle (Loopback mode).
- Modo deshabilitado (Disable mode).

Los modos de operaciones se solicitan poniendo los bits del registro (REQOP<2:0>) en el registro de control CAN (CiCON<26:24). El módulo CAN reconoce la entrada en el modo solicitado por los bits (CiCON<23:21>) del modo de operación (OPMOD<2:0>). El modo de transición se realiza con la red CAN.

La aplicación puede seleccionar ser interrumpida cuando ha ocurrido una solicitud de cambio de modo habilitando el bit de modo de cambio de interrupción (MODIE) en el registro de interrupción del CAN (CiINT<19>). Cuando el nuevo modo ha sido activado correctamente se generará una interrupción CAN. Alternativamente el usuario puede seleccionar los bits OPMOD<2.0> para determinar cuando el módulo CAN ha cambiado con éxito los modos.

4.5.2.1.-MODO DE CONFIGURACIÓN (CONFIGURATION MODE)

Después de un reajuste del dispositivo, el módulo CAN está en modo configuración (OPMOD<2:0> (CiCON<23:21>)=100). Los contadores de error están despejados y todos los registros contienen valores iniciales. El módulo CAN puede ser configurado o inicializado solamente en modo configuración. El modo configuración se solicita programando los bits del registro REQOP<2:0> a ‘100’. La aplicación debería esperar hasta que el módulo CAN esté en modo configuración. Está hecho para poner los bits de OPMOD<2:0> para un valor de ‘100’. Los siguientes registros y bits pueden ser modificados en el modo configuración solamente:

- Registro de configuración CAN (CiCFG).
- Registro CAN FIFO de base de dirección.

- Registro CAN de máscara de filtro de aceptación.
- Bits de tamaño FIFO (FSIZE) y el bit DONLY en el registro de control CAN FIFO (CiFIFOCONn).

Esto protege al usuario de violar accidentalmente el protocolo CAN a través de la programación de errores.

4.5.2.2.-MODO DE FUNCIONAMIENTO NORMAL (NORMAL OPERATION MODE)

En modo funcionamiento normal, el módulo CAN estará en el bus CAN y puede transmitir y recibir mensajes CAN. El modo de funcionamiento normal se solicita después de la inicialización programando los bits REQOP<2:0> a '000'. Cuando OPMOD<2:0>=000 el modulo trabaja con funcionamiento normal.

4.5.2.3.-MODO DE SOLO ESCUCHA (LISTEN ONLY MODE)

El modo de solo escucha es una variante del modo de funcionamiento normal. Si el modo de solo escucha está activado, el módulo que está presente en el bus CAN está pasivo. Recibirá mensajes pero no transmitirá. El módulo CAN no generará banderas de error ni reconocerá señales. Los contadores de error están desactivados en este estado. El modo de solo escucha puede ser usado para detectar la tasa (velocidad de transmisión) en baudios del bus CAN. Parece evidente que para poder utilizarlo, es necesario que haya al menos dos nodos remotos que se comuniquen el uno con el otro.

4.5.2.4.-MODO DE ESCUCHA DE TODOS LOS MENSAJES (LISTEN ALL MESSAGES MODE)

El modo de escucha de todos los mensajes es una variante del modo de funcionamiento normal. Si el modo de escucha de todos los mensajes está activado, la transmisión y recepción se realiza igual que el modo de funcionamiento normal con la excepción de que si un mensaje se recibe con un error, será transferido al buffer receptor como si no hubiese error. El buffer receptor contendrá el dato que recibió hasta el error. Los filtros todavía necesitan ser habilitados y configurados

4.5.2.5.-MODO EN BUCLE (LOOPBACK MODE)

Cuando el modo loopback se activa, la señal interna de transmisión y la señal interna de recepción se conectan en el límite del módulo, formando una especie de bucle cerrados.

4.5.2.6.-MODO DESHABILITADO (DISABLE MODE)

El modo deshabilitado es similar al modo configuración, excepto que los contadores de error del módulo CAN no están inicializados. El módulo está posicionado en modo deshabilitación poniendo los bits del REQOP<2:0> a '001'. En modo deshabilitación, el reloj interno del módulo CAN parará hasta que el módulo esté recibiendo o transmitiendo un mensaje. El módulo CAN no permitirá entrar en modo deshabilitación mientras esté teniendo lugar una transmisión o recepción para prevenir que se produzcan errores en el sistema bus. El módulo entrará en modo deshabilitación cuando el mensaje actual termina. Los bits OPMOD<2:0> (CiCON<23:21>) indican si el módulo entró en modo deshabilitación. El pin del módulo de transmisión CAN (CiTx) permanece en estado recesivo mientras el módulo está en modo deshabilitación para prevenir los errores en el bus CAN.

4.5.3.-TRANSMISIÓN DE MENSAJES

4.5.3.1.-BUFFERS DE TRANSMISIÓN

La FIFO de trasmisión de mensajes puede contener hasta 32 buffers de transmisión de mensajes CAN. Un buffer de transmisión de mensajes son 4 palabras (16bytes). Los bits en estos registros tienen correspondencia uno a uno con los campos del bit en el mensaje CAN definido por el bus CAN.

4.5.3.2.-PETICIÓN DE TRANSMISIÓN DEL MENSAJE

La aplicación debe configurar la FIFO para la transmisión. El mensaje de transmisión y la prioridad están controlados por los bits del registro CiFIFOCONn (Registro 34-20). El módulo CAN seleccionará el siguiente mensaje para ser transmitido basándose en la prioridad y procesará este mensaje para su transmisión.

Una vez que un mensaje ha sido cargado en la FIFO y está listo para ser transmitido, la aplicación del usuario puede iniciar una transmisión ajustando el bit de petición de envío de mensaje (TXREQ) en el registro de control CAN FIFO(CiFIFOCONn<3>). Cuando este bit está ajustado, los contenidos de la FIFO harán cola para la transmisión de acuerdo con la prioridad del mensaje, y el módulo CAN transmitirá todos los mensajes en la FIFO. Cuando todos los mensajes han sido transmitidos, el bit TXREQ será inicializado. Una aplicación podría cargar todas las FIFO de transmisión y ajustar el bit TXREQ para todas estas FIFO.

Alternativamente, la aplicación podría cargar una FIFO de transmisión, ajustar el bit TXREQ y luego cargar la siguiente FIFO mientras la FIFO previa está siendo procesada.

Los mensajes pueden ser añadidos a la FIFO mientras un mensaje está siendo transmitido. La aplicación debería usar el registro CIFICOUAn para obtener el puntero de cabeza de la FIFO y debería almacenar el mensaje en esta dirección.

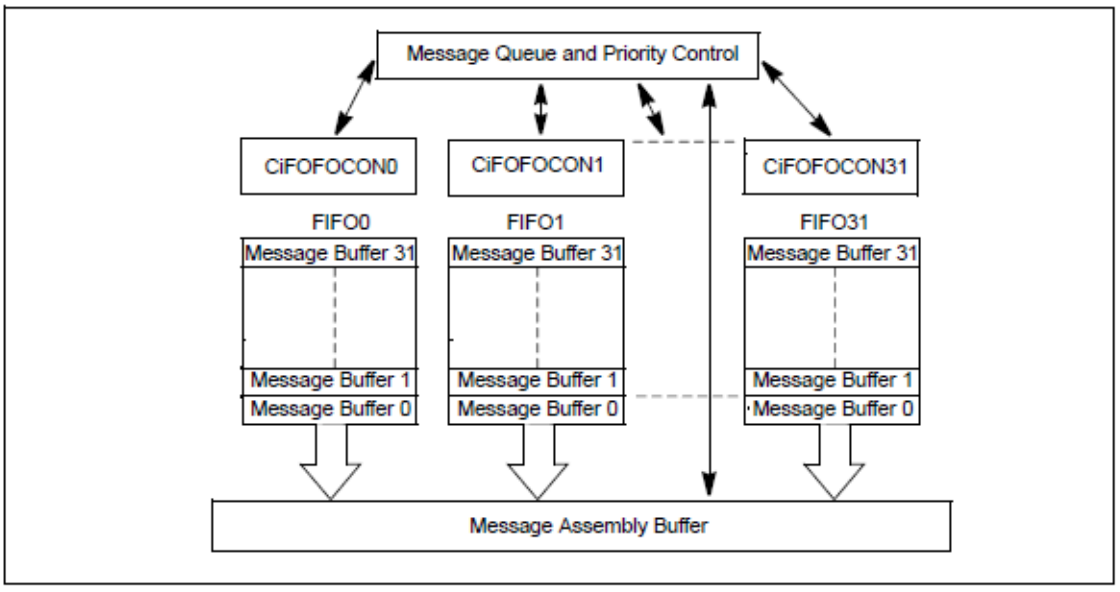


Figura 33 – Cap.4.5.3.2: Mensaje de transmisión CAN

4.5.3.3.-PRIORIDAD DE TRANSMISIÓN DEL MENSAJE

El módulo CAN permite a la aplicación controlar la prioridad de los buffers de transmisión de mensajes. Anteriormente al enviar un mensaje SOF, el módulo CAN compara la prioridad de todos los buffers que están listos para transmitir. El buffer de transmisión de mensajes con la prioridad más alta será el primero en enviar.

Los niveles de prioridad están controlados por los bits TXPRI<1:0> (CiFIFOCONn<1:0>). Hay 4 niveles de prioridad de transmisión definidos por los bits TXPRI. Si TXPRI<1:0> para un buffer de mensaje en particular se establece a '11' ese buffer tiene la máxima prioridad. Si TXPRI<1:0> par un buffer de mensaje en particular se establece a '10' o a '01', ese buffer tiene una prioridad intermedia. Si TXPRI<1:0> para un buffer de mensaje en particular se establece a '00', ese buffer tiene la mínima prioridad.

4.5.3.4.-ABORTANDO LA TRANSMISIÓN DE UN MENSAJE EN COLA

Un mensaje que ha sido puesto en cola para ser transmitido puede ser abortado limpiando el bit TXREQ (CiFIFOCONn<3>). Si se limpia TXREQ, el módulo procurará abortar la transmisión.

Si el mensaje se aborta satisfactoriamente, el bit TXBAT (CiFIFOCONn<2>) se ajustará por hardware. TXREQ permanecerá ajustado hasta que el mensaje aborte o sea transmitido satisfactoriamente.
Si el mensaje se transmite satisfactoriamente, los punteros de FIFO serán actualizados de forma normal. Si el mensaje es satisfactoriamente abortado, los punteros FIFO no cambiarán. El usuario puede entonces usar el índice interno (CFIFOCI) para determinar que mensajes han sido transmitidos si se requiere.

Al inicializar los punteros FIFO y borrar todos los mensajes pendientes, el usuario puede ajustar el bit FRESET (CiFIFOCONn<14>). La FIFO puede entonces ser habilitada de nuevo y cargada con nuevos mensajes para ser transmitidos.

4.5.4.-RECEPCIÓN DE MENSAJES

El módulo CAN supervisa continuamente mensajes en el bus CAN. Cuando el módulo CAN recibe los mensajes, se compara el identificador del mensaje con las combinaciones del filtro que están actualmente configurados.

Si se produce una unión, el módulo almacenará el mensaje en la FIFO apuntada por FSELn. Una vez que el mensaje ha sido recibido y almacenado en la FIFO, los siguientes bits serán actualizados.

- Los bits CFIFOCI<4:0> en el registro CiFIFOCIn será actualizado.
- RXOUFLIF, RXFULLIF, RXNEMPTYIF, RXHALFIF en el registro CiFIFOINTn será actualizado apropiadamente.
- La bandera de interrupción pendiente en la FIFO (FIFOIPn) en el registro CiFSTAT será actualizada apropiadamente.
- Una interrupción será generada si se permite.

Los bits del código de interrupción del módulo CAN (ICOD<6:0>) en el registro (CiVEC<6:0>) y los bits del filtro (FILHIT<4:0>) (CiVEC<12:8>) en el registro del código de interrupción CAN (CiVEC) serán actualizados también.

4.5.4.1.-RECEPCIÓN DE MENSAJES SOLAMENTE DE DATOS

El módulo CAN del PIC32 proporciona un modo especial de recepción donde solamente la parte útil de los datos del mensaje recibido es almacenado en el mensaje del buffer. El módulo descartará el identificador, bits reservados y bits DLC. El valor del tiempo estampa (fecha/hora) no es almacenado si este modo está deshabilitado. Este modo puede ser habilitado seleccionando el bit (DONLY) del (CiFIFOCONn<12>).
8 bytes de datos son almacenados, sea cual sea el valor del campo del DLC en el mensaje CAN. Los bytes que no se usan son cargados con el valor 0x00. Un posible uso de este modo es concatenar mensajes cuyos datos atraviesan mensajes múltiples.

4.5.4.2.-PROCESAMIENTO DE UN MENSAJE RECIBIDO

La aplicación usuario puede utilizar métodos de interrogación o usar el módulo de interrupción CAN para la recepción del mensaje.

El módulo CAN proporciona notificación sobre los diferentes eventos de la FIFO de recepción. La aplicación puede ser notificada cuando la FIFO de recepción no está vacía, está medio llena o está llena. Las aplicaciones leerían la FIFO de recepción frecuentemente para asegurarse que allí no hay desbordamiento de FIFO.
Cueste lo que cueste la técnica usada, los siguientes pasos serán usados para leer el mensaje desde la FIFO.

1- Leer el valor del registro CiFIFOUAn. Este provee 32 bits físicos de puntero de dirección al buffer de recepción de mensaje que la aplicación debe leer.
2- En el caso del tipo de recepción de mensajes de datos solamente, se procesan 2 palabras desde el buffer de mensaje.
3- En el caso del tipo de recepción de mensajes enteros, procesan 4 palabras desde el buffer de mensaje.
4- Después de procesar, el buffer del mensaje, selecciona el bit UINC en el registro asociado CiFIFOCONn.

4.5.5.-TEMPORIZACIÓN DE BITS

La tasa nominal de bit es el número de bits transmitidos en el bus CAN.

Tiempo nominal de bit = 1 + tasa nominal de bit.

Hay 4 segmentos de tiempo en un bit de tiempo para compensar cualquier cambio de fase debido a derivaciones o retrasos de propagación en el oscilador. Estos segmentos de tiempo no se solapan entre ellos y están representados en términos de quantum de tiempo (TQ). Una TQ es una unidad de tiempo derivado fijado por el reloj del oscilador. El número total de quantum de tiempo en un bit debe ser programado entre 8 y 25 TQ.

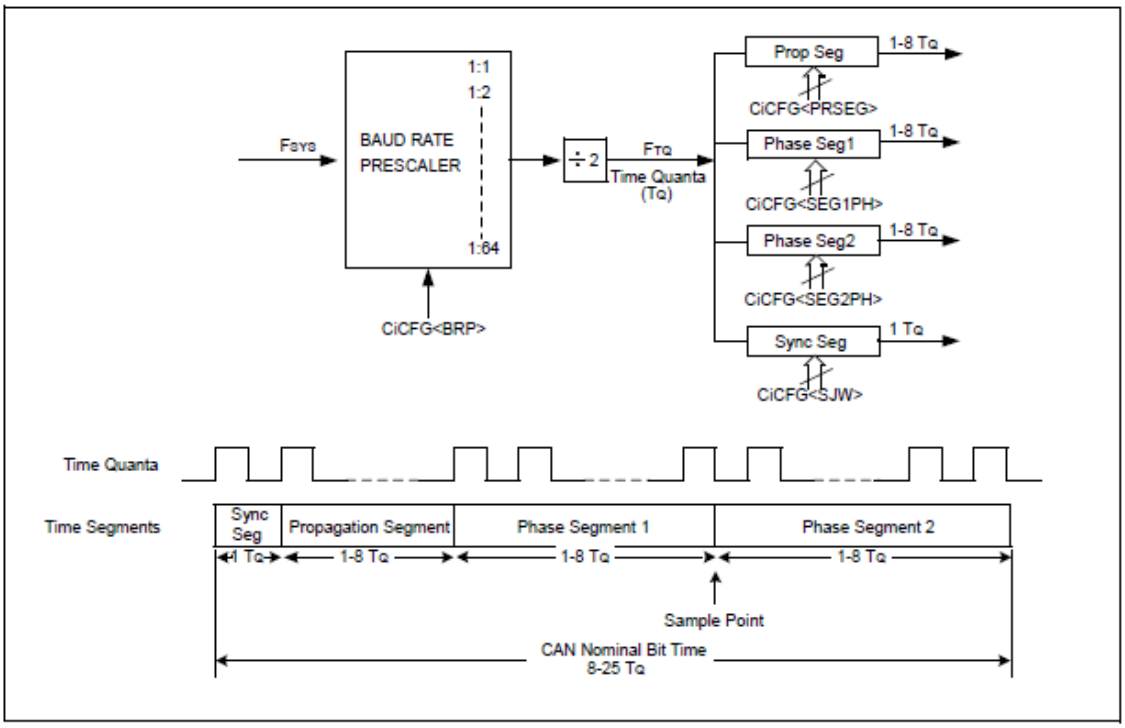


Figura 34 – Cap.4.5.5: Temporización de bits CAN

Segmentos de bit

Cada tiempo de bit de transmisión consta de 4 segmentos de tiempo:

- Segmento de sincronización (Synchronization Segment): este segmento de tiempo sincroniza los diferentes nodos conectados en el bus CAN. Se espera un bit borde dentro de este segmento. Basado en el protocolo CAN, el segmento de sincronización es asumido para ser un Quantum de Tiempo.

