Sistemas Informáticos

Curso 2009/2010

SIMULADOR DE ELEMENTOS DE CAMPO PARA EL CONTROL DE TRÁFICO FERROVIARIO

Daniel Galán Vicente
Francisco González Arribas

Dirigido por:
Prof. Jesús Manuel de la Cruz García
Dpto. Arquitectura de Computadores y Automática

Facultad de Informática
Universidad Complutense de Madrid

Agradecimientos

Un trabajo como este no habría sido posible sin la cooperación de mucha gente. En primer lugar queremos agradecérselo a nuestras familias que tanto han confiado y nos han dado. A nuestro tutor Jesús Manuel por conseguir llevar adelante este proyecto. Al Departamento de Automática, Ingeniería Electrónica e Informática Industrial que nos ha becado para su realización. A Javier Rainer y Ángel Luís Martínez que nos han ayudado a conocer el estado inicial del simulador SIMENC. En THALES, a David Sánchez, nuestro jefe de proyecto, a José Carlos, Julio y Javier, por hacernos sus compañeros.

Se autoriza a la Universidad Complutense a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Daniel Galán Vicente *Francisco González Arribas*

Resumen

En el presente proyecto se desarrolla una aplicación informática de simulación de elementos de campo (agujas, señales, pasos a nivel, etc.) para un sistema de control de tráfico ferroviario. Los requisitos de seguridad del software de control, en este sector, obligan a la realización de pruebas de verificación muy complejas, además, la implantación de la *alta velocidad* en España y otros países ha provocado la necesidad de rediseñar los "enclavamientos electrónicos" convencionales.

La aplicación IFBSIMCAM diseñada forma parte de una arquitectura software desarrollada mediante un acuerdo de colaboración entre la Universidad Politécnica de Madrid y la empresa THALES S.A. en el que han participado los alumnos autores del proyecto.

Las características fundamentales del trabajo vienen marcadas por el diseño software de simulación en un entorno de tiempo real complejo y las necesidades cambiantes del usuario final.

La aplicación simula el comportamiento de tarjetas controladoras de elementos de campo. Las tarjetas son de diseño específico de THALES ALEMANIA. Las tarjetas interactúan por un lado con el campo y por otro con el sistema de control mediante un protocolo específico sobre conexión CAN BUS. Esta filosofía viene marcada por las empresas del sector ferroviario como ADIF, RENFE, SCNF, etc.

El usuario puede simular cualquier escenario, incluyendo situaciones de error en comunicaciones, anomalías de las señales de campo, retardos en el envío de señales, etc., pudiendo garantizar que el software de control responde correctamente frente a cualquier situación posible. El proyecto supone para la empresa un importante ahorro económico dado que afecta en varios conceptos:

- Reduce el tiempo de verificación al poder realizar las pruebas en el propio centro de desarrollo y no en las estaciones de ferrocarril.

- Reduce el coste de los equipos de verificación que quedan reducidos al hardware y software de simulación, no siendo necesario disponer de lámparas y motores reales como se venía haciendo hasta ahora.

- Reduce el espacio físico de los sistemas de verificación ya que un armario con racks de 19 pulgadas es suficiente para simular la estación más compleja.

Palabras clave: Ingeniería del Software, Control de tráfico ferroviario, Sistemas de Tiempo Real, Sistemas Distribuidos, Protocolo CANBUS, Verificación de Software.

Summary

In the present project it is develop a simulation software application of field elements (needles, signals, level crossings, etc.) for a rail traffic control system. The security requirements for control software, in this sector, require the use of complex verification tests, moreover, the introduction of high speed in Spain and other countries has led to the need to redesign the conventional "electronic interlocking".

The design of the IFBSIMCAM application it is part of a software architecture developed through a collaborative agreement between the Universidad Politécnica de Madrid and the company THALES SA in which students have participated authors of the project.

The key features of the project are marked by the design of simulation software in a complex real time environment and the changing needs of end users.

The application simulates the behavior of controller cards of field elements. The design of the cards is specific from THALES GERMANY. Cards on the one hand interact with the field and the other with the control system through a specific CANBUS protocol connection. This philosophy is marked by the railway companies as ADIF, RENFE, SCNF, etc.

The user can simulate any scenario, including communications error conditions, abnormalities of the field signals, delays in sending signals, etc, so you can ensure that the control software responds correctly to any situation possible. The project calls for the company significant financial savings as it affects several concepts:

- It reduces the time to check the test in the own center of development and not on railway stations.

- It reduces the cost of verification teams, which are reduced to the hardware and simulation software, being not unrequired real lights and motors as was done so far.

- It reduces the physical space of the verification systems because a cabinet with racks of 19 inches is sufficient to simulate the most complex station.

Key words: Software Engineering, Rail Traffic Control, Real Time Systems, Distributed Systems, CANBUS Protocol, Software Verification.

Índice

1. Introducción

El presente proyecto se enmarca dentro de un acuerdo de colaboración entre el Departamento de Automática, Ingeniería Electrónica e Informática Industrial de la Universidad Politécnica de Madrid y la empresa THALES S.A. Desde el año 2000 se vienen desarrollando diferentes herramientas de simulación para la validación del software que esta empresa incorpora en sus enclavamientos electrónicos.

Desde enero de 2009 los alumnos, autores de este proyecto fin de carrera, participan en el desarrollo de un nuevo simulador para la empresa THALES S.A. como becarios. Se trata, pues, de un proyecto real de transferencia de tecnología entre Universidad y Empresa, en el que los conocimientos de programación de sistemas de tiempo real y de ingeniería software han sido fundamentales para su desarrollo y al mismo tiempo participar en un grupo de I+D+i trabajando en la propia empresa ha sido una experiencia muy valiosa.

Los enclavamientos electrónicos tradicionalmente basados en relés en su interacción con los elementos de señalización, cambio de agujas, pasos a nivel, etc. han sufrido un cambio radical con la incorporación de la alta velocidad. La empresa THALES S.A. ha apostado por la utilización de buses tipo CAN BUS para conectar estos elementos a sus enclavamientos. Los enclavamientos permiten la gestión del tráfico ferroviario en una estación y sus alrededores con la máxima seguridad.

La seguridad hace que se utilice redundancia 2 de 3 en los sistemas de control y que el proceso de verificación del software y las pruebas de aceptación por parte del gestor ferroviario (nacional o extranjero) sean muy exigentes.

Antes de su instalación en una estación determinada se realizan todo tipo de pruebas para verificar que el comportamiento del software desarrollado y su configuración específica para la estación reacciona de forma segura. THALES S.A. había solicitado a la Universidad simuladores que tuvieran el mismo comportamiento electrónico que los elementos de campo y a la vez forzar comportamientos anómalos: lámparas fundidas, cortocircuitos, motores bloqueados, falta de reacción frente a mandos, etc.

En las imágenes siguientes se muestran algunos de los sistemas desarrollados.






La inclusión de un bus de campo para la gestión supone, desde el punto de vista de la verificación del software, una complejidad adicional porque se necesitan verificar todas las posibles situaciones derivadas de fallos en el intercambio de mensajes.

Objetivos

Con este propósito se proponen los siguientes objetivos del proyecto:

1. Estudiar los fundamentos de la señalización ferroviaria
2. Estudiar las características funcionales de los enclavamientos desarrollados por la empresa THALES
3. Estudiar los simuladores SIMENC y SIMCAM desarrollados previamente
4. Diseñar un nuevo simulador que permita la verificación del intercambio de mensajes con protocolo BUS CAN entre el enclavamiento y el campo. La empresa aportará los requisitos de diseño y las pruebas que deban realizarse.
5. Validación del simulador
6. Documentación de acuerdo con el estándar de THALES

La duración estimada del proyecto es de 18 meses.

2. Requisitos del sistema

El Capítulo 2 está dedicado a recoger los requisitos marcados por Thales S.A. Se han clasificado en seis apartados en función de sus características: Generales, vista gráfica, mensajes, filtro de trazas, simulación e interfaces.