- Segmento de propagación (Propagation Segment): este segmento de tiempo compensa cualquier retraso de tiempo que pueden ocurrir debido a la línea del bus o debido a los diversos transceptores conectados en ese bus.

- Segmento de fase 1 (Phase Segment 1): este segmento de tiempo compensa los errores que pueden ocurrir debido a cambios de fase en los bordes. Este segmento de tiempo puede ser alargado durante la resincronización para compensar el cambio de fase.

- Segmento de fase 2 (Phase Segment 2): este segmento de tiempo compensa los errores que pueden ocurrir debido a cambios de fase en los bordes. El segmento de tiempo puede ser acortado durante la resincronización para compensar el cambio de fase.

Punto de muestra

El punto de muestra es el punto en un intervalo de tiempo en un bit CAN donde se toma la muestra y el estado del bus es leído en interpretado. Está situado entre el segmento de fase 1 y el segmento de fase 2. El bus CAN puede ser muestreado una vez o tres veces en el punto de muestra, configurado por el bit de la muestra de línea del bus CAN (SAM) en el registro de configuración de velocidad CAN (CiCFG<14>).

- Si CiCFG<SAM>=1, el bus CAN es muestreado tres veces en el punto de muestra. El valor más común de las 3 muestras determina el valor del bit.
- Si CiCFG<SAM>=0, el bus CAN es muestreado una sola vez en el punto de muestra.

Sincronización

Se usan dos tipos de sincronización: sincronización de trama o resincronización. Una sincronización de trama ocurre una vez en el comienzo de una trama. La resincronización ocurre dentro de una trama.

- Sincronización de trama: tiene lugar en la transición de recesivo a dominante del bit de comienzo. El tiempo de bit comienza desde ese borde.

- Resincronización: tiene lugar cuando un bit de borde no ocurre dentro del segmento de sincronización en un mensaje. Uno de los segmentos de fase es acortado o alargado una cantidad que depende del error de fase en la señal. La cantidad máxima que puede ser utilizada es determinada por el parámetro (CiCFG<SJW>).

La longitud del segmento de fase 1 y el segmento de fase 2 puede ser cambiada dependiendo de las tolerancias del oscilador en la trasmisión y recepción del nodo. La resincronización compensa cualquier cambio de

fase que puede ocurrir debido a los diferentes osciladores usados para la transmisión y recepción de nodos.

5.- DISEÑO DEL INTERFACE

5.1.- INTRODUCCIÓN

En este capítulo se va a detallar el diseño de nuestro prototipo de interface USB-CAN, así como se añadirá una completa explicación de los rutados realizados y la disposición de los componentes en nuestra placa.

Para comenzar, el diseño eléctrico de nuestra tarjeta se ha realizado mediante Microsim Schematics ayudándonos de las herramientas de las que dicho software dispone. Para la obtención del rutado de la placa se ha utilizado el programa MicroSim PCBBOARD.

A continuación se detallan cómo se ha realizado el diseño eléctrico y el rutado de la placa.

5.2.- DISEÑO ELÉCTRICO

Como ya se ha dicho anteriormente, para el diseño eléctrico hemos usado la herramienta Microsim Schematics, mediante la cual hemos realizado las conexiones eléctricas oportunas.

MICROPROCESADOR (PIC32MS575F512H)

Para comenzar a establecer esas conexiones eléctricas partimos de la necesidad de unas conexiones mínimas que necesita el PIC elegido, en este caso el PIC32MX575F512H. Este microcontrolador puede desempeñar ambas funciones, tanto CAN como USB, por eso nos decantamos por su elección y porque se trata de una nueva generación de procesadores que van a emplearse con mucha frecuencia en la industria actual.

Según podemos ver en el datasheet del microcontrolador (adjuntado en el CD) existen una serie de conexiones necesarias para mantener al mismo en unas condiciones óptimas de temperatura y funcionamiento y que nos permiten trabajar de manera segura y eficiente.

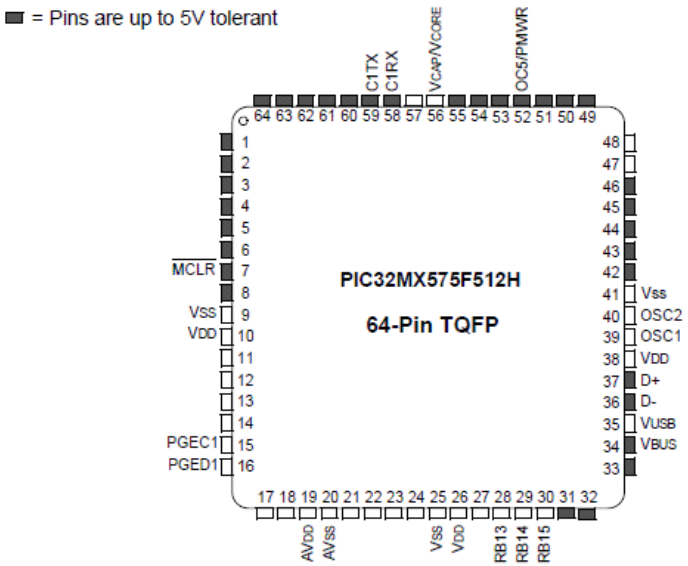


Figura 35 – Cap.5.2 : Diagrama de pines usados del PIC

Como sabemos, todo microcontrolador requiere de un circuito externo que le indique la velocidad a la que debe trabajar. Este circuito, se conoce con el nombre de oscilador o reloj, es muy simple pero de vital importancia para el buen funcionamiento del sistema. Existen varios tipos de osciladores: oscilador con resistencia y condensador (RC), Cristal de cuarzo (XT), etc.

Nuestro micro tiene osciladores internos de 8MHz y 32KHz, pero hemos optado por dotarlo externamente de un oscilador de cristal de 12MHz (el rango de frecuencias que nuestro PIC nos permite utilizar como oscilador es de 3 – 25MHz) porque consideramos que así nuestro PIC trabajaría de manera más eficiente. La elección de un oscilador de cristal y no de otro tipo viene determinada porque los osciladores de cristal ofrecen mayor precisión y estabilidad que el resto.

En dicho interface, también se precisa de un Circuito Reset para su correcto funcionamiento, según vemos en el plano 1 del esquema eléctrico.

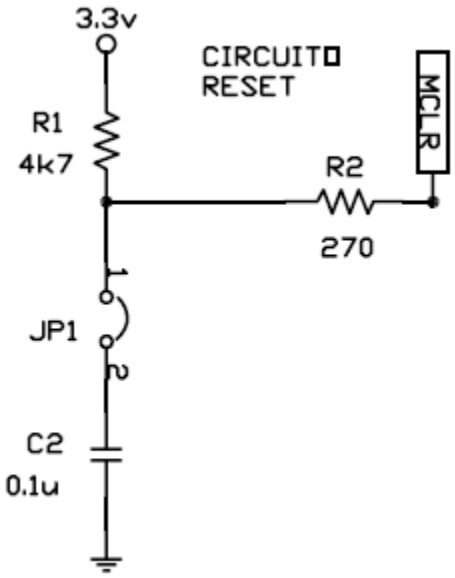


Figura 36 – Cap.5.2: Esquema del circuito Reset

CONECTORES

También hemos incorporado un conector USB para conectar el interface al ordenador y poder tratar la información recibida así como gestionar el envío de tramas CAN. A su vez se ha ubicado un conector DB-9 en el otro extremo para hacer las veces de conector CAN y poder recibir/enviar datos al bus CAN que estamos gestionando.
En la siguiente figura podemos comprobar el diagrama de pines de un conector USB tipo A como el empleado en nuestro prototipo.

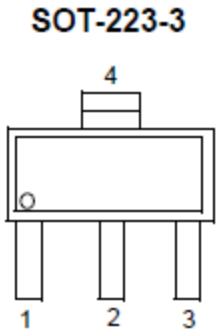
Pin	Signal	Description
1	VCC	+5V
2	D-	Data -
3	D+	Data +
4	GND	Ground

Figura 37 – Cap.5.2: Diagrama de pines del conector USB

Para programar el PIC sin necesidad de retirarlo (o desoldarlo de la PCB) se ha decidido introducir un conector ICSP (*InCircuit Serial Programming*) que permite tener acceso a las patillas de programación del microcontrolador (PGED1 y PGEC1).
Programar los microcontroladores sin tener que desmontarlos de los circuitos donde están ubicados es una particularidad que tienen casi todos los dispositivos, pero si, además, su memoria de programa es del tipo Flash les confiere aún mayor flexibilidad, pues se pueden realizar cambios en los programas y volver a reprogramar el microcontrolador tantas veces como sea necesario para depurar la aplicación. A la hora de llevar a cabo este tipo de programación/depuración del microcontrolador se usará la herramienta PICKit 3. En la figura 42 se puede ver la manera de conectar el PICKit 3 al interface para realizar la programación.

REGULADOR DE TENSIÓN

Se ha incorporado un regulador de tensión que transforma los 5voltios que suministra el puerto USB2.0 de un PC para alimentación externa, en los 3.3voltios que es la tensión de alimentación que necesita nuestro PIC para trabajar de forma adecuada y estable.

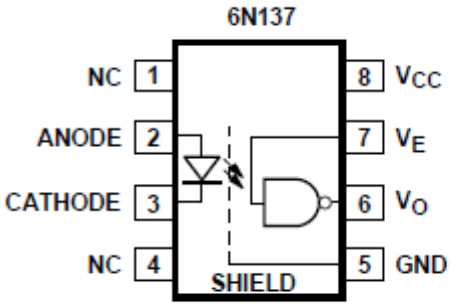


Pin	
1	V_{IN}
2	GND (TAB)
3	V_{OUT}
4	GND (TAB)

Figura 38 – Cap.5.2: Diagrama de pines del regulador de tensión

OPTOACOPLADORES

Dado que la tarjeta puede ser utilizada para transferir información desde vehículos que usan 12V (coches) y también desde vehículos que usan 24V tales como camiones, se han introducido 3 optoacopladores (6N137). Lo que van a hacer estos dispositivos es proteger nuestro PIC, que solo va a necesitar 3,3V para ser alimentado, de tensiones superiores procedentes de otros sistemas. Por tanto los optoacopladores evitan ese tipo de sobretensiones en el PIC y evitan que nuestro prototipo se estropee o dañe.
A continuación podemos observar el diagrama de pines de los optoacopladores así como la tabla de verdad de los mismos



TRUTH TABLE
(POSITIVE LOGIC)

LED	ENABLE	OUTPUT
ON	H	L
OFF	H	H
ON	L	H
OFF	L	H
ON	NC	L
OFF	NC	H

Figura 39 – Cap.5.2: Diagrama de pines y tabla de verdad de los optoacopladores

FUENTE DE ALIMENTACIÓN

Para un mejor control de las tensiones de entrada, así como para una protección contra cortocircuitos, hemos ubicado en nuestro interface un convertidor DC/DC (TMR0521 de TRACO POWER). Los convertidores DC/DC tienen varias aplicaciones en las que podemos necesitarlos, como pueden ser: proporcionar aislamiento de la fuente principal o variar una tensión en una zona del diseño. Estos convertidores además proporcionan:

- Estabilidad del voltaje de salida
- Protección contra cortocircuitos
- Limitación de la corriente de salida
- Protección contra sobretensiones
- Mayor estabilidad del voltaje de salida
- Disminución del ruido de salida

TRANSCEPTOR MCP2551

El transceptor MCP2551 que hemos incluido es un dispositivo de alta velocidad que sirve como interface entre el bus físico y los controladores en un sistema con protocolo CAN. Este dispositivo permite transmitir y recibir entre muchos nodos conectados a una red CAN. También permite, juntos con otros componentes CAN, realizar un nodo CAN completo.
Este componente tiene como principales características soportar la tasa de transferencia máxima de 1 Mbps que se especifica en el protocolo CANopen, implementar los requerimientos de la capa física ISO-11898, es capaz de detectar fallos en la tierra y permite conectar un máximo de 112 Nodos al Bus. En la siguiente figura se define el patillaje y funcionalidad del transceptor.

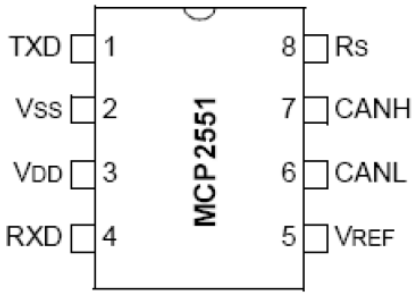


Figura 40 – Cap.5.2: Patillaje del MCP2551

En la tabla siguiente se muestra la descripción de cada pin.

Nº de pin	Nombre del pin	Función del pin
1	TxD	Entrada de transmisión de datos.
2	V_{SS}	Tensión de masa.
3	V_{DD}	Tensión de alimentación.
4	RxD	Salida de recepción de datos.
5	V_{ref}	Voltaje de referencia.
6	CANL	Nivel bajo de la señal CAN.
7	CANH	Nivel alto de la señal CAN.
8	R_s	Resistencia de control de modo.

Tabla 4 – Cap.5.2: Descripción de los pines del MCP2551

Se emplean 3 leds que permiten comprobar de manera luminosa el tipo de operación que estamos realizando con nuestra tarjeta, así como si el envío de tramas se realiza de manera correcta. Los tres leds escogidos son:

1- Rojo: cuando conectamos la tarjeta al PC, por el puerto USB del mismo, parpadea mostrándonos que está bien conectado.
2- Verde: se encenderá cuando enviemos una trama correctamente al bus CAN y cuando pulsemos la tecla 'V' en la función principal (main) como método de comprobación de que tanto el led como la tarjeta funcionan adecuadamente.
3- Amarillo: se encenderá cuando recibamos una trama correctamente desde el bus CAN y cuando pulsemos la tecla 'A' en la función principal (main) como método de comprobación de que tanto el led como la tarjeta funcionan adecuadamente.

5.3.- RUTADO Y DISPOSICIÓN DE COMPONENTES

El rutado de las pistas de nuestro prototipo de interface se ha realizado manualmente mediante la herramienta MicroSim PCBoards.
Hemos trabajado sobre una superficie de placa de 11,7 x 7,3cm aproximadamente y en dicha superficie hemos dispuesto todos y cada uno de los componentes. Decir en este apartado que la placa de nuestro prototipo está fabricada en fibra de vidrio de doble cara, ya que situamos componentes en ambas caras para facilitar la conexión de los mismos y evitar de esta manera pistas demasiado largas entre ellos y así también que el ruido en la tarjeta sea lo menor posible.

Por lo tanto, en estas dos caras (Component y Solder) hemos situado los componentes respetando que el microcontrolador estuviese situado en el centro y alrededor de él se ubican el resto, procurando tener lo más cerca posible del mismo el oscilador y los condensadores de acoplo para evitar pistas largas y ruidos innecesarios.

Ya que se trata de un prototipo para transmitir datos de USB a CAN y viceversa, ambos conectores (USB y DB9) se han situado en extremos

opuestos, ubicando cerca del DB9 el transceptor y los 3 optoacopladores que van a evitar que sobretensiones y descargas estropeen el dispositivo cuando se conecte al bus CAN de vehículos con tensiones mayores a las de un coche normal (12V).

Los leds indicativos se han situado en un sitio visible para que puedan identificarse cada una de las funciones que se realizan con la tarjeta.

El conector ICSP que nos va a permitir programar nuestro microcontrolador está situado en un lateral de la tarjeta para facilitar de esta manera la conexión con el programador PICKit3 que va a transferir el programa diseñado en nuestro PC mediante la herramienta MPLAB.

Por otra parte, decir que las pistas empleadas son de 12 y 25 milésimas de pulgada y para las pistas que pasan por encima de la superficie del microcontrolador y próximas al mismo se han empleado pistas de 8 milésimas de pulgada para evitar cortocircuitos.

Se ha diseñado un Areafill que recubre toda la superficie y que sirve de radiador para disipar la temperatura de la tarjeta y de los componentes más sensibles.

5.4.- FABRICACIÓN Y MONTAJE

Una vez realizado el diseño y rutado de la tarjeta procedemos al taladrado y fresado de las diferentes pistas sobre la placa de fibra de vidrio. Mediante la herramienta Routepro2008 permitimos a nuestra máquina fresadora poder interpretar toda la información procedente de MicroSim PCBoard, tanto la ubicación de los taladros, anchura de las pistas, dimensiones de la tarjeta y rotulaciones que se hayan inscrito.

Nuestra máquina fresadora automáticamente realizará esta operación y ya sólo tenemos que estar pendientes del montaje de los componentes.
Dicho montaje se realizará mediante soldadura con pasta de estaño procurando que no se produzca ningún tipo de cortocircuito con ninguna pista ni ningún otro componente, comenzando con el microcontrolador y tratando de transmitirle la menor temperatura posible procedente del soldador para evitar que se estropee, ya que es un componente muy sensible a la temperatura.

En la siguiente imagen podemos ver el resultado final de nuestro montaje, con todos los componentes ubicados en su lugar correspondiente en la placa.

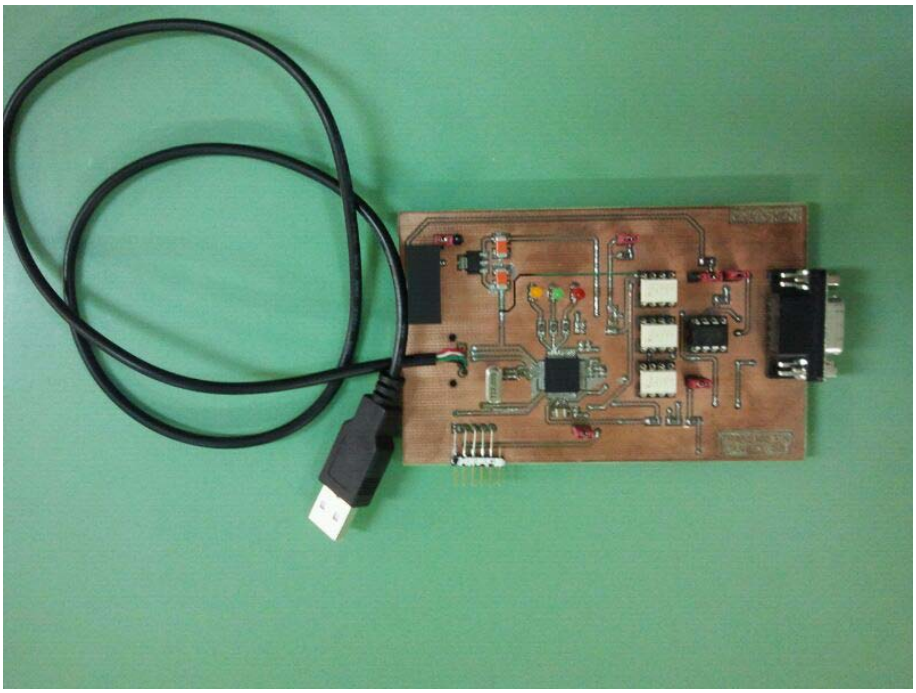


Figura 41 – Cap.5.4: Imagen de la tarjeta fabricada

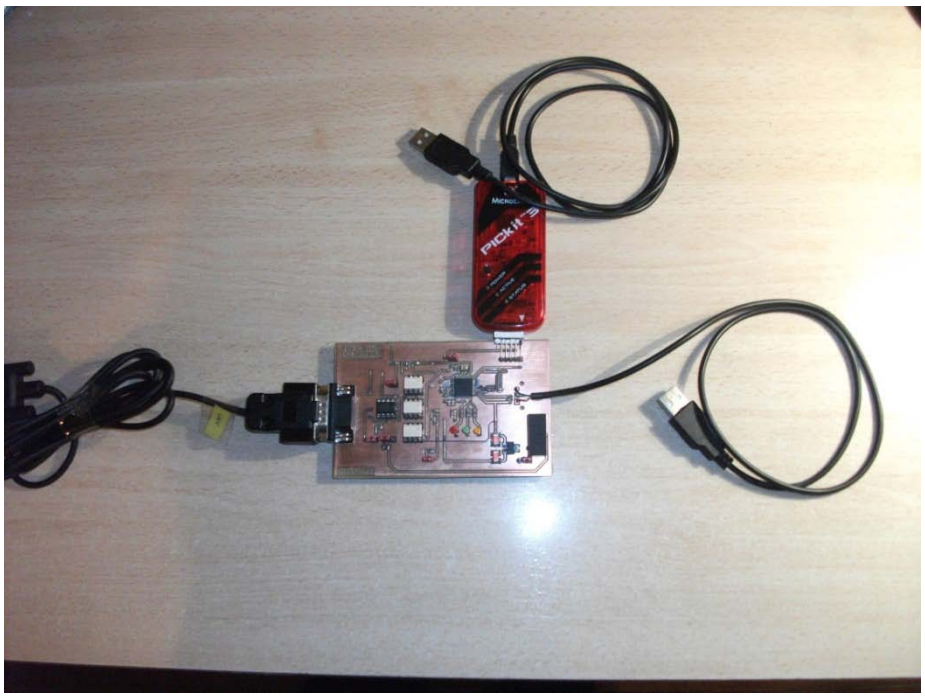


Figura42 – Cap.5.4: Conexión del PICKit3 al dispositivo

Una vez realizado el montaje de todos los componentes se realizará un pequeño test para comprobar que todo está conectado correctamente. En caso afirmativo conviene guardar la tarjeta en una bolsa antielectrostática que evite que las pequeñas descargas puedan estropear nuestra placa.

6.- FIRMWARE

6.1.- INTRODUCCIÓN

La programación del Microcontrolador PIC32MX575F512H la hemos realizado en lenguaje C, mediante el entorno de PIC C COMPILER, de C32.

Hemos escogido este lenguaje de programación por la facilidad de uso, así como por su amplio desarrollo entre los programadores del campo industrial.

Para la compilación del programa ha sido necesaria la utilización del Compilador de C para el entorno MPLAB. Este programa tiene la versatilidad de poder compilar tanto lenguaje C como lenguaje ensamblador, lo que resulta muy útil para la realización de determinada programación crítica que requiera de un plus de velocidad en algún momento, debido a su gran compactación en cuanto a espacio de programación se refiere.

El entorno MPLAB permite simular el programa creado antes de insertarlo en el Microcontrolador, de manera que siempre se pueden depurar pequeños aspectos del código, que en ensayos reales con el microcontrolador, en cambio, no se podrían observar con tanta claridad.

El código fuente ha sido incluido en los ANEXOS como CÓDIGO FUENTE.

A continuación se muestran los diagramas de flujo de las diferentes rutinas implementadas. En ellas se describe de forma explícita y, a grandes rasgos, pero de manera muy intuitiva, los procesos que sigue el microcontrolador durante la ejecución del código. En estos diagramas intentamos reflejar todas las acciones que se deben realizar para conseguir el propósito plasmado en este documento.

6.2.- DIAGRAMAS DE FLUJO

Para facilitar la comprensión del programa que controla el funcionamiento de la tarjeta que hemos diseñado, tenemos los siguientes diagramas de flujo, donde podemos ver la sencillez y robustez del software de nuestro interfaz.

A continuación podemos observar la estructura general de nuestro programa en C desde dónde se llamarán a las distintas funciones que a su vez incluirán otras llamadas a funciones que permitirán a nuestra tarjeta trabajar del modo deseado.

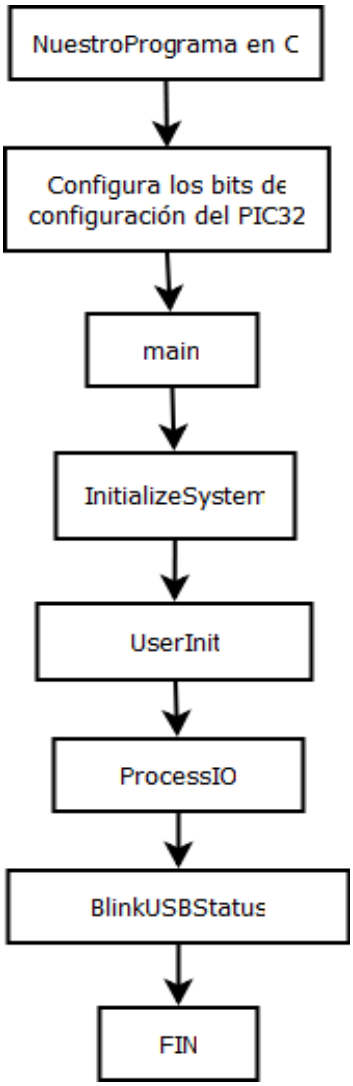


Figura 43 – Cap.6.2: Estructura general del programa

Desde esta función (NuestroPrograma en C) es desde donde se llama a las funciones que inicializan y configuran el USB. También aquí se configurará el PIC para que esté operativo

En el siguiente diagrama se representa la función CAN_Functions_mi. Al llamar a esta función dentro de nuestro programa se inicializa y configura el CAN. También se encuentran las funciones para transmitir y recibir tramas, así como se habilitarán las interrupciones del CAN para recibir tramas.

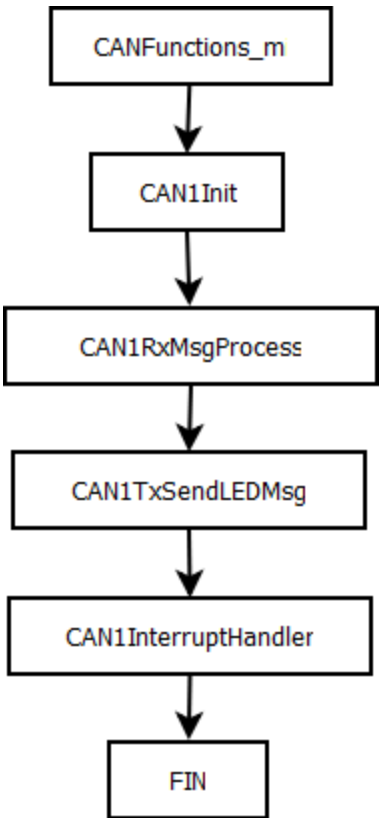


Figura 44 – Cap.6.2: Diagrama de la función CANFunctions_mi

Con la siguiente función configuraremos el sistema para tener un máximo rendimiento. Se habilitará todo el sistema de interrupciones e inicializaremos los módulos USB y CAN. Desde aquí comprobaremos si se reciben tramas o no y si el usuario ha seleccionado alguna opción en el menú pulsando la tecla correspondiente, mediante el muestreo de la función ProcessIO.

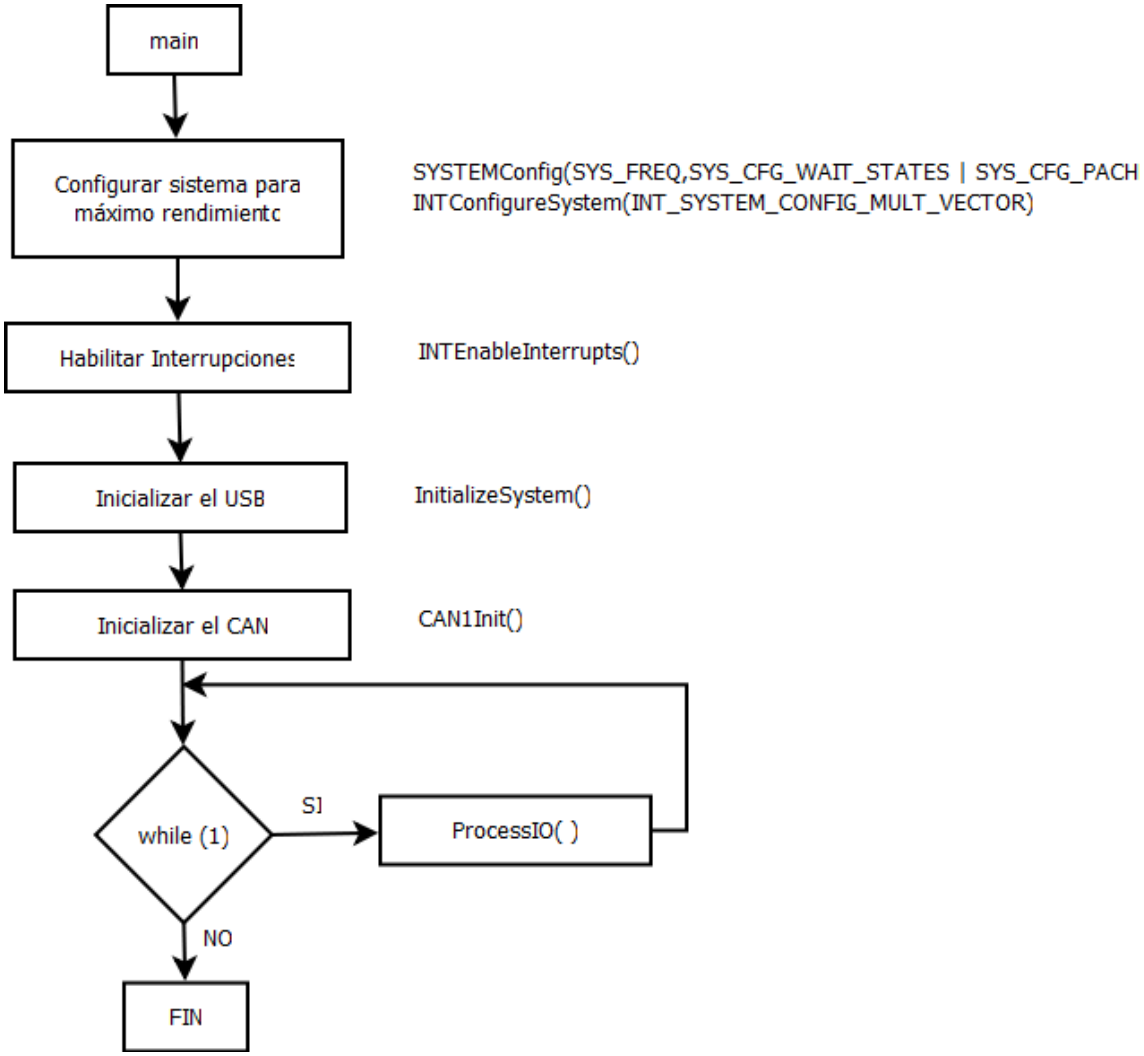


Figura 45 – Cap.6.2: Diagrama de la función main

Con la siguiente función (InitializeSystem) configuraremos el puerto B cómo entradas/salidas digitales, inhabilitaremos el puerto JTAG porque usa varios pines del puerto B y además inicializaremos el módulo SFR (registro de funciones especiales) del USB así como las variables firmware.

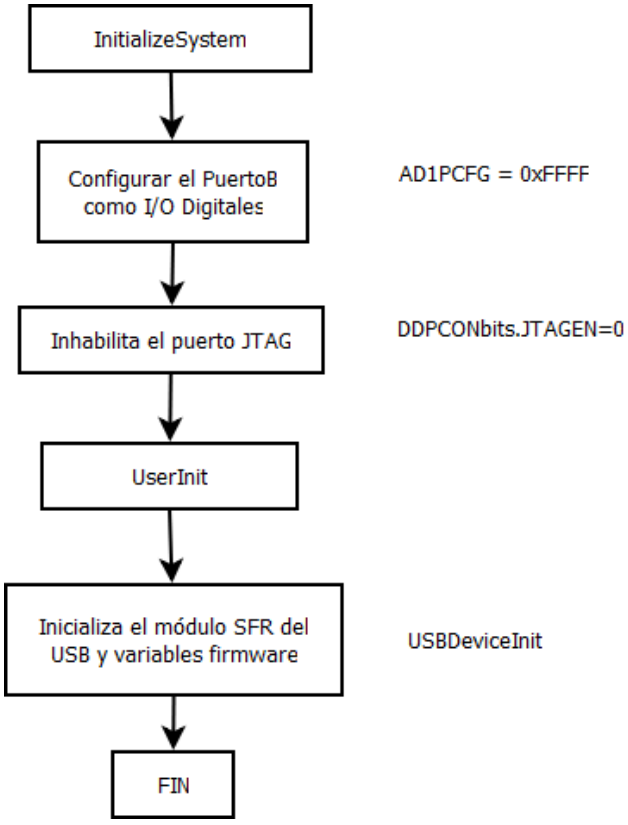


Figura 46 – Cap.6.2: Diagrama de la función InitializeSystem

En el siguiente diagrama se representa la función UserInit en la que inicializaremos todas las variables anti rebotes y los pines de los leds

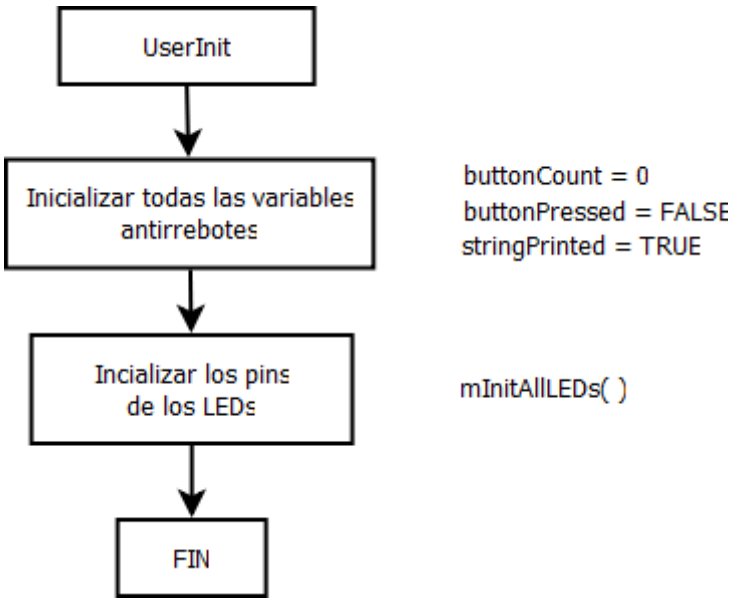


Figura 47 – Cap.6.2: Diagrama de la función UserInit

A continuación, se refleja el diagrama de ProcessIO, que es la función donde desarrollaremos el núcleo de nuestro código y donde implementaremos lo que queremos que haga nuestra tarjeta. Esta es la función que muestreamos desde el main para comprobar qué tecla se ha pulsado, para así realizar la operación deseada por el usuario. También esta función es muestreada para saber si el PIC ha recibido alguna trama y así mostrarla por pantalla para que el usuario la vea.