2.1. Requisitos generales

- R1.1: La aplicación estará dividida en cuatro partes diferenciadas: la vista gráfica de IFBs, el Espía de mensajes, las Propiedades de simulación y el Filtro de trazas.

- R1.2: La vista gráfica de IFBs mostrará la monitorización de las tarjetas y para cada una de ellas sus canales A/B, los comandos y las indicaciones.

- R1.3: El Espía de mensajes permitirá controlar los mensajes que siguen el protocolo FEC-IFBs y variar el número de mensajes monitorizados.

- R1.4: En Propiedades de la simulación se mostrará la conexión de EC's y/o Simenc, permitiendo aplicar un filtro a los mensajes y cerrar la recepción de información desde los Canales A/B del Can Bus.

- R1.5: Se permitirá ejecutar acciones sobre las tarjetas tales como enviar warnings, enviar LIFEACK o apagarlas.

- R1.6: El Filtro de trazas mostrará las trazas de los mensajes que hayan sido filtrados.

2.2. Requisitos de la vista gráfica de IFBs

- R2.1: En la vista gráfica de IFBs, las tarjetas se crearán en la posición adecuada de acuerdo con su armario, su rack y su subrack, es decir, en función de su posición real.

- R2.2: En la vista gráfica de IFBs, se representará mediante leds simulados las acciones de las tarjetas (si están conectadas, si están apagadas, si sus comandos e indicaciones están en on/off)

- R2.3: En la vista gráfica de IFBs, el usuario puede seleccionar cualquiera de las tarjetas representadas y esto determinará qué IFB será simulada y qué trazas se visualizarán en el Espía de mensajes.

- R2.4: En Propiedades de la simulación, el usuario podrá activar o desactivar los canales A/B y, en consecuencia, el envío y la recepción de mensajes de todas las tarjetas del proyecto. Por defecto, estos valores estarán activos.

- R2.5: Las EC's representarán, mediante un código de colores, el estado del conector DB9.

- R2.6: La representación de las IFBs para los elementos de tipo EIO debe:
    - Mostrar qué canales están activos.
    - 8 leds simulados. Representarán 4 comandos dobles que recibirá del EC.
    - 12 leds simulados, que representan 12 indicaciones.

- R2.7: La representación de las IFBs para los elementos de tipo SEC debe:
    - Mostrar los canales activos.
    - 2 leds simulados. Representarán un comando que recibirá del EC.
    - 16 leds simulados, que representan 8 filamentos principales y 8 filamentos dobles si existen.

- R2.8: La representación de las IFBs para los elementos de tipo PT debe:
    - Mostrar qué canales están activos.
    - 2 leds simulados. Representarán un comando que recibirá del EC.

- 4 leds simulados, que representan el estado de las indicaciones para este tipo de elemento.

- R2.9: Las indicaciones de los datos recibidos desde la aplicación SimEnc, serán representados correctamente en los LED's indicadores del IFB.

2.3. Requisitos del espía de mensajes

- R3.1: Los mensajes recibidos desde los EC's serán automáticamente almacenados en un buffer interno con una capacidad para 200 mensajes.

- R3.2: Si un mensaje corresponde con la selección del usuario de la vista gráfica será automáticamente mostrado en la ventana de vista de mensajes.

- R3.3: Existirá una interfaz con opción para ver y registrar el envío de mensajes Lifeack y Lifetick. Por defecto, estará desactivada.

- R3.4: Se podrá registrar en un archivo txt el log de todas las trazas entrantes y salientes de la tarjeta CAN localizada en el IFBSimCam, o tan sólo el log de aquellas que estén siendo filtradas

- R3.5: El formato de los mensajes registrados identificará los siguientes campos:
  - Enviado/Recibido
  - Dirección de origen
  - Dirección de destino
  - Byte_code (en referencia al protocolo utilizado)
  - Traducción (en referencia al protocolo utilizado)
  - Número de secuencia (en referencia al protocolo utilizado)
  - Número iniciador de secuencia (en referencia al protocolo utilizado)
  - Fecha

- R3.6: Las trazas de los mensajes tendrán el siguiente formato:

- CABECERA:
  - **rack:** El rack en el que la IFB se encuentra en el armario.
  - **Slot:** La ranura que ocupa la IFB en el rack dado del armario.
  - **channel:** 'A' o ' B'
  - **direction:** "TX" para los mensajes trasmitidos del simulador al EC y "RX" para los mensajes recibidos en el simulador provenientes de un EC.
  - **seconds:** Segundos del time stamp.
  - **milliseconds:** Milisegundos del time stamp
  - **source_addres:** La dirección XFibuss de la fuente de la conexión en formato hexadecimal.
  - **target_address:** La dirección XFibuss del destino de la conexión en formato hexadecimal.
  - **priority:** Prioridad del mensaje en XFibuss. Debe ser 1 para esta aplicación
  - **port:** Puerto del mensaje en XFibuss. Debe ser 0 para esta aplicación.

- CONTENIDO:
  - **num_bytes:** El número de bytes en formato decimal que tiene el cuerpo del mensaje XFibuss.
  - **$byte_i$:** Valor hexadecimal del byte número "i" de la traducción(número de secuencia)[indicaciones] del mensaje XFibuss
  - **translation**: Un string contendrá el significado del mensaje XFibuss en el protocolo. El string puede contener: CONNECT, CONNACK, USRCMD2, USRACK2, USRIND, LIFETICK, LIFEACK para los mensajes conocidos y para los desconocidos: UNKNOWN.
  - **sequence_number:** Dos bytes con el formato: ($0xbyte_1byte_2$)

2.4. Requisitos del filtro de trazas

- R4.1: En el filtro de trazas, podremos elegir si queremos ver los mensajes correspondientes a todas las IFB's, a una en concreto o no ver ningún mensaje; y también los lifeacks y los mensajes provenientes del broadcast.

- R4.2: En el filtro de trazas, podremos filtrar aquellos mensajes que cumplan unos valores concretos en sus bits.

2.5. Requisitos de Propiedades de la simulación

- R5.1: Permitirá simular el fallo físico de una tarjeta determinada y ésta se parará, rechazando todos los mensajes entrantes, no enviando más mensajes, no estando disponible y actualizando los correspondientes LED's en la vista gráfica de IFB's.

- R5.2: Permitirá reiniciar un canal y en la vista gráfica se simulará la recuperación de este fallo físico.

- R5.3: Permitirá simular warnings de potencia y de temperatura en una tarjeta. En consecuencia, la IFB alertará su estado y mandará los mensajes adecuados al EC.

- R5.4: Permitirá cambiar la dirección de una tarjeta determinada variando el valor de las direcciones de sus canales A y B.

- R5.5: Permitirá configurar aspectos del protocolo, filtrado de mensajes y el modo de lifetick.

- R5.6: Un componente gráfico mostrará el estado de conexión de los EC's y SimEnc.

2.6. Especificación de Interfaces

Este punto del proyecto explica cómo serán los mensajes que se transmitirán entre las diferentes interfaces que se engloban en el proyecto, así como una breve descripción de la estructura de éstas.

Este apartado corresponde a un documento suministrado por la empresa, del que se han tomado los requisitos que afectan al programa que se quiere diseñar. Se ha mantenido el texto original en inglés.

2.6.1. IFBSimCam – SimEnc interface (I3)

The communication between IFBSimCam and SimEnc is a TCP/IP socket based communication. A specific protocol was defined for this operation.