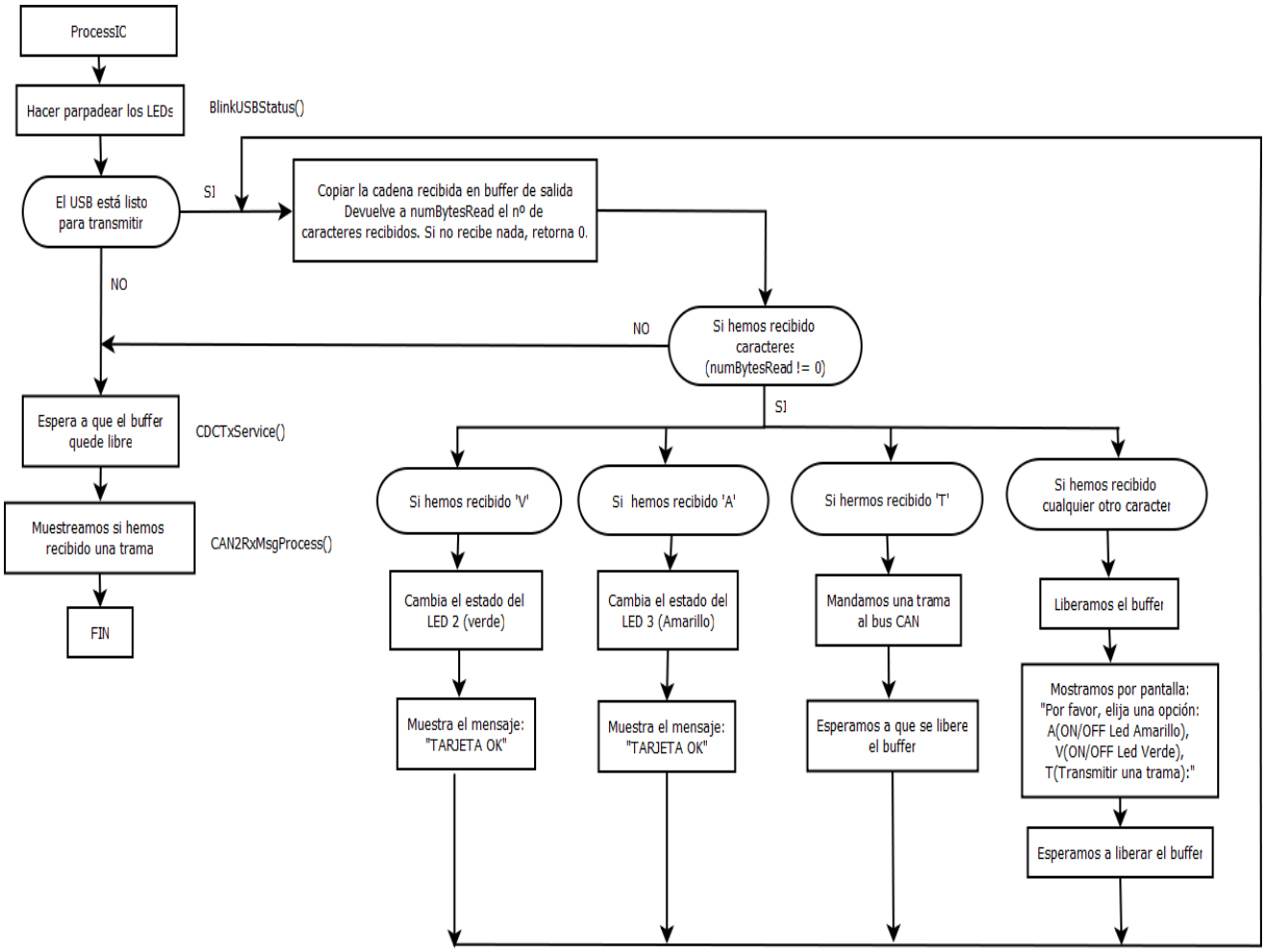


Figura 48 – Cap.6.2: Diagrama de la función ProcessIO

El siguiente diagrama representa la estructura de la función BlinkUSBStatus. Esta función es la que realiza el juego de LEDs según el estado de la tarjeta. Por ejemplo, si la tarjeta está ‘POWERED’ se encenderá el LED 1 (Rojo), mientras que si está ’CONFIGURED’ apagará exclusivamente el LED 1 (Rojo).

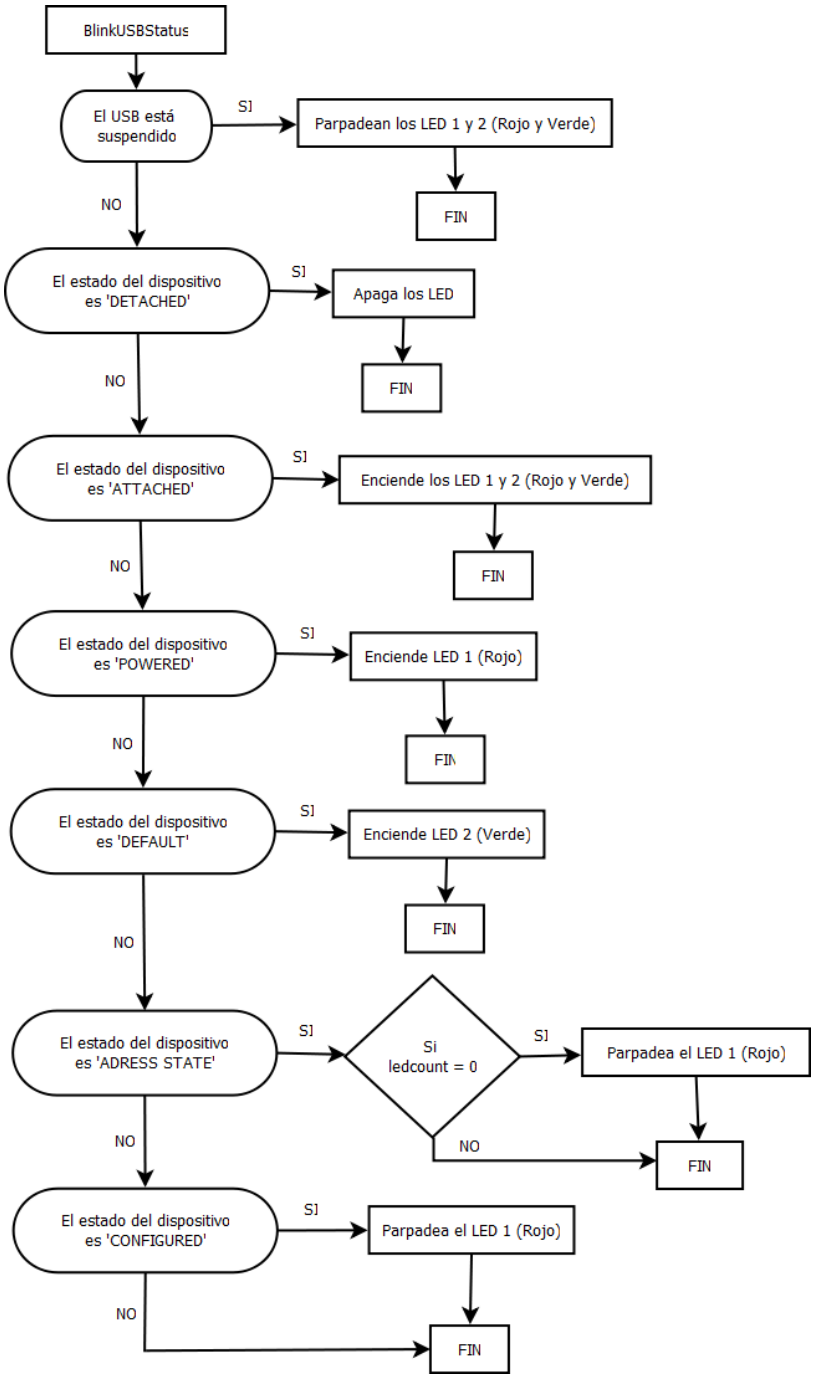


Figura 49 – Cap.6.2: Diagrama de la función BlinkUSBStatus

Esta función es la que activa y configura todo el módulo CAN. Desde aquí configuraremos el reloj del módulo CAN, estableceremos la función en baudios, asignaremos un buffer de memoria al módulo CAN, estableceremos el canal 0 para transmitir con un tamaño de 8 mensajes y

prioridad media y el canal 1 para recepción con un tamaño de 8 mensajes también. También configuraremos los filtros y máscaras y habilitaremos los eventos e interrupciones (ejemplo: RX_EVENT), asignaremos las prioridades y finalmente se pondrá el módulo CAN en módulo normal para que esté operativa.

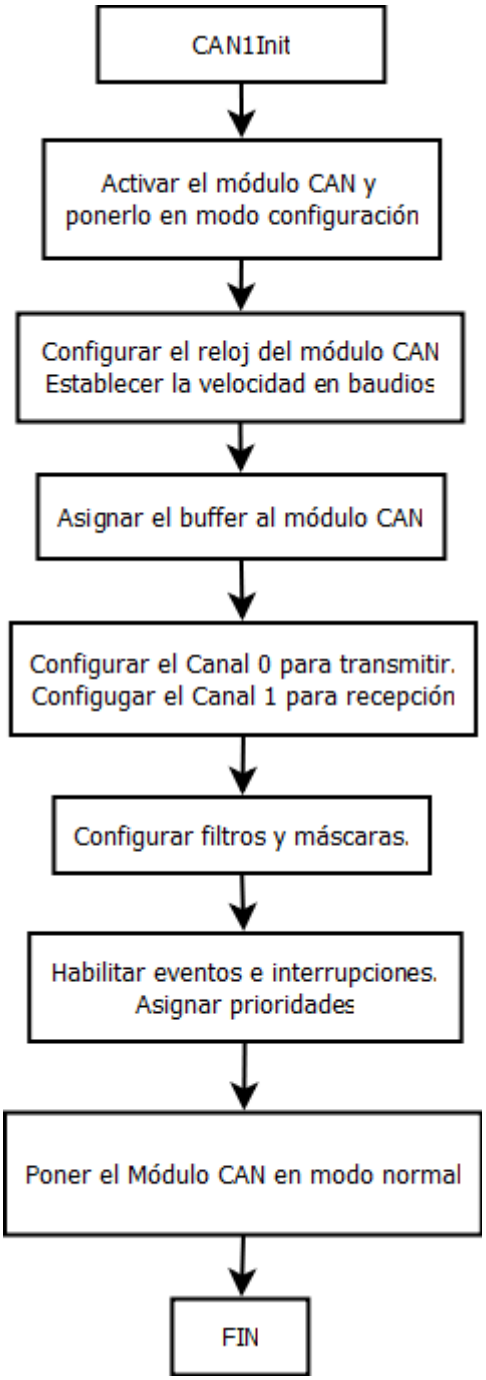


Figura 50 – Cap.6.2: Diagrama de la función CAN1Init

La siguiente función (CAN1RxMsgProcess) es la que desarrolla la recepción de tramas. En resumen el proceso es el siguiente: comprobamos si se ha recibido una trama, guardamos la trama, comprobamos el tipo de trama que es y según el tipo de trama mostraremos por pantalla unos campos u otros. Después se desactivará la bandera para que se vuelva a detectar si llega otro mensaje y por último el LED 3 durante unos segundos para que el usuario vea que se ha recibido una trama.

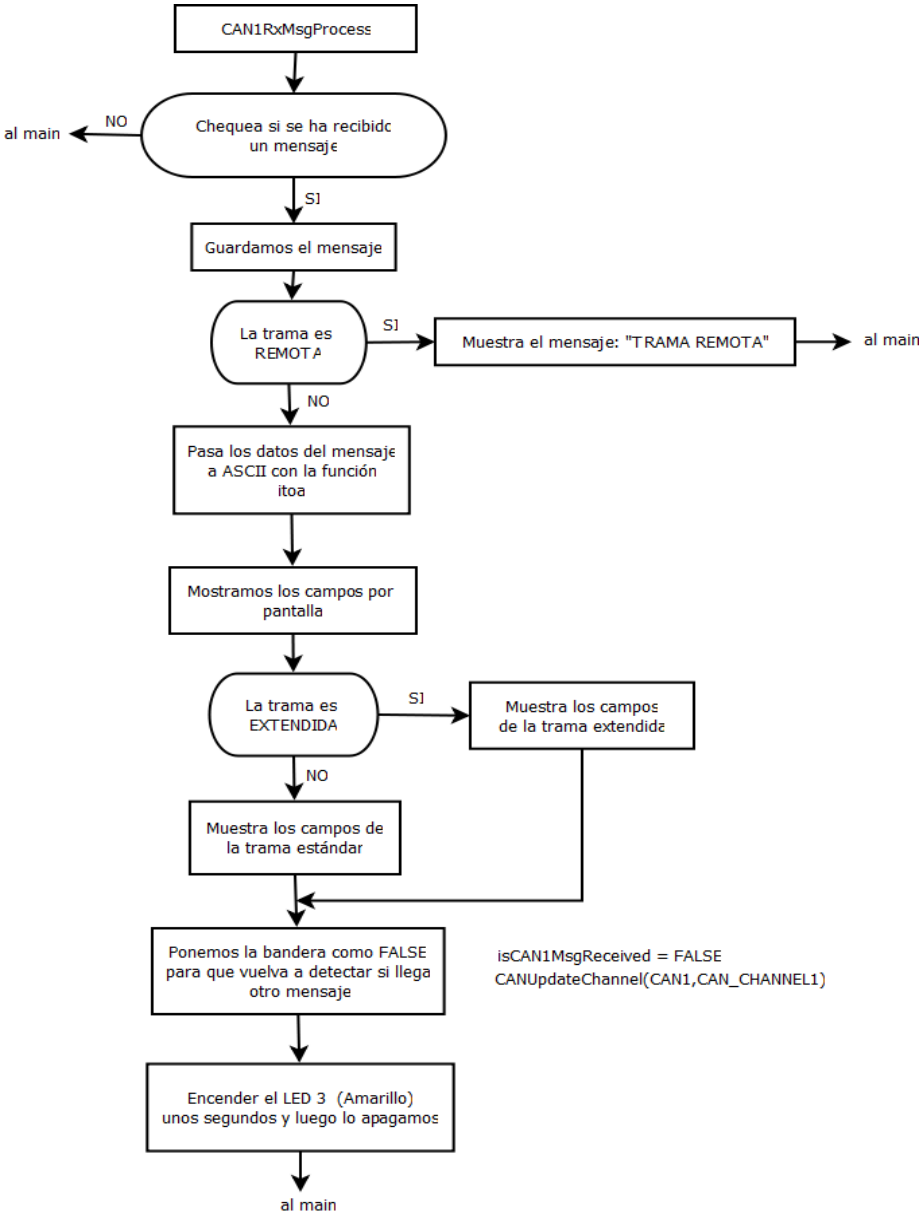


Figura 51 – Cap.6.2: Diagrama de la función CAN1RxMsgProcess

En el siguiente diagrama representamos la función (CAN1TxSendLEDMsg) que es la que gestiona la transmisión de tramas. Dependiendo del tipo de trama que vayamos a enviar (remota o de datos) se realizará un proceso u otro como podremos ver en la figura 46.

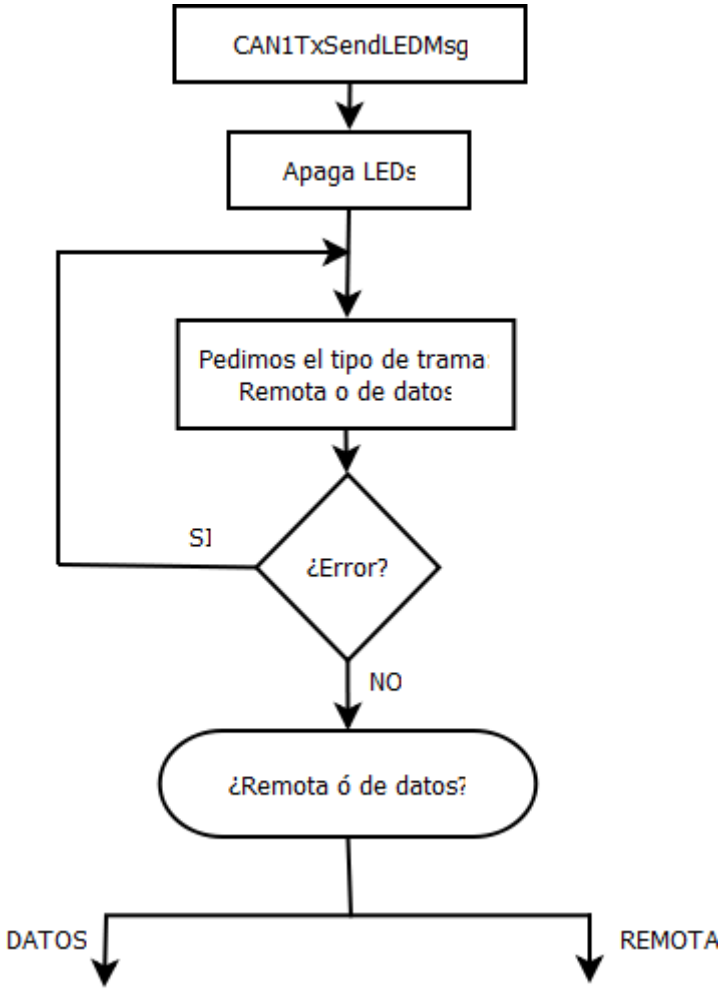


Figura 52 – Cap.6.2: Diagrama básico de la función CAN1TxSendLEDMsg

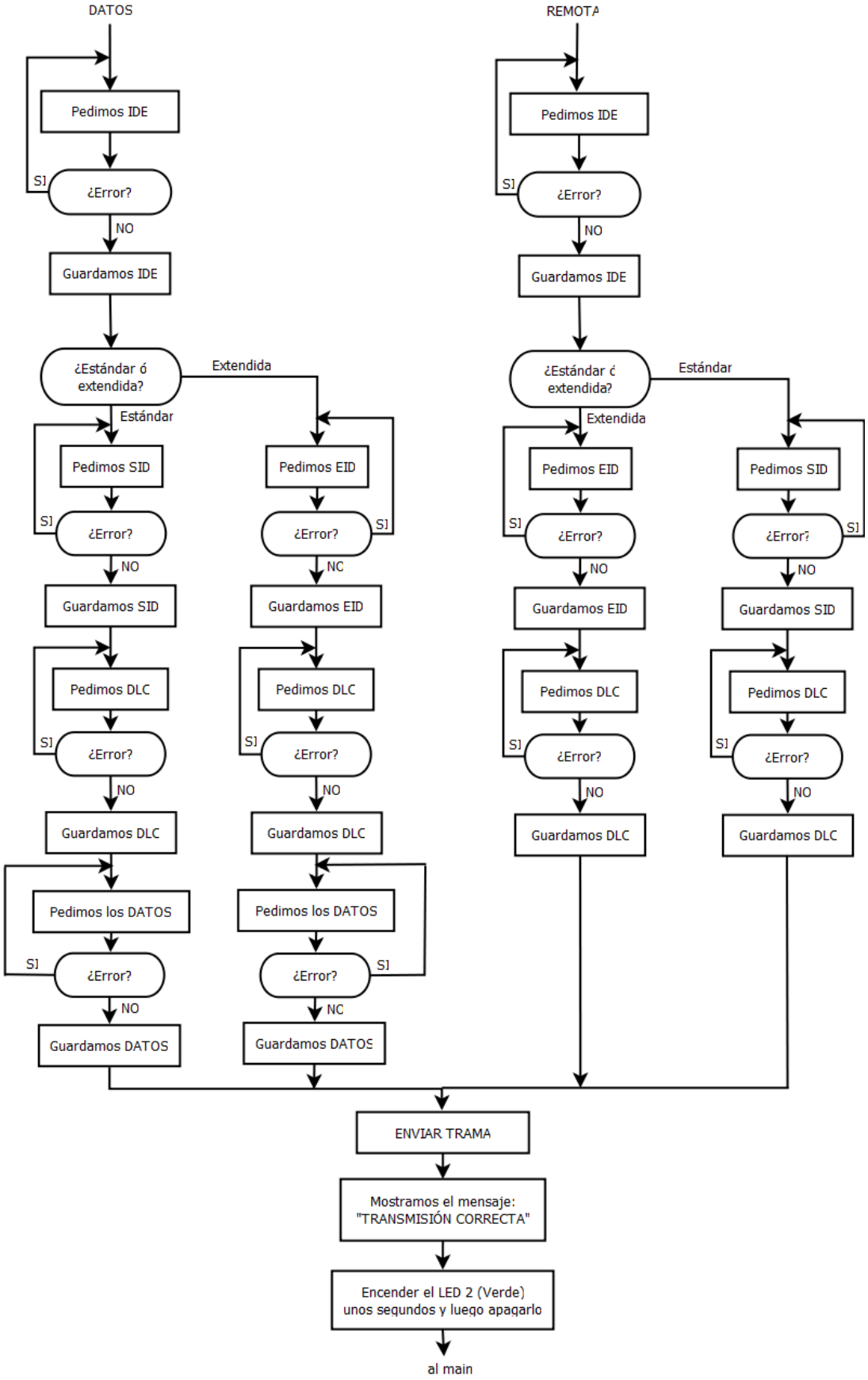


Figura 53 – Cap.6.2: Diagrama del proceso de transmisión de tramas

Como podemos observar en la figura anterior, al transmitir una trama de datos ó remota tenemos la opción de enviar una trama extendida o estándar.

En esta función se encargará de ir pidiendo los campos de cada trama al usuario (SID, EID, DLC, Datos…) a medida que el usuario los vaya introduciendo correctamente, para finalizar con un mensaje por pantalla que nos avise de que la transmisión se ha realizado correctamente a la vez que se visualiza el encendido del LED 2 (Verde) durante unos segundos.

Por último, en la figura 47 se representa el diagrama de la función CAN1InterruptHandler. Desde aquí gestionaremos las interrupciones del módulo CAN. Un módulo CAN tiene muchas fuentes de eventos, pero en nuestro caso solo habilitaremos el RX_EVENT. Primero se comprueba que, efectivamente, la fuente de la interrupción es RX_EVENT, comprobaremos si hay un evento en el módulo CAN y que la fuente de la interrupción es RX_EVENT. Si el evento es causado por el Canal 1(Recepción) entonces es que se ha recibido un mensaje. Después se desactivará la bandera de recepción. Si no hay un evento en CAN1 ó no es RX_EVENT se desactiva la bandera de interrupción.

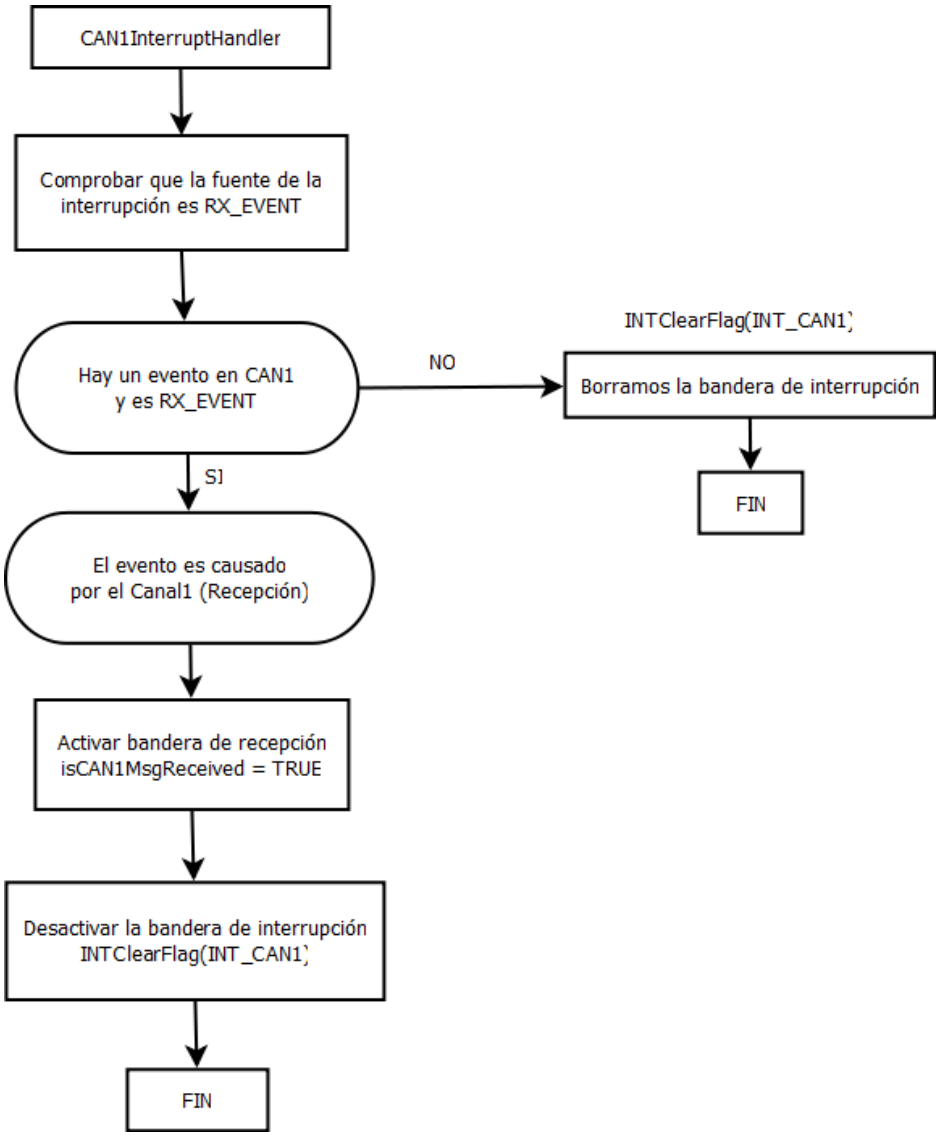
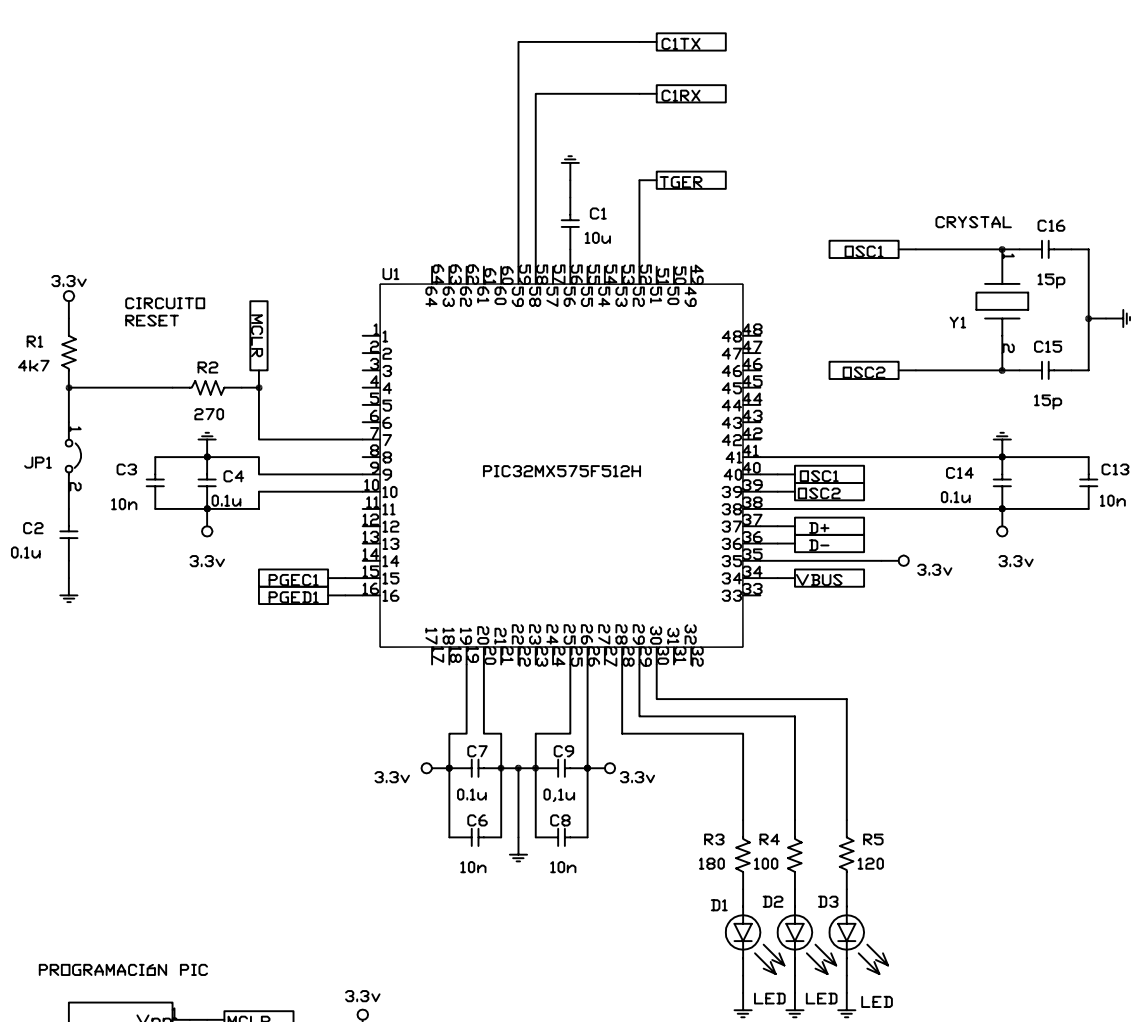
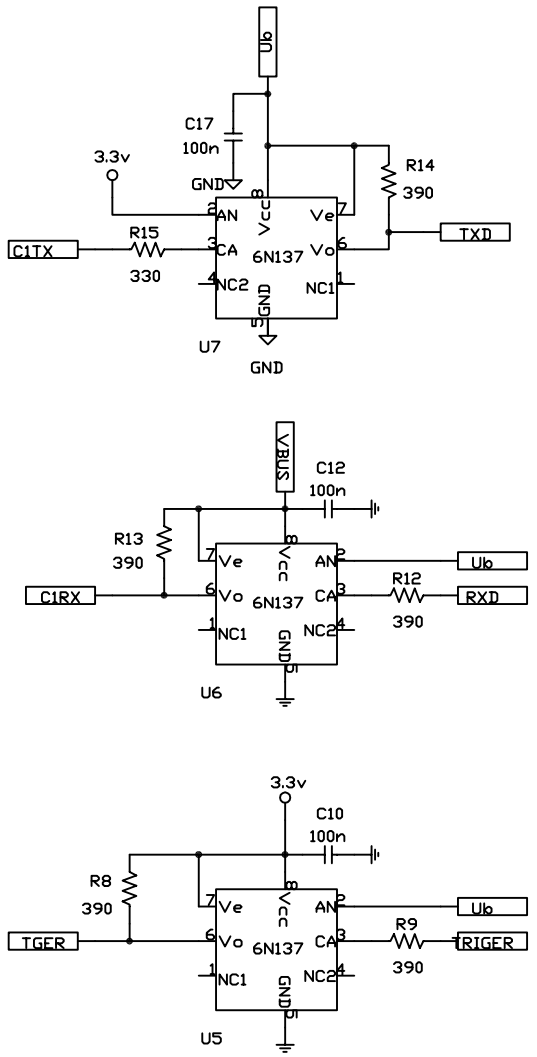


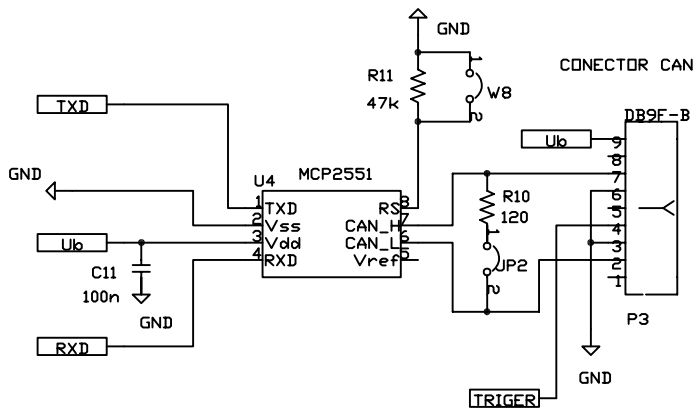
Figura 54 – Cap.6.2: Diagrama de la función CAN1InterruptHandler