Messages from IFBSIMCAM:

IFB_SetEIOCommand:

// msg[1,2] contains address of element to set

// msg[3] contains command number

// msg[4] contains command value Channel A

// msg[5] contains command value Channel B

IFB_RequestEIOStatus:

// msg[1,2] contains address of element to set

// msg[3] contains command number

// msg[4] contains command value Channel A

// msg[5] contains command value Channel B

// msg[6] contains command type (1,2) CMD or Status

IFB_SetLampCommand:

// msg[1,2] contains address of lamp element

// msg[3] contains lamp index

// msg[4] contains 0x01 for blinking or 0x00 for not blinking

IFB_SetDayNight:

// msg[1,2] contains address of lamp element

IFB_LampInfoRequest:

// msg[2] contains address of lamp whose info is requested

// msg[3] contains the filament number

// msg[4] contains 0 (OFF) 1 (BLINKING) 2(ON)

IFB_SetCurrentValues:

// msg[1,2] contains address of lamp element

// msg[3] contains lamp index

// msg[4] contains current set number

// msg[5] contains max current off

// msg[6] contains min-night-on current value

// msg[7] contains max-night-on current value

// msg[8] contains min-day-on current value

// msg[9] contains max-day-on current value

// msg[10]*256+msg[11] contain current measure delay in ms

IFB_SetDoubleFilament:

// msg[1,2] contains address of lamp element

// msg[3] contains lamp index

// msg[4] contains 0x00 for single filament or 0x01 for double filament

IFB_SetVoltageValues:

// msg[1,2] contains address of lamp element

// msg[3] contains lamp index

// msg[4] contains min-night-on voltage value

// msg[5] contains max-night-on voltage value

// msg[6] contains min-day-on voltage value

// msg[7] contains max-day-on voltage value

IFB_SetBlinkingPermission:

// msg[1,2] contains address of lamp element

// msg[3] contains lamp index

// msg[4] contains 0x00 if lamp is not enabled to blink, 0x01 it is enabled to blink

// msg[5]*256+msg[6] contain blinking period value. It only applies if msg[3]==0x01

// msg[7]*256+msg[8] contain blinking ON time. It only applies if msg[3]==0x01.

IFB_ThrowPoint

// msg[1] & [2] = The first contact

IFB_ConfigPTtime

// msg[1,2] contains address of point element

// msg[3] contains minimum point machine moving time

Messages from SIMENC:

IFB_ConfigEIO:

msg[0]= IFB_ConfigEIO;

msg[1]= 4;

msg[2] & [3] = The first contact

msg[4]= Commands Count

msg[5]= Indications Count

msg[6]= contains protocol ID

msg[7]= contains application ID

msg[8]= contains address A.

msg[9]= contains address B

IFB_ConfigSEC:

msg[0]= IFB_ ConfigSEC;

msg[1]= 5;

msg[2] & [3] = The first contact

msg[4]= Commands Count

msg[5]= Indications Count

msg[6]= contains protocol ID

msg[7]= contains application ID

msg[8]= contains address A.

msg[9]= contains address B

case IFB_SetEIOIndication:

msg[0]= IFB_SetEIOIndication;

msg[1] & [2] = The first contact

msg[3]= Not used

msg[4]= Indications Count

msg[5…] = indications

case IFB_RequestEIOStatus:

msg[0]= IFB_SetEIOIndication;

msg[1] & [2] = The first contact

case IFB_SetLampIndications:

msg[0]= IFB_ SetLampIndications;

msg[1] & [2] = The first contact

msg[3] contains lamp index

msg[4]= contains 0x00 for day or 0x01 for night

msg[5] contains lamp status, for which the following codes are defined:

// 0x00: off

// 0x01: fused

// 0x02: on

// 0x03: blink

msg[6] contains diagnosis code, for which the following values are defined:

// 0x00: no failure

// 0x01: main filament fused

// 0x02: auxiliary filament fused

// 0x03: Voltage measurement 3 times mismatch with NBS

// 0x04: Current measurement 3 times mismatch with NBS

// 0x05: Short circuit

// 0x06: Current in lamp off-state

// 0x07: Degrade blinking

// 0x08: Out of measurement window

// 0x09: Overvoltage

// 0x0a: cross check

case IFB_PreLampIndications:

msg[0]= IFB_PreLampIndications;

msg[1] & [2] = The first contact

msg[3] contains lamp index

msg[4]= contains 0x00 for day or 0x01 for night

msg[5] contains lamp status, for which the following codes are defined:

// 0x00: off

// 0x01: fused

// 0x02: on

// 0x03: blink

msg[6] contains diagnosis code, for which the following values are defined:

// 0x00: no failure

// 0x01: main filament fused

// 0x02: auxiliary filament fused

// 0x03: Voltage measurement 3 times mismatch with NBS

// 0x04: Current measurement 3 times mismatch with NBS

// 0x05: Short circuit

// 0x06: Current in lamp off-state

// 0x07: Degrade blinking

// 0x08: Out of measurement window

// 0x09: Overvoltage

// 0x0a: cross check

IFB_SetPointIndication

msg[1] & [2] = The first contact

msg[3]= 0

msg[4]= Indications Count

msg[5]…= The indication values

IFB_ConfigPT

msg[1]= 3;

msg[2] & [3] = The first contact

msg[4]= Commands Count

msg[5]= Indications Count

msg[6]= contains protocol ID

msg[7]= contains application ID

msg[8]= contains address A.

msg[9]= contains address B

The class CSocketNode was defined to be a common support for all applications.

2.6.1.1. CSocketNode messages

Class CSocketNode in IFBSimCam encapsulates the sockets library, transparently manages all connection and disconnections. It is message oriented in the sense that data is sent and received in user packets, so the user does not have to separate messages. It is an "alive" class, using an internal thread to communicate without blocking the thread that uses this class because of socket blocking calls, and minimizing processor load and delays due to communications.

Internally, this class packs user messages as follows:

User Message:

Vector of bytes, not necessary understood as an ASCII string. It can be used to transfer binary data.

Length of vector:	n
Data:	DataByte0, DataByte1, ……, DataByte(n-1)

CSocketNode message:

Vector of bytes, not necessary understood as an ASCII string. It can be used to transfer binary data.

Length of vector:	n+2
Data:	HeaderByte0, HeaderByte1, LengthByte0, LengthByte1, DataByte0, DataByte1, ……, DataByte(n-1)

Header: Is a short integer (2 bytes) with fixed value used to identify the beginning of a message and to reject possible incoming messages not belonging to the used protocol.

Length: Is a short integer (2 bytes) with the size in bytes of the user data. This field is used to let the class wait and identify when a complete user message has arrived. The maximum user message size is limited then to 65535 bytes.

When CSocketNode class receives bytes from the communication partner, it stores them in an internal buffer until a complete message has been received. Then the OnMsg() callback is called, passing the whole message to the user function that must interpret the message.

Then at the reception procedure, 2 sockets with reading functions are called. The first one waiting for 4 bytes, the header and the length. Once a correct header has been received, a second read function is executing, waiting for a message of "length" bytes.

2.6.2. IFBSimCam – EC interface

The overview of this interface understood as just the Can Bus interface is shown in next figure:

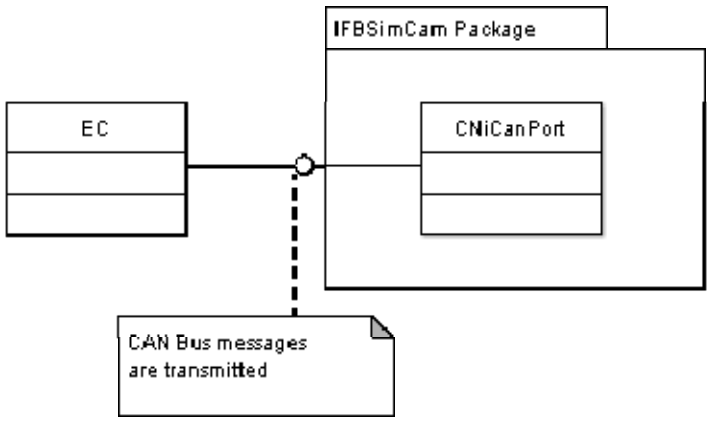


FIGURE: IFBSIMCAM - EC INTERFACE

But several classes are involved in the generation of Can Bus frames transmitted to ECs. These classes are represented in the following diagram:

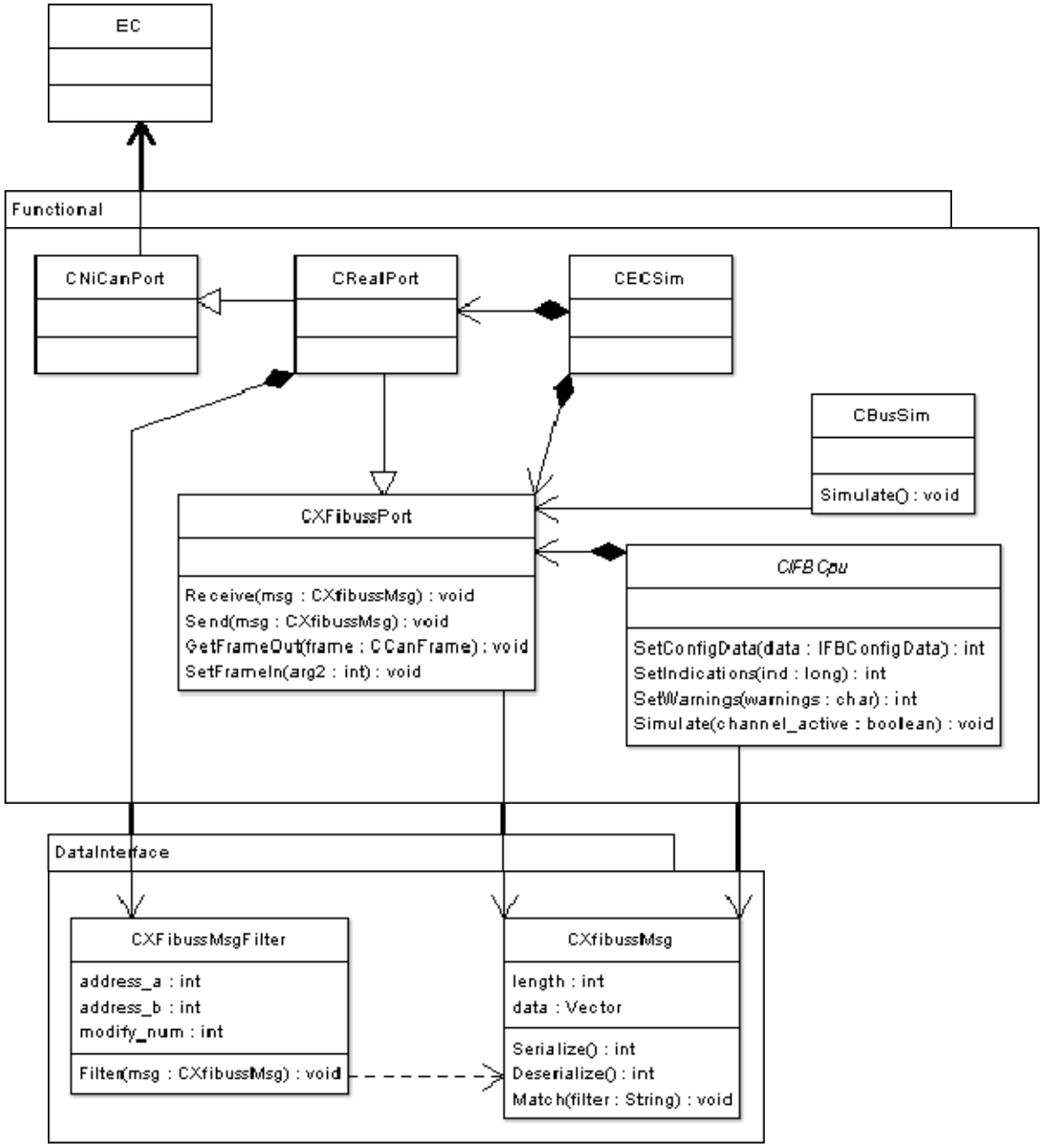


FIGURE: CAN BUS FRAME GENERATION CLASSES (1)

The way these classes are instantiated and connected can be represented in the following collaboration diagram:

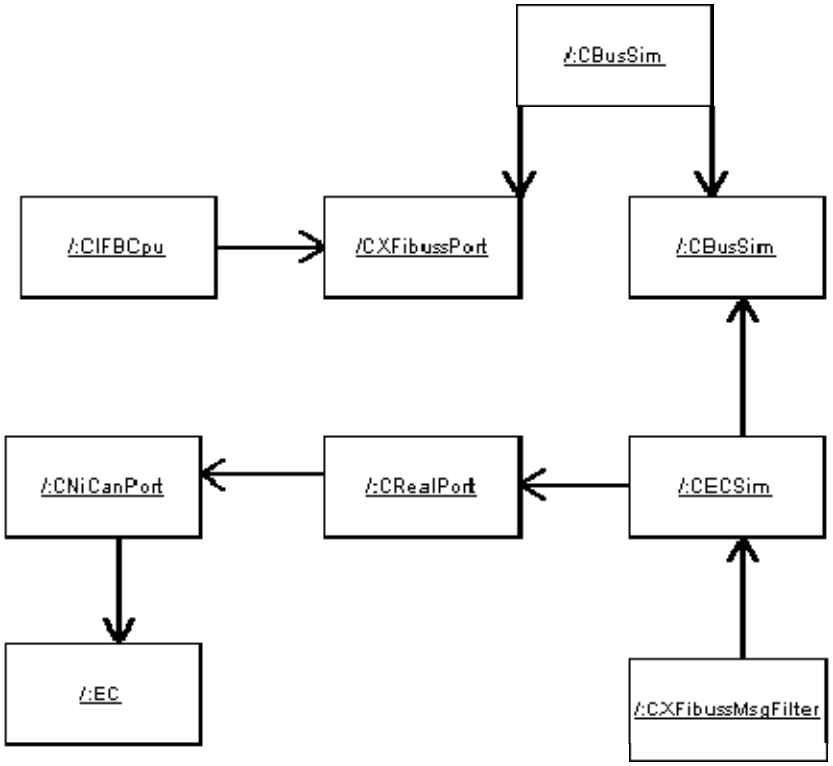


FIGURE: CAN BUS FRAME GENERATION CLASSES (2)

Although can bus frames are transmitted and received by CNiCanPort from/to EC through the real Can Bus, the messages are really routed from/to CIFBCpu objects. CIFBCpu objects are the objects where information is actually processed.

CIFBCpu processes and generates XFibuss messages that are converted to can frames through the CXFibussPort class. These messages are described in the following sections:

2.6.2.1. XFibuss protocol

The XFIBUSS protocol behaves like the underlying CAN-Protocol with semantic enhancements:

- XFIBUSS allows datagram oriented data transfer in a not connected way.
- XFIBUSS benefits from the CAN inherent object model approach, broadcast with the meaning of unaddressed services and acceptance filtering for these services is supported.
- The data transfer between XFIBUSS nodes is not reliable, no services for retransmission of messages are defined, and messages may be lost.

Main differences to CAN

- Message size may exceed 8 Bytes.
- CAN-IDs have predefined semantics. XFIBUSS provides a host address and port number abstraction, the priority relevant part of the CAN-ID is reduced to 8 classes

The transformation that XFibuss does to Can Frames, see **[CBus]**, is represented in the following figure, where:

APPL (application layer)
DTI (Data Integrity Layer)
TRS (Transmission layer)
NET (network layer)
PRIO: Xfibuss message priority
SRC: Xfibuss source address
DEST: Xfibuss target address
PORT: port
LEN: Length of Data in bytes.

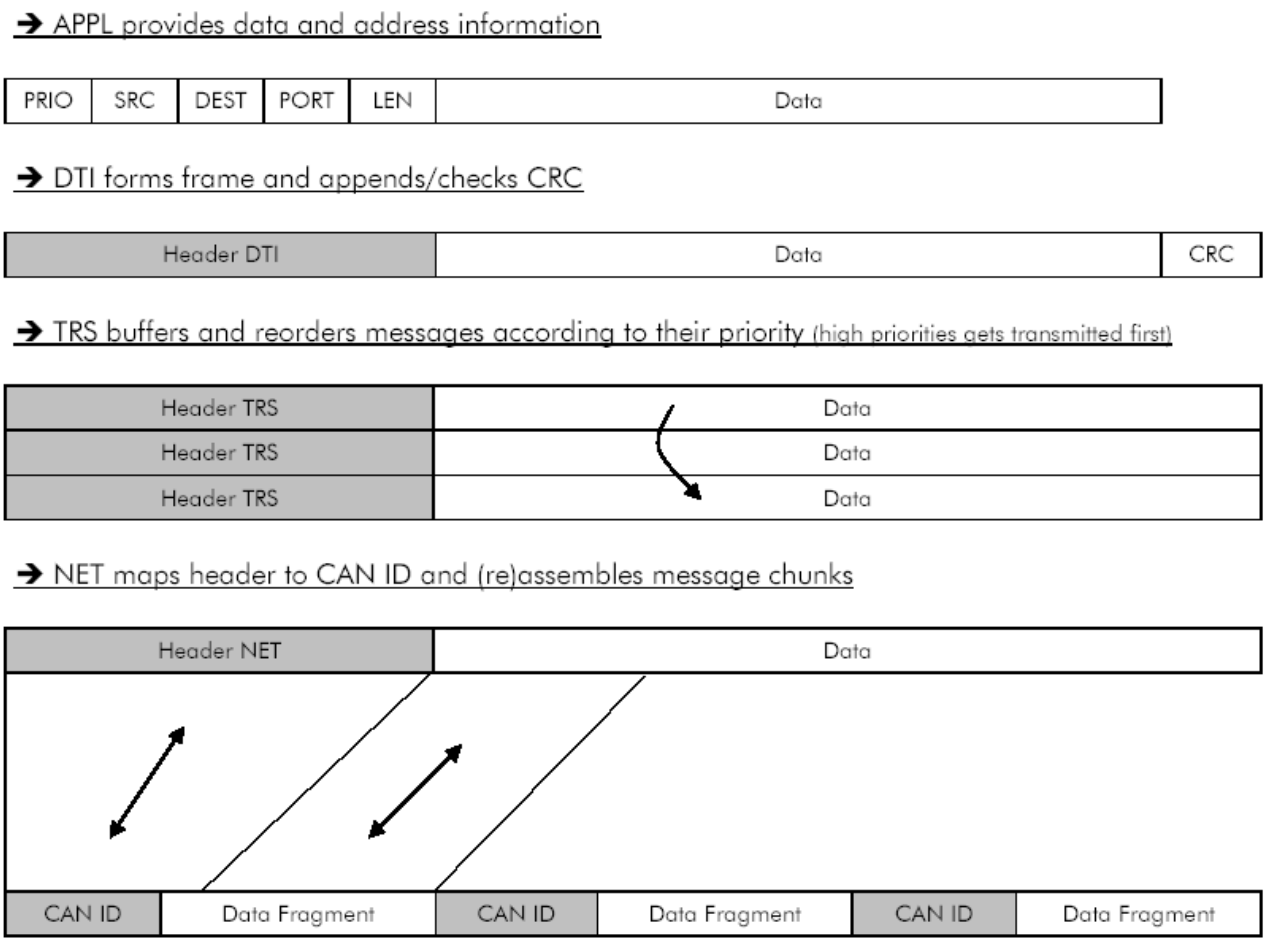


FIGURE: XFIBUSS PROTOCOL

2.6.2.1.1. CXfibussPort class

This class implements the XFibuss protocol, converting XFibuss messages to can frames, and can frames to XFibuss messages. It can also simulate the physical disconnection of the cable connected to that port. It is intended to be instanced by any processor that requires a conversion between these two communication layers.

2.6.2.1.2. Class CNiCanPort

Encapsulates the NI Can Card library calls for a Can Bus channel. It is the connection point the real Can Bus and the application. This class is able to send and receive CCanFrames can frames objects through the real can bus card port.

2.6.2.1.3. CRealPort class

This class inherits both from CXfibussPort, thus providing the functionality of can frames – XFibuss translation, and CNiCanPort, thus providing the capability of real communication to an EC. This class is the responsible of modifying outgoing (simulated IFBs -> real ECs) XFibuss messages as defined and established by the user, thus containing a CXfibussMsgFilter object for that purpose. This class is also the responsible of logging XFibuss message traces to be sent to SimEnc.

2.6.2.2. XFibuss Messages from IFB Simulator to EC

Messages are described in **[CBUS]** protocol can bus document.

2.6.3. SimEnc -SimCam interface (I4)

The communication between SimEnc and SimCam is a TCP/IP socket based communication. A specific protocol was defined for these operations.

Messages from SIMCAM:

FSIM_Mode:

Msg[1] // Mode

FSIM_CommandValue:

Msg[1],[2] element number

Msg[3…] commands

FSIM_ReturnCardValues:

Msg[1] Card

Msg[2] Value

FSIM_IndicationValuesAck:

Msg[1] 0: OK 1 Wrong type

Msg[2][3] Element number

Msg[4] Error

Messages from SIMENC:

SIM_SetMode

Msg [2] 0: //Restart 1: //Load Data 2: //Simulation3: //Close

SIM_IndicationValues

Msg[1,2] Element Number.

MSg[3] Element Type

Msg[4] Number of Indications

Msg[5 ...] New values of Indications

SIM_DefineElement

Msg[1,2] The element Number

Msg [3] Card Addres

Msg[4] First bit Position

Msg[5] Number of Indications

Msg[6] Number of Commands

Msg[7] Card Type

SIM_TEST_SetValues

Msg[1] Card

Msg[2] Value Low

Msg[3] Value High

SIM_TEST_GetValues

Msg[1] Card

SIM_SignalValues

Msg[1,2] Element Number.

Msg[3] Element Type

Msg[4] Number of Indications

Msg[5 ...] New values of Indications

// For normal SECs

Msg[5] 8 signals (bit to 1 Normal bit to 0 Fused)

Msg[6] 4 signals

Msg[7] 8 signals (bit to 1 ShortCircuit)

Msg[8] 4 signals

// For continuous range control

Msg[5] 8 signals (bit to 1 filament selected (only one))

Msg[6] Not used

Msg[7] mA Value

Msg[8] Not used

The class CSocketNode is used to be a common support for all placations as sockect based communication.

3. Señalización Ferroviaria

3.1. Señalización Ferroviaria. Introducción

3.1.1. Definición, objetivo y alcance de la señalización ferroviaria

La red ferroviaria se define a lo largo de las vías que la integran, en la que existen instalaciones que permiten la circulación de los trenes por esta red. La señalización define el funcionamiento de cada uno de los elementos que se encuentran en las instalaciones ferroviarias y que pueden estar en las estaciones, en los trayectos, en los trenes y en puestos de centralización.

El objetivo de este capítulo es una introducción a cada uno de los elementos que intervienen en la señalización.

3.2. Clasificación de las líneas férreas

Todas las líneas que integran la red ferroviaria no son de la misma naturaleza, independientemente de las instalaciones que la integran. La clasificación de las líneas puede hacerse atendiendo a criterios diversos como la velocidad de circulación, la densidad del tráfico y, dentro de éstos, atendiendo a la máxima masa por eje y la máxima masa por metro lineal de los vehículos que van a circular por ellas.

3.2.1. Aproximación a los elementos que integran una línea férrea

La tarea de gestión del tráfico ferroviario se realiza desde el puesto central de circulación (PCC). Al PCC llega toda la información necesaria de las instalaciones existentes en la vía para poder realizar las operaciones de regulación del tráfico por la línea.

Una línea ferroviaria queda definida por un conjunto de estaciones. En las estaciones existen una serie de instalaciones de seguridad que garantizan el tránsito de una circulación dentro del dominio de la misma.

Las instalaciones existentes son las siguientes:

- *Equipos de detección de tren completo.*

  Los equipos de detección de tren completo son elementos que determinan la presencia del tren y que permiten determinar si la sección de vía férrea que supervisan está libre u ocupada.

  La sección supervisada por el elemento de detección la definimos como **circuito de vía** o **cantón**.