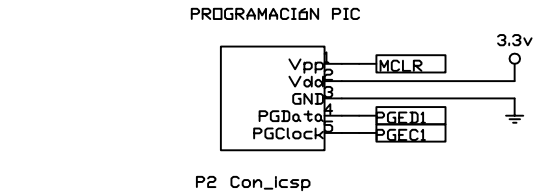
7.- PLANOS

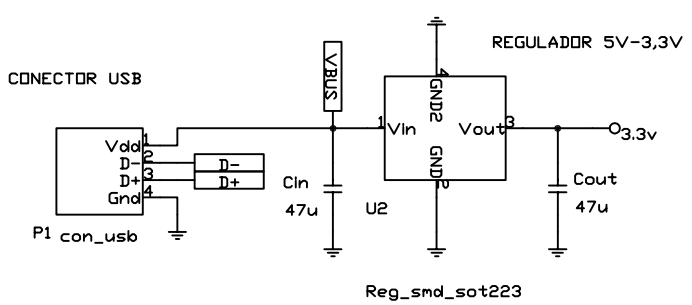
PLANO 1

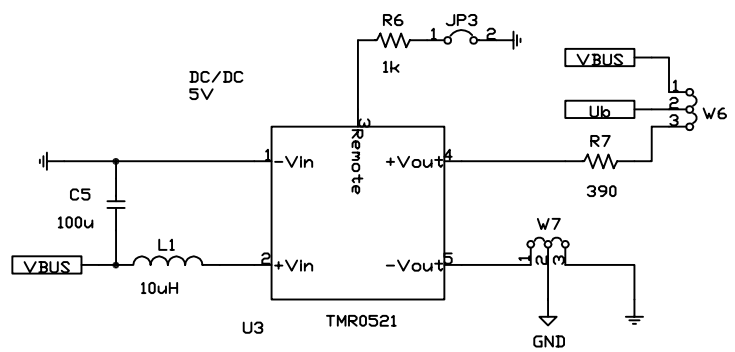








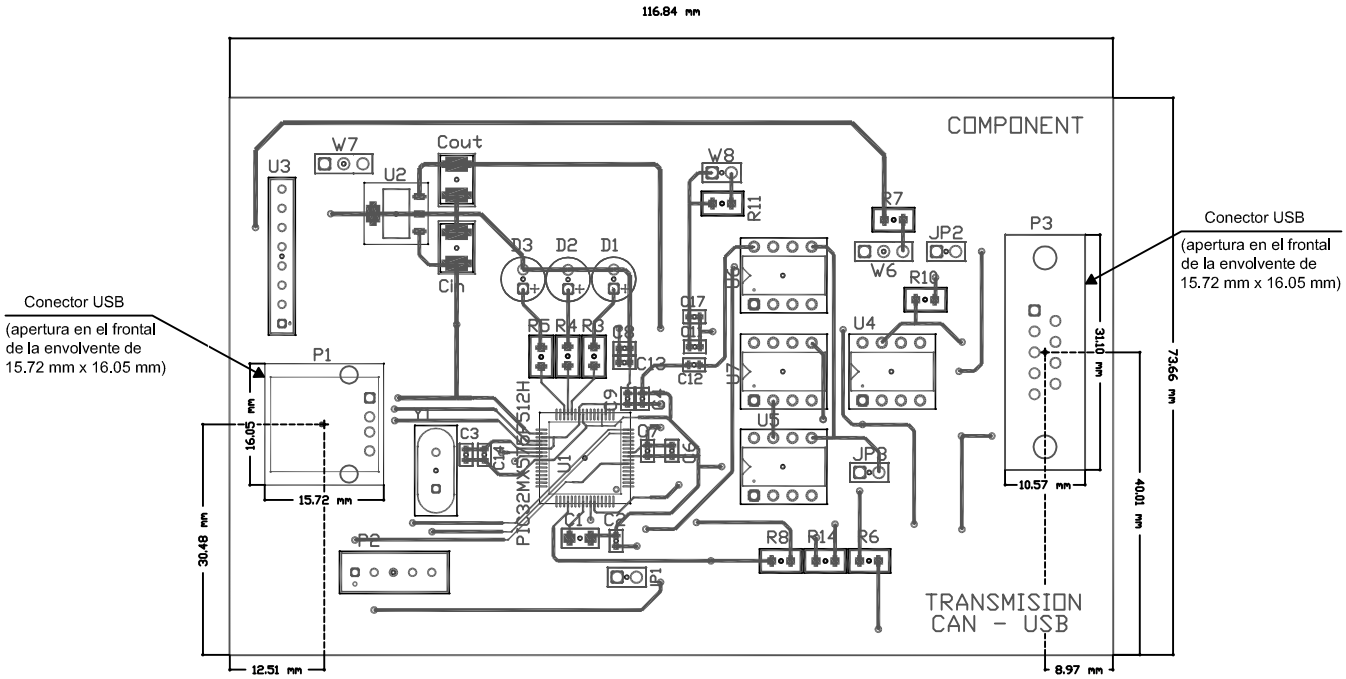




UNIVERSIDAD DE VALLADOLID
ESCUELA DE INGENIERÍAS INDUSTRIALES

DEP. DE TECNOLOGÍA ELECTRÓNICA	DENOM. PLANO: ESQUEMA ELÉCTRICO
TÍTULO: DESARROLLO DE UN INTERFACE USB - CAN	PLANO Nº: 1
AUTOR: ALVEAR GRANJA, FCO. JAVIER HERNANDO ESTEBAN, JUAN FCO.	ESCALA:
	FECHA: JUNIO 2012

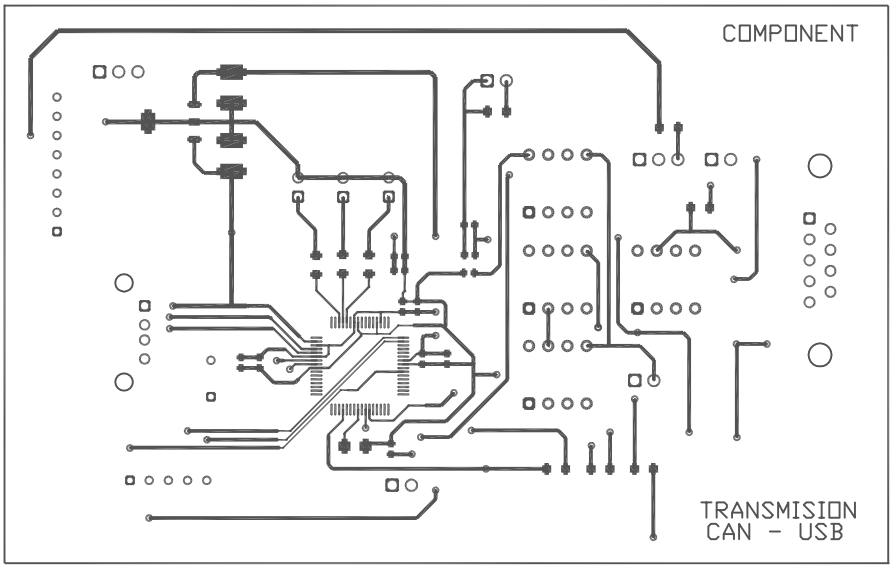
PLANO 2



UNIVERSIDAD DE VALLADOLID
ESCUELA DE INGENIERÍAS INDUSTRIALES

DEP. DE TECNOLOGÍA ELECTRÓNICA	DENOM. PLANO: DIMENSIONES
TÍTULO: DESARROLLO DE UN INTERFACE USB - CAN	PLANO Nº: 2
AUTOR: ALVEAR GRANJA, FCO. JAVIER HERNANDO ESTEBAN, JUAN FCO.	ESCALA: 1:1 FECHA: JUNIO 2012

PLANO 3

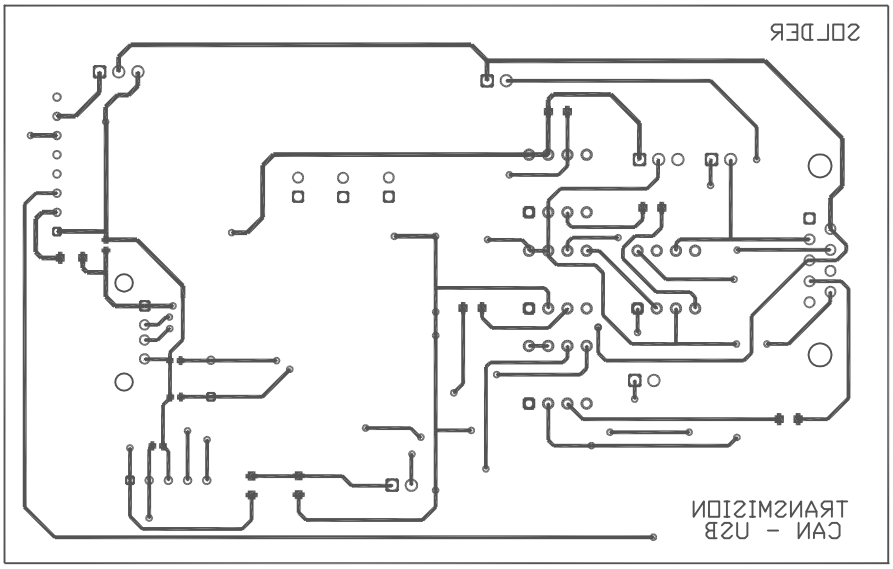


UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍAS INDUSTRIALES

DEP. DE TECNOLOGÍA ELECTRÓNICA	DENOM. PLANO: RUTADO CAPA COMPONENTES
TÍTULO: DESARROLLO DE UN INTERFACE USB - CAN	PLANO Nº: 3
AUTOR: ALVEAR GRANJA, FCO. JAVIER HERNANDO ESTEBAN, JUAN FCO.	ESCALA: 1:1 FECHA: JUNIO 2012

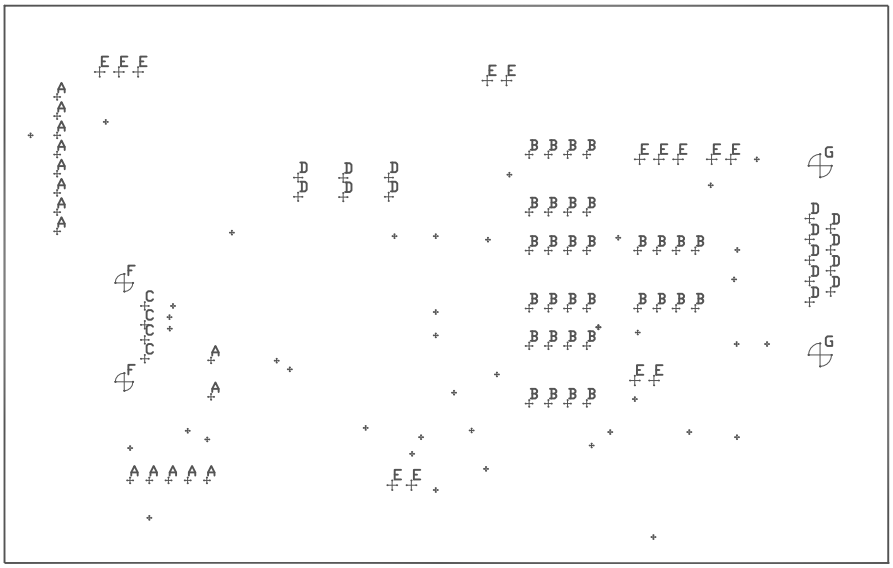
PLANO 4



UNIVERSIDAD DE VALLADOLID
ESCUELA DE INGENIERÍAS INDUSTRIALES

DEP. DE TECNOLOGÍA ELECTRÓNICA	DENOM. PLANO: RUTADO CAPA SOLDADURA
TÍTULO: DESARROLLO DE UN INTERFACE USB - CAN	PLANO Nº: 4
AUTOR: ALVEAR GRANJA, FCO. JAVIER HERNANDO ESTEBAN, JUAN FCO.	ESCALA: 1:1 FECHA: JUNIO 2012

PLANO 5



TALADRO	DIÁMETRO	SÍMBOLO	PLATEADOS	NO PLATEADOS	TOTAL
T1	20mil	+	42		42
T2	30mil	A	15		15
T3	35mil	B	32		32
T4	40mil	C	4		4
T5	43mil	D	15		15
T6	50mil	E	14		14
T7	91mil	F		2	2
T8	120mil	G		2	2
		TOTAL:	**122**	**4**	**126**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍAS INDUSTRIALES

DEP. DE TECNOLOGÍA ELECTRÓNICA

DENOM. PLANO: TALADROS

TÍTULO: DESARROLLO DE UN INTERFACE USB - CAN

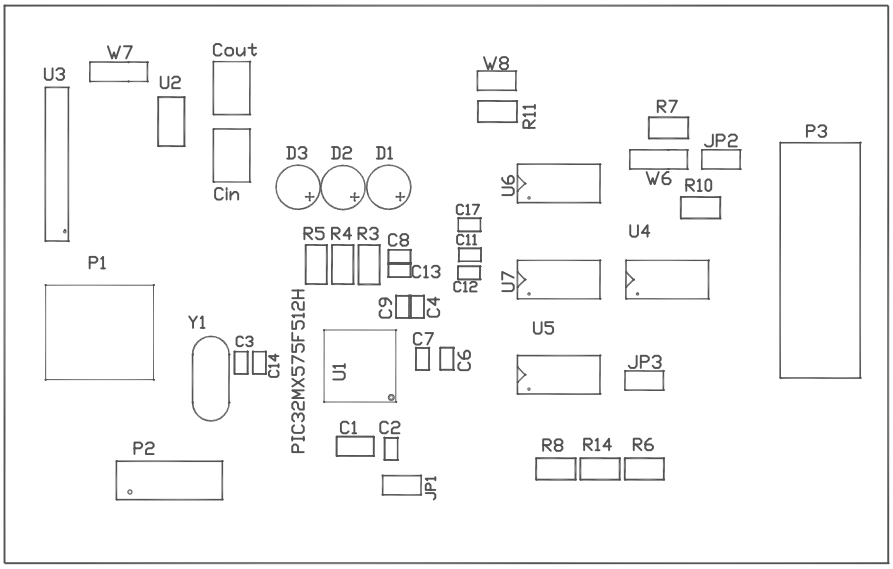
PLANO Nº: 5

AUTOR: ALVEAR GRANJA, FCO. JAVIER
HERNANDO ESTEBAN, JUAN FCO.

ESCALA: 1:1

FECHA: JUNIO 2012

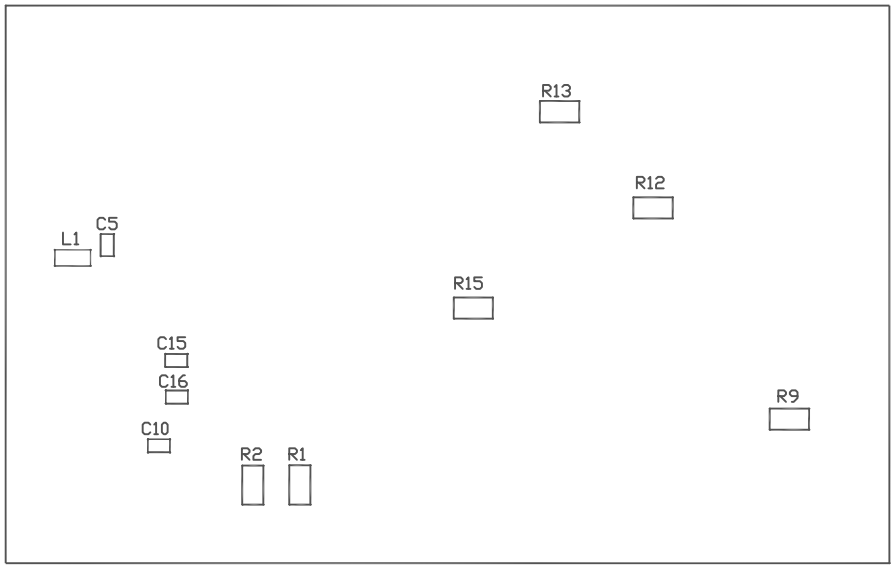
PLANO 6



UNIVERSIDAD DE VALLADOLID
ESCUELA DE INGENIERÍAS INDUSTRIALES

DEP. DE TECNOLOGÍA ELECTRÓNICA	DENOM. PLANO: CAPA DE SERIGRAFÍA SUPERIOR
TÍTULO: DESARROLLO DE UN INTERFACE USB - CAN	PLANO Nº: 5
AUTOR: ALVEAR GRANJA, FCO. JAVIER HERNANDO ESTEBAN, JUAN FCO.	ESCALA: 1:1 FECHA: JUNIO 2012

PLANO 7

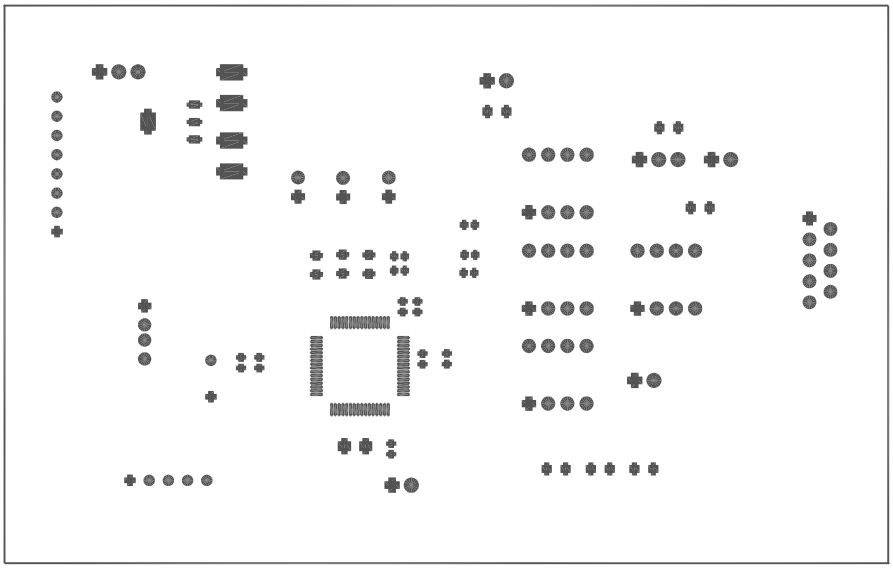


UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍAS INDUSTRIALES

DEP. DE TECNOLOGÍA ELECTRÓNICA	DENOM. PLANO: CAPA DE SERIGRAFÍA INFERIOR
TÍTULO: DESARROLLO DE UN INTERFACE USB - CAN	PLANO Nº: 7
AUTOR: ALVEAR GRANJA, FCO. JAVIER HERNANDO ESTEBAN, JUAN FCO.	ESCALA: 1:1 FECHA: JUNIO 2012

PLANO 8

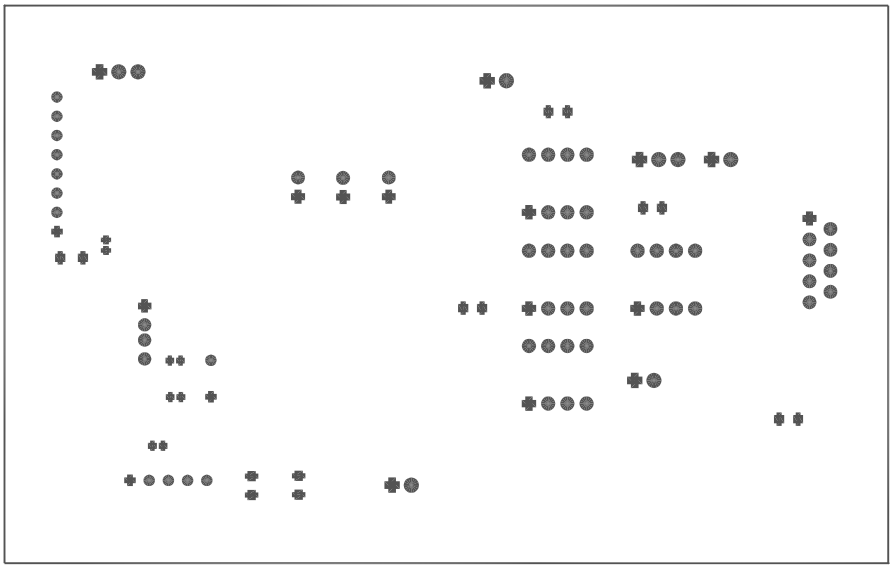


UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍAS INDUSTRIALES

DEP. DE TECNOLOGÍA ELECTRÓNICA	DENOM. PLANO: CAPA MÁSCARA DE SOLDADURA SUPERIOR
TÍTULO: DESARROLLO DE UN INTERFACE USB - CAN	PLANO Nº: 8
AUTOR: ALVEAR GRANJA, FCO. JAVIER HERNANDO ESTEBAN, JUAN FCO.	ESCALA: 1:1 FECHA: JUNIO 2012