- *Enclavamiento.*

  La estación es el enclave ferroviario que se habilita a fin de permitir movimientos de circulaciones por la misma. De este modo, el enclavamiento es el sistema que se define para asegurar la circulación de un tren por la estación, estableciendo y supervisando un determinado movimiento.

  El interfaz hombre-máquina hacia el operador se denomina cuadro de mandos o, en general, puesto de operación local (POL). Desde él se realizan las órdenes para solicitar movimientos y en él se indica la situación actual de cada elemento del campo.

  Los elementos del campo que se gobiernan desde este sistema son las **señales ferroviarias** de la estación, que son las que autorizan al maquinista proseguir la marcha del tren; las **agujas**, que son los elementos que permiten encaminar el tren en su tránsito por la estación; los **pasos a nivel** que se ven afectados por la estación, es decir, que se ven afectados por el movimiento de trenes dentro de la misma.

  Los elementos de campo (señales, agujas, circuitos de vía y pasos a nivel) se visualizan en el POL. De este modo, el gestor de la estación es capaz de conocer, en todo momento, el estado en que se encuentran los elementos mencionados y, por consiguiente, el estado de la estación en todo instante. También se visualiza todo aspecto relacionado con el establecimiento de un itinerario y del avance de la circulación hasta completarlo.

  Para poder autorizar un movimiento es necesario satisfacer las condiciones impuestas por la **consigna de explotación** de la estación. En esta consigna se establece, para cada movimiento, las secciones que deben estar libres, la posición de las agujas y, además, se define el aspecto que puede dar cada una de las señales ferroviarias para permitir la marcha del tren.

  Además, se define en dicha consigna el **cuadro de incompatibilidades de movimientos**. En este cuadro se indica si un movimiento, que se desea establecer en la estación, es compatible con los existentes en ese instante.

- *Equipo de bloqueo.*

Este sistema asegura la circulación de los trenes entre dos estaciones colaterales. Obedece órdenes del enclavamiento y transfiere indicaciones al mismo.

Es tarea de esta instalación el gobierno de las señales ferroviarias que existan en el trayecto entre las estaciones colaterales y autorizar la expedición del tren entre ellas.

- *El puesto satélite.*

  El PCC recibirá la información de las estaciones y del trayecto entre estaciones a través del equipamiento de **puesto satélite**. Este equipo es el encargado de transferir las informaciones de la estación y del trayecto, que recibirá del estado de los circuitos de vía, del enclavamiento y del sistema de bloqueo; además transferirá al enclavamiento las órdenes dadas desde el PCC a fin de ejecutarlas en el mismo.

3.2.2. El centro de tráfico centralizado

De esta manera, el sistema de gestión ferroviaria es un sistema de centralización de los sistemas que se encuentran instalados en la línea con fines de realizar un **Control de Tráfico Centralizado** o C.T.C. Para facilitar la gestión del tráfico el CTC, además de tener una imagen con el estado actual de la línea centralizada más un interfaz para mando sobre los enclavamientos existentes, se realizan funciones de **seguimiento de trenes**.

Cada circulación que entra en la línea ferroviaria se identifica con un número de tren. La función de seguimiento de trenes consiste en la supervisión de la secuencia de ocupación del tren en movimiento sobre las distintas secciones de vía, desde la estación origen del tren hasta la estación de destino del mismo.

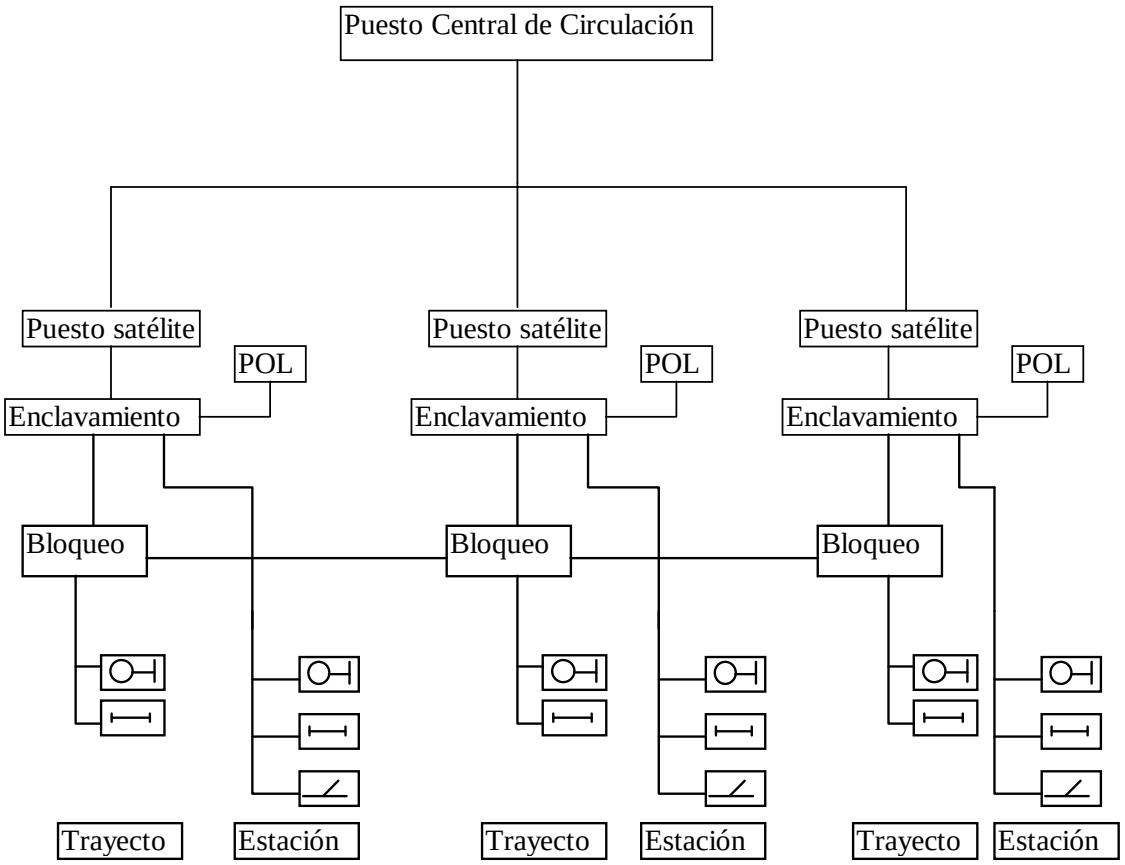
La asignación del número de tren se realiza de acuerdo a una **planificación de horarios**. Esta planificación es el horario que deben satisfacer los trenes en movimiento sobre la línea y es la guía para la explotación de la misma. En la siguiente figura se muestra la arquitectura de un sistema CTC. En la misma se identifica cada una de las instalaciones existentes y las conexiones entre las mismas.

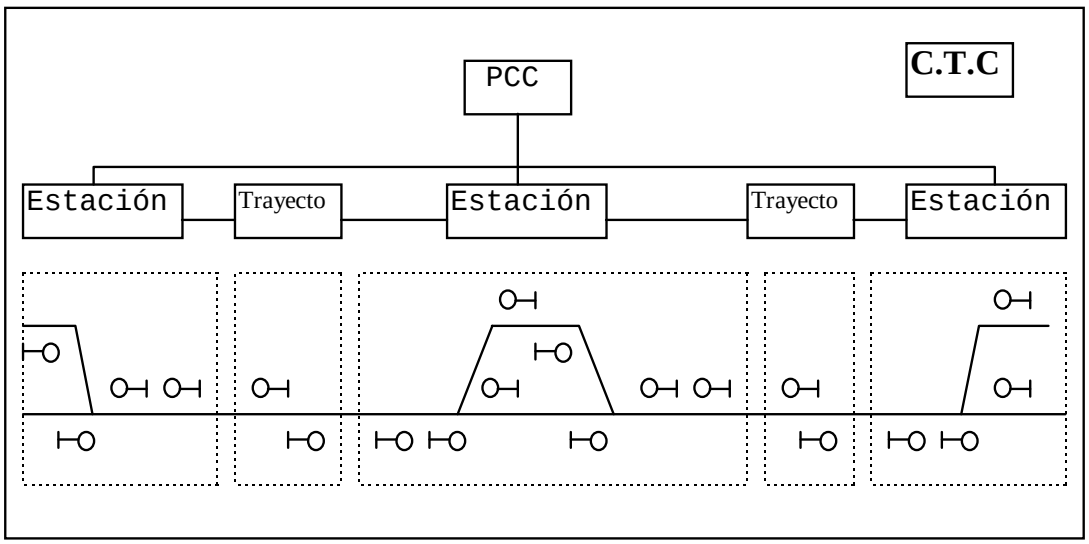
Los símbolos para identificar los elementos de la estación y que aparecen en la figura son los siguientes:

SEÑAL

AGUJA

CIRCUITO DE VÍA


Puesto Central de Circulación
Puesto satélite
POL
Enclavamiento
Bloqueo
Trayecto
Estación
Puesto satélite
POL
Enclavamiento
Bloqueo
Trayecto
Estación
Puesto satélite
POL
Enclavamiento
Bloqueo
Trayecto
Estación


C.T.C
PCC
Estación
Trayecto
Estación
Trayecto
Estación

3.2.3. Relación entre los sistemas y el tren

El tren en su avance encuentra las señales, que realizan una múltiple función:

- Como interfaz entre la vía (control) y los trenes (flujo a controlar).
- Como elemento regulador de la velocidad del tren en su avance por la vía.
- Como elemento de protección para el tren que la rebasa.

Efectivamente, el avance del tren se realiza conforme al aspecto que la señal ofrece al maquinista y que es fruto de la condición de seguridad de las instalaciones correspondientes al enclavamiento y al bloqueo. La cantidad de señales que se disponen en un trayecto va a depender de la capacidad de la línea y, por tanto, permitirá una secuencia de trenes en su recorrido.

Además, las señales se definen para proteger al tren que la rebasa. De esta manera el aspecto de una señal informa al tren de la proximidad de la circulación anterior. Se define así la sección o cantón como la fracción de vía que es protegida por una señal.

La sección de vía dispone de un elemento de supervisión de la misma y con el cual se permite ejecutar las funciones propias de la detección del tren y que servirán al resto de las instalaciones para definir la integridad del tren al paso de una a otra sección.

Por último, para poder localizar los trenes en la línea es necesario realizar una clasificación que los unifique en su recorrido. Cada una de las líneas que integran la red férrea tiene una cabecera y un final de línea. Los trenes recorren la línea de una a otra estación y, con ello, se define un concepto de paridad de los trenes; así, los trenes serán pares o impares según avancen desde la cabecera o se dirijan a la misma.

Las señales protegen la marcha de los trenes pares e impares, tanto en vía doble como en vía única. Este concepto, así expresado, se denomina banalización.

3.2.4. Definición de los movimientos de un tren en su recorrido

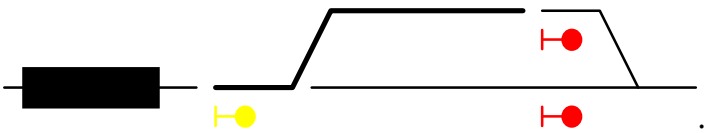
En este apartado se van a definir los movimientos que autorizan el avance de un tren desde la estación de origen a la estación de destino.

- *Salida del tren de una estación*

Es aquel movimiento que autoriza el avance de un tren, situado en una determinada vía de estacionamiento de la estación, hasta la señal de entrada de la estación colateral.

- *Entrada del tren a una estación*

  Es aquel movimiento que autoriza el avance de un tren desde la señal de entrada de la estación hasta una determinada vía de estacionamiento



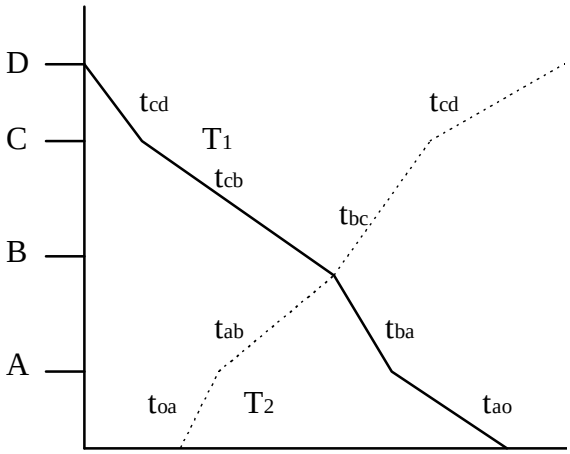
- *Paso directo de un tren por una estación*

  Es la suma de un movimiento de entrada a la estación más un movimiento de salida de la misma.

3.2.5. Representación del movimiento del tren

Para representar los movimientos del tren recorriendo la línea se define la **gráfica de espacio-tiempo**. En esta gráfica se mide en abscisas el tiempo planificado para el recorrido del tren; en el eje de ordenada se disponen los puntos de paso de los trenes.

Un ejemplo de esta gráfica se ilustra en la siguiente figura. En ella A, B, C y D representan los puntos kilométricos donde se localizan estas estaciones; t es el tiempo planificado. O es la estación origen.



En esta gráfica se marcan los recorridos de los trenes T_1 y T_2. El primero de ellos recorre línea en sentido descendente y el primero de ellos en sentido ascendente.

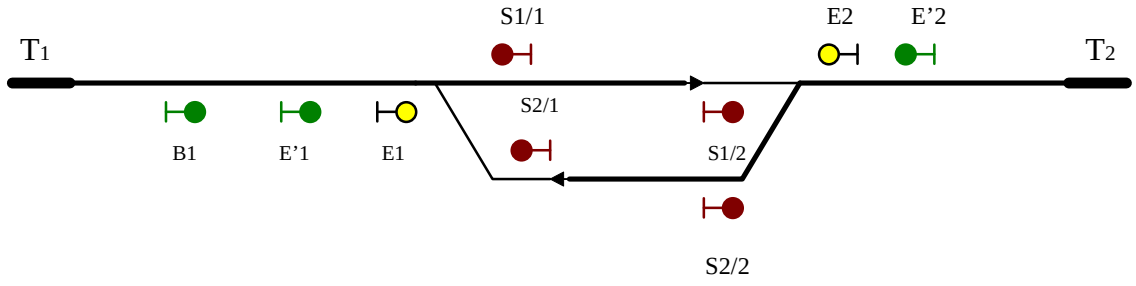
Además, cada segmento t_{ij} representa el tiempo desde que un tren sale de una estación hasta su estacionamiento en la siguiente.

La gráfica de espacio-tiempo se considera un parámetro de partida para el problema de la gestión del tráfico. Esta es la única herramienta existente para el seguimiento y cumplimiento de horarios de las circulaciones.

Todos los trenes no son de igual naturaleza. La naturaleza del tren va a depender de las características que se den en el planificador de horarios. La prioridad de un tren se define a priori y puede ser modificada a medida que el tren avanza en su recorrido, dependiendo este cambio de prioridad del progreso de la marcha del tren.

3.2.6. Consideraciones sobre las autorizaciones de los movimientos

- Por motivos de seguridad y protección de las circulaciones, no está permitido autorizar entradas simultáneas a la estación.



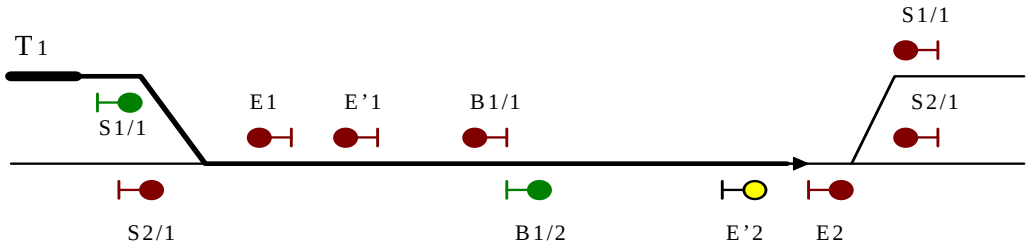
Por tanto, cuando un tren se encuentre en la proximidad de la estación (sección de vía inmediatamente anterior a la señal de avanzada E'1 o E'2 en la figura) se podrá establecer el itinerario del tren a la estación.