PLANO 9



UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍAS INDUSTRIALES

DEP. DE TECNOLOGÍA ELECTRÓNICA	DENOM. PLANO: CAPA MÁSCARA DE SOLDADURA INFERIOR
TÍTULO: DESARROLLO DE UN INTERFACE USB - CAN	PLANO Nº: 9
AUTOR: ALVEAR GRANJA, FCO. JAVIER HERNANDO ESTEBAN, JUAN FCO.	ESCALA: 1:1 FECHA: JUNIO 2012

7.10.-LISTADO DE COMPONENTES

	UDS.	DISPOSITIVO	IDENTIFICADOR	ENCAPSULADO	REFERENCIA
1	3	Optoacoplador	6N137	DIP8	U5,U6,U7
2	4	Condesadores	C	C1310 (0.1u)	C2,C4,C7, C14
3	4		C	C1310 (10n)	C3,C6,C8, C13
4	1		C	C1310 (100u)	C5
5	1		C	C1310 (0,1u)	C9
6	4		C	C1310 (100n)	C10, C11, C12, C17
7	2		C	C1310 (15p)	C15, C16
8	2		C	C4532 (47u)	Cin, Cout
9	1		C	C3216 (10u)	C1
10	1	Conectores	Con_icsp	SIP5	P2
11	1		con_usb	USB_A	P1
12	1		DB9 F-B	DSHELL09-F	P3
13	1	Oscilador	CRYSTAL	HC-18U	Y1
14	4	Jumper	JUMPER1X2	HDR2	JP1, JP2, JP3,W8
15	2		JUMPER1X3	HDR1X3	W6,W7
16	1	Bobina	l	L3216C (10uH)	L1
17	3	Led	LED	T-1_3/4/WIRE_ENDED	D1,D2,D3
18	1	Transceptor	MCP2551	DIP8	U4
19	1	Microprocesador	PIC32MX575F512H	QFP	U1
20	6	Resistencias	R	R1206 (390)	R7,R8,R9, R12,R13, R14
21	1		R	R1206 (47k)	R11
22	1		R	R1206 (330)	R15
23	1		R	R1206 (100)	R4
24	2		R	R1206 (120)	R5, R10
25	1		R	R1206 (180)	R3
26	1		R	R1206 (4k7)	R1
27	1		R	R1206 (1k)	R6
28	1		R	R1206 (270)	R2
29	1	Regulador	Reg_smd_sot223	REGULADOR	U2
30	1	Convertidor DC/DC	TMR521	sip8	U3

Tabla 5 – Cap.7.10: Listado de componentes

7.11.- LOCALIZACIÓN DE COMPONENTES

SIMBOLO	ORIGEN (x,y)mil	EMPLAZAMIENTO (x,y)mil	ROTACIÓN	CAPA
Y1	(1124, 865)	(1124, 960)	90	Component
JP3	(3330.26, 950)	(3380.26, 950)	0	Component
U6	(2780.26, 1825)	(2930.26, 1975)	0	Component
C15	(965.559, 1054)	(938, 1054)	0	Solder
C16	(967.559, 864)	(940, 864)	0	Solder
P2	(702.5, 430)	(902.5, 430)	0	Component
U7	(2780.26, 1325)	(2930.26, 1475)	0	Component
P3	(4240.76, 1793)	(4295.26, 1575)	90	Component
U5	(2780, 830)	(2930, 980)	0	Component
C10	(819.441, 610)	(847, 610)	180	Solder
W8	(2562, 2510)	(2612, 2510)	0	Component
U4	(3345.26, 1325)	(3495.26, 1475)	0	Component
C7	(2225.37, 1089)	(2225.37, 1062)	270	Component
C13	(2078.56, 1596)	(2106.12, 1596)	0	Component
C8	(2133.12, 1522)	(2105.56, 1522)	180	Component
C12	(2495.12, 1510)	(2467.56, 1510)	180	Component
C11	(2444.56, 1604)	(2472.12, 1604)	0	Component
C17	(2442.56, 1760)	(2470.12, 1760)	0	Component
C1	(1819.44, 608)	(1874.56, 608)	0	Component
JP1	(2068, 405)	(2118, 405)	0	Component
C3	(1281, 1070.12)	(1281, 1042.56)	270	Component
C14	(1376, 1070.12)	(1376, 1042.56)	270	Component
U1	(2126.38, 877.3)	(1900, 1025)	90	Component
D1	(2050, 1906)	(2050, 1956)	90	Component
D2	(1812, 1904)	(1812, 1954)	90	Component
D3	(1578, 1906)	(1578, 1956)	90	Component
P1	(779, 1337.8)	(529, 1200)	270	Component
U2	(1039, 2205)	(917, 2295)	90	Component
Cout	(1232, 2474)	(1232, 2474)	90	Component
Cin	(1232, 2118)	(1232, 2118)	270	Component
W6	(3355.26, 2100)	(3455.26, 2100)	0	Component
JP2	(3730.26, 2100)	(3780.26, 2100)	0	Component
C5	(578, 1626.56)	(578, 1654.12)	90	Solder
L1	(457.055, 1588)	(398, 1588)	0	Solder
C6	(2353, 1090.56)	(2353, 1063)	270	Component
C2	(2062, 621.559)	(2062, 594)	270	Component
W7	(544, 2556)	(644, 2556)	0	Component
U3	(322, 1725)	(322, 2075)	90	Component

R11	(3826, 3112)	(2614, 2350)	180	Component
R13	(4142.99, 1585)	(2930.99, 2347)	0	Solder
R7	(2295.63, 1502)	(3507.63, 2264)	0	Component
R10	(4883.26, 2612)	(3671.26, 1850)	180	Component
R6	(4591, 1253)	(3379, 491)	180	Component
R14	(1940, -273)	(3152, 489)	0	Component
R8	(1709, -273)	(2921, 489)	0	Component
R5	(910.016, 2763)	(1672.02, 1551)	270	Component
R4	(2573, 343)	(1811, 1555)	90	Component
R3	(2710, 342)	(1948, 1554)	90	Component
C4	(2199, 1306.32)	(2199, 1333.88)	90	Component
C9	(2122, 1361.44)	(2122, 1333.88)	270	Component
R12	(2210.26, 2610)	(3422.26, 1848)	180	Solder
R15	(1274, 2089.98)	(2486, 1327.98)	180	Solder
R9	(2920, 1512)	(4132, 750)	180	Solder
R1	(2343.76, 1617)	(1581.76, 405.2)	270	Solder
R2	(2099, 1616)	(1337, 404)	270	Solder

Tabla 6 – Cap.7.11: Localización de componentes

8.- PLIEGO DE CONDICIONES

8.1.-DISPOSICIONES Y ABARQUE DEL PLIEGO DE CONDICIONES

8.1.1.-OBJETIVO DEL PLIEGO

El objetivo de este proyecto es el Desarrollo de un Interface USB-CAN.

Este proyecto es un proyecto de diseño orientado a la posible industrialización del aparato. Esto implica que el diseño haya tenido en cuenta la accesibilidad y la fiabilidad sin omitir su desarrollo industrial. En caso de una futura aplicación industrial se debería tener presente el pliego de condiciones, que tiene como principal función regular las condiciones entre las partes contratantes considerando los aspectos técnicos, facultativos, económicos y legales.

El pliego de condiciones define entre otros los siguientes aspectos:

- Obras que componen el proyecto.
- Características exigibles a los materiales y componentes.
- Detalle de la ejecución.
- Programa de obras.

Dado el amplio abanico de detalles tratados si se presentan dudas a la hora de poner en marcha el proyecto lo más recomendable es ponerse en contacto con el proyectista.

8.1.2.-DESCRIPCIÓN GENERAL DEL MONTAJE

Las diferentes partes que componen la obra a realizar por parte del instalador, poniendo especial énfasis en el orden establecido, no efectuando una actividad concreta sin haber realizado previamente la anterior.

- Encargo y compra de los componentes necesarios.
- Fabricación de la placa de circuito impreso.
- Montaje de los componentes de la placa.

- Puesta en marcha del equipo.

- Controles de Emisiones Electromagnética.

- Controles de calidad y fiabilidad.

- Mantenimiento para el correcto funcionamiento del sistema.

Todas las partes que en conjunto forman la obra de este proyecto, tendrán que ser ejecutadas por montadores cualificados, sometiéndose a las normas de la Comunidad Europea, países o incluso comunidades internacionales que se tengan previstas para este tipo de montajes, no haciéndose responsable el proyectista de los desperfectos ocasionados por su incumplimiento.

8.2.- NORMATIVA DE OBLIGADO CUMPLIMIENTO

En la siguiente lista se enumeran las normativas más relevantes que regulan el diseño, montaje y fabricación del regulador. No son las únicas y en todo caso siempre se seguirá las instrucciones expuestas en el Reglamento Electrotécnico de Baja Tensión de 2002.

- Directiva 73/23/CEE: Material de Baja Tensión.

- Directiva 89/336/CEE: Compatibilidad Electromagnética.

- Directiva 2001/95/CE: Seguridad General de Productos.

- Directiva 2002/95/CE de Restricción de ciertas Sustancias Peligrosas en aparatos eléctricos y electrónicos, (RoHS del inglés “Restricción of Hazardous Substances”).

- Directiva 2002/96/CE: Residuos de aparatos eléctricos y electrónicos (RAEE).

- Directiva 2003/108/CE: Modifica la Directiva 2002/96/CE sobre residuos de aparatos eléctricos y electrónicos (RAEE).

- Directiva 2004/108/CE: Legislación común en los estados miembros sobre la compatibilidad electromagnética.

- Directiva 2006/95/CE: relativa a la aproximación de las legislaciones de los Estados miembros sobre el material eléctrico destinado a utilizarse con determinados límites de tensión.

- RD.1580/2006: Por el que se regula la compatibilidad electromagnética en aparatos eléctricos y electrónicos.

- EN 61010-1:2001: Equipos de Medida Control. Requisitos generales para la seguridad.

- EN 61326:1998+A1:1998+A2:2001: Material eléctrico para la medida, control y uso en laboratorio. Requisitos de Compatibilidad Electromagnética (CEM).

- UNE 20-050-74 (I). Código para las marcas de resistencias y condensadores. Valores y tolerancias.

- UNE 20-524-75 (I): Técnica circuitos impresos. Parámetros fundamentales. Sistemas de cuadrícula.

- UNE 20-524. Equipos electrónicos y sus componentes. Soldabilidad de circuitos impresos.

- UNE 20-524-77 (II). Técnica de circuitos impresos. Terminología.

- UNE 20-531-73. Series de valores nominales para resistencias y condensadores.

- UNE 20-543-85 (I). Condensadores fijos de equipos electrónicos.

- UNE 20-545-89. Resistencias fijas para equipos electrónicos.

- UNE 20916: 1995: Estructuras mecánicas para equipos electrónicos. Terminología.

- UNE 21352: 1976: Explicación de las cualidades y funcionamientos de equipos de medida electrónicos.

- UNE-EN61000-4-3-1998: Compatibilidad electromagnética.

- EN123500: 1992: Especificación intermedia: placas de circuitos impresos flexibles con taladros para la inserción de componentes.

- EN 60715: 2001: Montaje estandarizado sobre carriles para apoyo mecánico de dispositivos eléctricos en switchgear e instalaciones controlgear. Específica exigencias dimensionales y funcionales para el montaje compatible de dispositivos variados eléctricos sobre algunos tipos de carriles.

8.3.-CONDICIONES DE LOS MATERIALES

En este apartado se explican las características técnicas exigibles de los componentes presentes en la ejecución de la obra.

La adquisición de los componentes debe realizarse teniendo en cuenta sus especificaciones técnicas, como a la hora de la obtención de las placas de circuito impreso, basándonos en las pautas expuestas en el punto “Normativa de obligado cumplimiento” y las que se concretan a continuación.

8.3.1.- ESPECIFICACIONES ELÉCTRICAS

REGLAMENTO ELECTROTÉCNICO DE BAJA TENSIÓN

Todos los aspectos técnicos de la instalación que, directa o indirectamente, estén incluidos en el Reglamento Electrotécnico de Baja Tensión, tendrán que cumplir lo que se disponga en las respectivas normas.

Las instrucciones más importantes relacionadas con la realización del proyecto son las siguientes:

- I.T.C.B.T.002: Normas de referencia en el RBT.

- I.T.C.B.T.019 a la B.T.024: Instalaciones interiores o receptoras.

- I.T.C.B.T.030: Instalaciones en locales de características especiales.

- I.T.C.B.T.036: Instalaciones a muy baja tensiones.

- I.T.C.B.T.037: Instalaciones a tensiones especiales.

- I.T.C.B.T.043: Instalaciones de receptores. Prescripciones generales.

- I.T.C.B.T.045: Instalaciones de receptores. Aparatos de caldeo.

- I.T.C.B.T.048: Instalaciones de Receptores. Transformadores y autotransformadores. Reactancias y rectificadores. Condensadores.

- I.T.C.B.T.051: Instalaciones de sistemas de automatización, gestiona técnica de la energía y seguridad para viviendas y edificios.

PLACAS DE CIRCUITO IMPRESO

Todos los circuitos se realizarán sobre placas de fibra de vidrio de sensibilidad positiva, en diferentes medidas, utilizándose de doble cara según el diseño, con un espesor mínimo de 1,7mm.

CONDUCTORES ELÉCTRICOS

Los conductores utilizados serán internos a excepción de la alimentación y de la interconexión entre las sondas y el regulador, que reunirán condiciones especiales requeridas para los conductores expuestos al exterior.

COMPONENTES PASIVOS

Los componentes pasivos utilizados en el proyecto son los disponibles tecnológicamente en el momento de la realización del proyecto. Las características técnicas se han introducido en el Anexo.

COMPONENTES ACTIVOS

Los componentes activos utilizados en el proyecto son los disponibles tecnológicamente en el momento de la realización del proyecto. Las características técnicas se han introducido en el Anexo.

RESISTENCIAS

Existen resistencias con una gran precisión en el valor, lo que implica fijar tolerancias muy bajas, pero se tendrá en cuenta que su precio aumenta considerablemente y serán necesarias en aplicaciones muy específicas como en nuestro caso. Estando normalmente destinadas a usos generales las tolerancias estandarizadas de 5%, 10% y 20%.

CIRCUITOS INTEGRADOS Y SEMICONDUCTORES

En este proyecto el microcontrolador (Microchip PIC32MX575F512H), los optoacopladores (6N137), convertidor DC/DC (TRACO POWER TMR0521), regulador de tensión 5V-3.3V (MCP 1826S), entre otros. Todos ellos se tendrán que alimentar a una tensión adecuada, las características de tensión y corriente de entrada-salida, tiempos de retardo, etc., se encuentran en las hojas del fabricante del Anexo.

8.3.2.- ESPECIFICACIONES MECÁNICAS

Todos los materiales escogidos son de una calidad que se adapta al objetivo del proyecto, no obstante si no se pudiera encontrar en el mercado algún producto por estar agotado, el instalador encargado del montaje tendrá que estar capacitado para su substitución por otro similar o equivalente.

La placa de circuito impreso se realizará en fibra de vidrio. Se recomienda el uso de zócalos torneados, para la inserción de componentes de agujero pasante. De esta forma se reduce el tiempo de reparación y además se disminuye el calentamiento de los pines de los componentes electrónicos en el proceso de soldadura que podría producir su deterioro.

8.4.- CONDICIONES DEL PROCESO DE FABRICACIÓN

8.4.1.- PREPARACIÓN DE COMPONENTES

La adquisición de los componentes debe realizarse teniendo en cuenta sus especificaciones técnicas, como a la hora de la obtención de las placas de circuito impreso, basándonos en las pautas anteriores.

La compra de los materiales, componentes y aparatos necesarios tendrá que realizarse con el tiempo necesario, de manera que estén disponibles a la hora de comienzo del ensamblaje de los componentes.

8.4.2.- MATERIAL DEL CIRCUITO IMPRESO

El material elegido es la fibra de vidrio, de un espesor de 1,7mm obteniendo así mayor resistencia y a los cambios climáticos y mecánicos.

Los materiales y aparatos para la realización de la placa de circuito impreso son: insoladora (ó lámpara de luz actínica), revelador (ó en su

defecto disolución de sosa cáustica y agua, atacador rápido que se puede sustituir por una disolución con la siguiente composición: 33% de HLC, 33% de agua oxigenada de 110 volúmenes y 33% de agua destilada), y por último se necesitan la placa de circuito impreso de material fotosensible positivo de doble cara y fibra de vidrio.

Para la fabricación en serie se recomienda contar con un sistema de fresado indicado para ello o con sistemas semiautomáticos o automáticos.
Una vez se termine de realizar la fabricación de la placa de circuito impreso, se realizarán los agujeros para soldar los terminales de agujero pasante y las vías.

8.4.3.- SOLDADURA Y MONTAJE DE COMPONENTES

El proceso de montaje de los componentes electrónicos, se realizará siguiendo de manera exhaustiva las pautas marcadas en el diseño del circuito. Consultar el capítulo de PLANOS.

Se debe tener muy en cuenta la manipulación de los componentes, ya que algunos de los elementos que componen el circuito son sensibles a descargas electrostáticas, por tanto se realizará la manipulación de los componentes, con las medidas necesarias para evitar este tipo de descargas.
Debe tenerse especial cuidado con el material a la hora de su transporte e instalación en circuito impreso.

Para la realización de las vías hay distintos métodos pero preferimos el de por deposición. Puede hacerse con el PCB ya grabado o no, pero con los agujeros ya hechos. Se le coloca una máscara autoadhesiva de ambas caras y luego por serigrafía se aplica a ambos lados la pasta de soldar en base de plata, quedando los agujeros tapados. Luego se absorbe el sobrante en una mesa de vacío dejando solo una capa en la pared interna de los agujeros y se hornea a la temperatura específica que solidifica la pasta para así crear la vía entre las dos caras.

Los componentes SMD deberán ser fijados con pasta soldadora y verificados antes de proceder con la soldadura.

En el proceso de fabricación será necesario realizar una soldadura de baño por ola cumpliendo la normativa RoHS. Se recomienda aplicar una soldadura por ola doble o turbulenta seguida por una de ola laminar dado el gran número de elementos de soldadura superficial.

8.4.4.- CONDICIONES PARA EL PROCESO DE PRUEBA

Una vez terminada la fase de montaje del dispositivo se pasará a realizar al 100% de los dispositivos un test del correcto funcionamiento del equipo según las especificaciones.

Se someterá al equipo a ensayos de compatibilidad electromagnética tanto radiados como inducidos para comprobar que el dispositivo es inmune a las radiaciones procedentes de elementos ajenos, comprobando que no se produce variación alguna con respecto a su modo de funcionamiento normal.

En todo caso se seguirá la normativa vigente sobre compatibilidad electromagnética (Directiva 2004/108/CE y su transposición R.D.1580/2006 por el que se regula la compatibilidad electromagnética en aparatos eléctricos y electrónicos).

Antes de la comercialización del producto, y por tanto del certificado CE, se deberá realizar un proceso de evaluación de conformidad con las distintas normativas que son de aplicación al regulador.

8.4.5.- CONDICIONES FACULTATIVAS

Los permisos de carácter obligatorio necesarios para realizar el proyecto o la utilización de la misma tendrán que obtenerse por parte de la empresa contratante, quedando la empresa contratista al margen de todas las consecuencias derivadas de la misma.

Cualquier retardo producido en el proceso de fabricación por causas debidamente justificadas, siendo estas alienas a la empresa contratista, será aceptada por el contratante, no teniendo este último derecho a reclamación por daños o perjuicios.

Cualquier demora no justificada supondrá el pago de una multa por valor del 6% del importe total de fabricación, para cada fracción del retardo temporal (acordado en el contrato).

La empresa contratista se compromete a proporcionar las mayores facilidades al contratista para que la obra se realice de una forma rápida y adecuada.

El aparato cumplirá los requisitos mínimos respecto al proyecto encargado, cualquier variación o mejora sustancial en el contenido del mismo tendrá que ser consultada con el técnico diseñador (proyectista).

Durante el tiempo que se haya estimado la instalación, el técnico proyectistas podrá anunciar la suspensión momentánea si así lo estimase oportuno.

Las características de los elementos y componentes serán los especificados en la memoria y el pliego de condiciones, teniendo en cuenta su perfecta colocación y posterior uso.

La contratación de este proyecto se considerará válida una vez que las dos partes implicadas, propiedad y contratista, se comprometan a concluir las cláusulas del contrato, por el cual tendrán que ser firmados los documentos adecuados en una reunión conjunta en haber llegado a un acuerdo.

Los servicios de la empresa contratista se consideran finalizados desde el mismo momento en que el aparato se ponga en funcionamiento, después de la previa comprobación de su correcto funcionamiento.

El presupuesto no incluye los gastos de tipo energético ocasionados por el proceso de instalación, ni las obras que fuesen necesarias, que irán a cargo de la empresa contratante.

El cumplimiento de las elementales comprobaciones por parte de la empresa instaladora, no será competencia del proyectista, el cual queda fuera de toda responsabilidad derivada del incorrecto funcionamiento del equipo como consecuencia de esta omisión.

8.4.6.- SOLICITUD DE HOMOLAGACIÓN DE TIPO CE

El marcado CE indica que un producto es presuntamente conforme con todas las disposiciones de las directivas que son de aplicación al equipo en cuestión. Igualmente, garantiza que el fabricante ha tomado todas las medidas oportunas para garantizar el cumplimiento de las mismas en cada uno de los productos comercializados. Por lo tanto, tanto el fabricante como el producto cumplen con los requisitos esenciales de las directivas de aplicación.

Es totalmente indispensable que todo producto comercializado o puesto en servicio posea el correspondiente marcado CE. Esto no implica que todo producto deba llevar el marcado CE, ya que sólo es obligatorio que lo posean únicamente aquellos productos que estén regulados por directivas comunitarias de marcado CE.

El caso que nos ocupa se rige principalmente por la Directiva 2006/95/CE sobre material de baja tensión, la Directiva 2004/108/CE sobre la compatibilidad electromagnética y la Directiva 2002/96/CE sobre residuos de aparatos eléctricos y electrónicos, pero siempre en línea con todas las normativas que le son de aplicación. No es aceptable la conformidad parcial, es decir, la conformidad con sólo algunas de las directivas aplicables. Cumpliendo los requisitos esenciales de estas normativas.

El fabricante es el responsable de los procedimientos de certificación y, en su caso, certificación de la conformidad de un producto. Básicamente tiene que:

- Garantizar el cumplimiento del producto con los requisitos esenciales de las Directivas de aplicación.

- Firmar la Declaración “CE” de conformidad.

- Elaborar la documentación o expediente técnico.

- Fijar el marcado “CE”.

8.4.6.1.- EXPEDIENTE TÉCNICO DE CONSTRUCCIÓN

El marcado CE lo debe poner siempre el fabricante o su representante legal autorizado, ya que éste es principal responsable de la comercialización o puesta en servicio del producto y de la garantía de su seguridad. Para ello debe realizar un ETC (Expediente Técnico de Construcción) que contará con la siguiente relación:

- Descripción general del producto.

- Análisis de los requisitos esenciales de la/s directivas aplicables.

- Análisis de riesgos. Descripción de las soluciones adoptadas para prevenir los riesgos presentados por el producto.

- Lista de las normas aplicadas total o parcialmente, y la descripción de las soluciones adoptadas para cumplir los aspectos de seguridad de la Directiva en cuestión, en los casos en que no hayan sido aplicadas las normas.

- Informes técnicos con los resultados de los ensayos efectuados o certificados obtenidos de un organismo o laboratorio competente. Tales informes de ensayo serán necesarios si el fabricante declara conformidad con una norma armonizada y podrán ser efectuados por él mismo o bien por un organismo o laboratorio competente. Resultados de los cálculos efectuados en el diseño, de los controles realizados, etc.

- Planos de diseño y de fabricación, y esquemas de los componentes, subconjuntos, circuitos, etc. Explicaciones y descripciones necesarias para la compresión de los mencionados planos y esquemas, y del funcionamiento del producto.

- Homogeneidad de la producción. Todas las medidas necesarias adoptadas por el fabricante para que el proceso de fabricación garantice la conformidad de los productos manufacturados.

8.4.6.2.- DECLARACIÓN DE CONFORMIDAD DEL PRODUCTO

Para certificar la conformidad del producto, el fabricante o su representante establecido en la Comunidad, deberá elaborar una Declaración de Conformidad.

- Nombre y dirección del fabricante o de su representante establecido en la Comunidad. En caso de productos fabricados fuera de la Comunidad, se deberá indicar tanto el nombre del fabricante como el nombre del representante legal. Se debe hacer constar la dirección completa de la sede o de una de las fábricas o la de uno de los establecimientos del país destino.

- Descripción del producto.

- Todas las disposiciones pertinentes a las que se ajuste el producto. Referencia a las Directivas de aplicación. Aunque no es obligatorio, también se puede incluir las referencias a las transposiciones

nacionales, es decir, referencia a los Reales Decretos que transponen las Directivas de aplicación.

- Referencia a las normas armonizadas. Aunque las normas armonizadas no son de obligatorio cumplimiento, al fabricante le interesa que se indiquen dichas normas, ya que dichas normas proporcionan al producto una presunción de conformidad con los requisitos esenciales de la Directiva. Se podrá hacer referencia a la norma europea o directamente a la norma nacional. Debido a que el estado normativo avanza continuamente, se debería indicar la edición y fecha de publicación de la norma en cuestión. Por otro lado, en caso de no utilizar dichas normas armonizadas, se deberá especificar el procedimiento alternativo empleado para satisfacer los requisitos esenciales.

- Identificación del signatario apoderado para vincular al fabricante o a su representante. Es necesario mencionar el nombre del signatario, ya que es una señal de autenticidad. Aunque no es obligatorio, también se suele incluir el lugar y fecha de la firma.

- Nombre y dirección del organismo notificado y número de certificación CE de tipo, si procede. Necesario para aquellas máquinas que hayan obtenido un examen CE de tipo de un organismo notificado.

- Nombre y dirección del organismo notificado al que se haya comunicado o que haya efectuado la comprobación del ETC, si procede.

8.4.6.3.- MARCADO CE SOBRE EL PRODUCTO

Una vez finalizado el proceso completo, el fabricante o representante legal puede proceder a poner el marcado CE sobre el producto. El marcado CE debe colocarse de modo visible, legible e indeleble sobre el equipo o su placa de características. En determinados casos, es aceptable que el marcado CE se ponga sobre el embalaje del producto.

Los diferentes elementos del marcado CE deberán tener una dimensión vertical apreciablemente igual, que no será inferior a 5mm. En caso de reducirse o aumentarse el tamaño del marcado CE, siempre deberán conservarse las proporciones del logotipo de la siguiente figura.

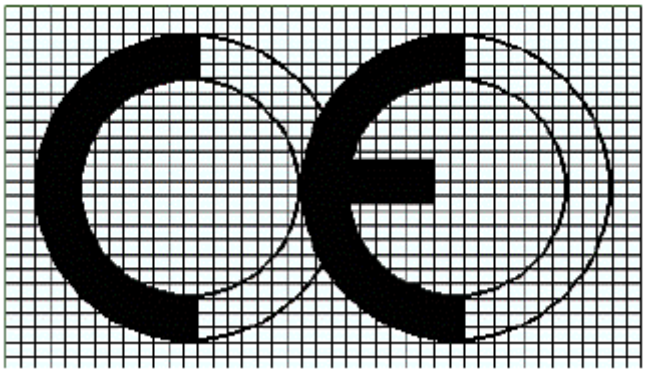


Figura 55 – Cap.8.4.6.3: Logotipo de marcado CE

8.4.7.- MARCA DE RECICLADO DE APARATOS ELÉCTRICOS Y ELECTRÓNICOS

A partir de la Directiva 75/442 se regula a nivel europeo la recogida de residuos. Por lo que se refiere al proyecto que nos atañe, la regulación la marca la Directiva 2002/96 sobre reciclado de aparatos eléctricos y electrónicos. En el artículo 2, refiriéndose al anexo 1 de este, se incluye el proyecto dentro del ámbito de aplicación incluido en el grupo 9 como Reguladores de Calefacción.

Este proyecto cumple la Directiva presente en lo que se refiere al diseño como se explica en el artículo 4 de la misma y se ha tenido en cuenta en la realización del diseño.

Por lo que se refiere al proceso de producción, se exige que los aparatos eléctricos y electrónicos se marquen con el símbolo específico. También se estipula que sean los productores o distribuidores los encargados de la recuperación de estos aparatos para valorarlos. Este punto puede ser de interés a la hora de reducir gastos, ya que la obligatoriedad existe y se debería estudiar un posible beneficio.

A partir de este proyecto será el fabricante el encargado de seguir debidamente el cumplimiento de la normativa.

8.4.7.1.- MARCA DE APARATOS ELÉCTRICOS Y ELECTRÓNICOS

El símbolo indica la recogida selectiva de aparatos eléctricos y electrónicos es el contenedor de basura tachado, tal como aparece

representado a continuación este símbolo se estampará de manera visible, legible e indeleble.

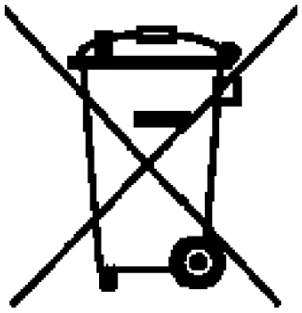


Figura 56 – Cap.8.4.7.1: Logotipo de marcado de aparatos eléctricos y electrónicos

8.4.8.- CONCLUSIONES

Las partes interesadas manifiestan que conociendo los términos de este Pliego de Condiciones y del proyecto adjunto, y están de acuerdo con lo que en él se manifiesta.

9.- BIBLIOGRAFÍA

1. DISEÑO DE LA PLACA (Datasheets)

- PIC32MX575MX512H: www.microchip.com
- MCP 2551: www.microchip.com
- Regulador 5v – 3.3v www.microchip.com
- Optoacopladores HP-6N137: www.datasheetcatalog.net
- Convertidor DC/DC (Traco Electronic AG) TMR 0521: www.tracopower.com
- Conector ICSP: www.farnell.com es.rs-online.com
- Conector USB: www.farnell.com es.rs-online.com
- Conector DB9: www.farnell.com es.rs-online.com
- Resistencias: www.farnell.com es.rs-online.com
- Bobinas: www.farnell.com es.rs-online.com
- Condensadores: www.farnell.com es.rs-online.com

2. PROTOCOLOS CAN BUS

- Bosch Controller Area Network – Version 2.0 Protocol Standard – MOTOROLA
- BOSCH – CAN Specification Version 2.0
- www.semiconductors.bosch.de/de/20/can/index.asp

- www.can.bosch.com
- http://es.wikipedia.org/wiki/CAN_bus
- http://www.can-cia.org/products/pg2005/html/index.htm
- http://www.canusb.com/index.htm

3. PROTOCOLO USB

- Trabajo de LSED: Protocolo USB aplicado a placas con microcontroladores. (Manuel Cabeza, Priscilla Cruz, Ángel Díaz, María José Fresneda, Mónica Jiménez, Inma Montell).

4. PROGRAMACIÓN

- www.microchip.com
- http://www.aquihayapuntes.com/curso-pic32/osciladores-timer.html
- http://picmind.es.tl/

5. OTRA INFORMACIÓN

- http://www.ixxat.de/
- http://www.kvaser.com/index.htm
- www.serconet.com/usrl/laureanog/HPV1B_31.HTM
- www.ihs.com.es

10.- ANEXOS

ANEXOS

EN EL DISCO QUE SE ENCUENTRA EN LA CONTRAPORTADA, SE INCLUYEN LAS SIGUIENTES CARPETAS CON LOS SIGUIENTES ARCHIVOS:

- CARPETA CON ARCHIVOS MICROSIM ENTRE LOS QUE SE INCLUYEN EL ESQUEMA ELÉCTRICO (SCHEMATICS) Y EL RUTADO CON LOS COMPONENTES (PCBOARD), ASI COMO TODOS LOS CORRESPONDIENTES A NUESTRA BIBLIOTECA CREADA PARA EL CORRECTO Y REAL DISEÑO DE NUESTRO PROTOTIPO.

- CARPETA CON EL CÓDIGO FUENTE DEL PROGRAMA DISEÑADO EN MPLAB ASI COMO TODOS LOS ARCHIVOS EMPLEADOS PARA EL FUNCIONAMIENTO Y COMPILACIÓN DEL PROGRAMA.

- CARPETA CON ARCHIVOS PDF CON LOS DATASHEET DE TODOS LOS COMPONENTES EMPLEADOS EN LA TARJETA.

- CARPETA CON TODOS LOS PLANOS EN FORMATO PDF Y LOS ARCHIVOS (.DXF) PARA EL DISEÑO DE LOS MISMOS:
  - PLANO 1: ESQUEMA ELÉCTRICO
  - PLANO 2: DIMENSIONES
  - PLANO 3: RUTADO CAPA COMPONENTES
  - PLANO 4: RUTADO CAPA SOLDADURA
  - PLANO 5: TALADROS
  - PLANO 6: CAPA SERIGRAFÍA SUPERIOR
  - PLANO 7: CAPA SERIGRAFÍA INFERIOR
  - PLANO 8: CAPA MÁSCARA SOLDADURA SUPERIOR
  - PLANO 9: CAPA MÁSCARA SOLDADURA INFERIOR

- CARPETA CON MANUALES DE AYUDA DE MICROSIM, USB, CAN, PIC32 Y COMPILADOR C30.

- DRIVERS DEL USB.

I.- ÍNDICE DE FIGURAS

II.- ÍNDICE DE TABLAS