De esta manera, para que los trenes T_1 y T_2 crucen en la estación y puesto que las entradas no pueden ser simultáneas, primero se autorizará el movimiento del que antes llegue a la proximidad de la estación. Sólo cuando este tren se encuentre sobre la vía de estacionamiento se podrá autorizar la entrada del segundo tren.

- Cuando una vía de estacionamiento se encuentra ocupada y se desea hacer una entrada sobre esa vía, no todas las instalaciones autorizan este movimiento denominado **rebase autorizado**. Si la instalación no es así hay que establecer comunicación verbal con el maquinista para autorizar la entrada.

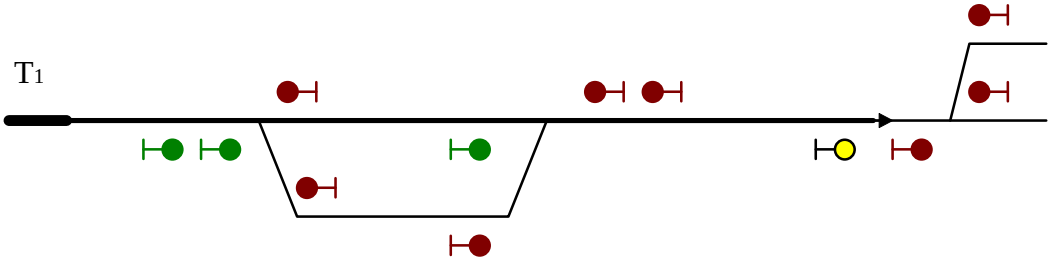
  Este movimiento impone restricciones de velocidad.

- La autorización de salida de un tren de la estación puede hacerse sólo si la instalación de bloqueo autoriza la expedición del mismo hacia la estación colateral.



Además, sólo se podrán efectuar autorización de expedición de trenes en el sentido en el que se encuentre establecido el bloqueo.

- Autorización de paso directo de un tren. Si un tren se encuentra en la proximidad de una estación y no se esperan circulaciones de la colateral, entonces se puede autorizar el paso directo por la estación como suma de los movimientos de entrada y salida de la misma. Por tanto, se permite el avance del tren desde la entrada de la estación hasta la entrada de la estación colateral.



3.2.6.1. Señales ferroviarias.

Para agilizar el tráfico ferroviaria en la estación y en el trayecto entre estaciones se define una secuencia de trenes.

Para ejecutar una regulación de la secuencia entre trenes se disponen en la estación y en el trayecto de señales. Estas señales dan la orden de marcha al tren. Es decir, cada señal protege una sección de vía y el aspecto que muestra al maquinista indica la autorización que se da para avanzar en la vía protegida.

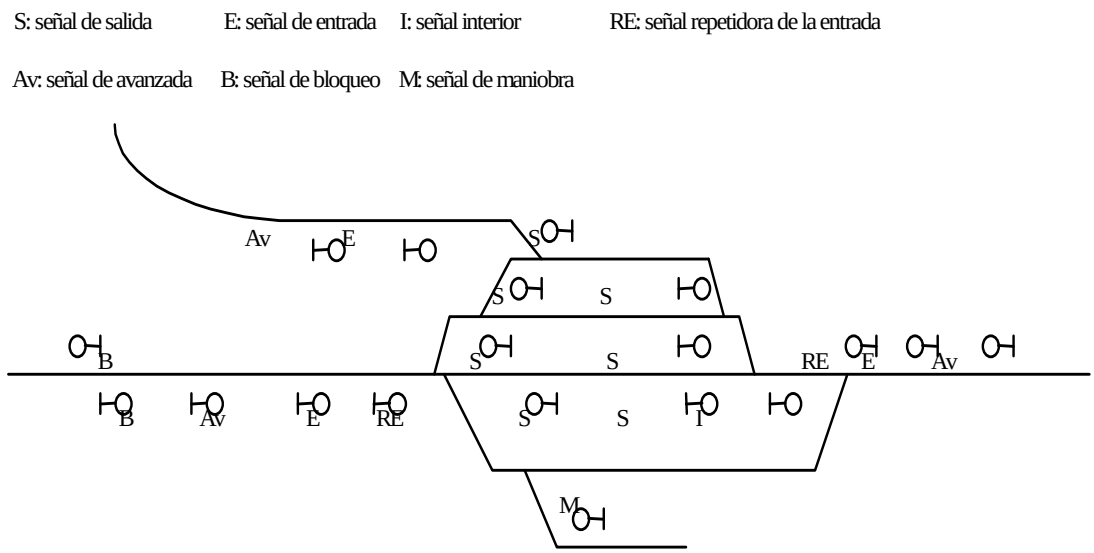
La clasificación de señales ferroviarias por tipos es la siguiente:

- Señal de salida: autoriza el avance del tren hacia la siguiente estación.
- Señal de entrada: autoriza el avance del tren hacia los estacionamientos e informa del estado en que se puede encontrar la señal de salida.
- Señal repetidora de la entrada: recuerda al maquinista la situación en la que se va a encontrar los desvíos que intervienen en el movimiento de entrada.

  Si se está realizando una maniobra de entrada a la estación autoriza la marcha del tren hacia el estacionamiento.
- Señal de avanzada: autoriza el avance del tren hasta la señal de entrada, anunciando el estado de la señal de entrada.
- Señal de bloqueo: autoriza el avance del tren en el trayecto entre estaciones colaterales hasta la siguiente señal.
- Señal de maniobra: autoriza el avance del tren en la estación sólo si se trata de un movimiento de maniobra.

- Señales intermedias en la estación: autorizan un avance parcial del tren dentro de los límites de la estación; por tanto, permiten hacer entradas y salidas interiores en la estación.

En la figura siguiente se muestra la localización de estas señales sobre una cierta configuración de vías de una estación.



Los aspectos que pueden dar las señales, del más al menos restrictivo para el avance del tren, son los siguientes:

- **Rojo** indica parada inmediata.

- **Rojo - blanco intermitente** autoriza el avance del tren con “marcha a la vista”; es decir, sólo se permite el avance del tren para casos de entrada a una vía ocupada o a una topera.

- **Rojo - blanco** sólo autoriza avance del tren en maniobras. La velocidad máxima es de 20 Km/h.

- **Amarillo** indica anuncio de parada. Quiere decir que la siguiente señal se puede encontrar cerrada (rojo). La velocidad máxima es de 30 Km/h.

- **Verde intermitente** indica que hay que reducir la marcha, pues la próxima señal podría encontrarse en amarillo. La velocidad máxima es de 160 Km/h. Esta señal sólo existe en aquellos trayectos donde la velocidad máxima sea superior a 160 Km/h.

- **Verde** indica vía libre. La velocidad máxima la marca la consigna de circulación sobre el trayecto en la que se encuentre la señal.

Además hay que hacer las siguientes consideraciones para las señales:

- La señal de avanzada está en verde - amarillo.

  Es necesaria una reducción de velocidad porque la señal de entrada está cerrada o bien porque está en amarillo.

  Si no hay explotación de bloqueo automático la señal de avanzada no tiene foco rojo.

- La señal de avanzada está en rojo.

  Si la señal de avanzada tiene foco rojo indica este aspecto que el cantón que protege la señal está ocupado o que se está haciendo una maniobra en la estación.

- La señal de entrada está en amarillo.

  La señal de salida se encuentra en rojo o se va a hacer una entrada a vía desviada. En este último caso la velocidad máxima depende del tipo de desvío, estableciéndose un valor de 30 Km/h variable por consigna.

- La señal de entrada está en verde - amarillo.

  El tren va a realizar un paso por la estación por vía desviada. La señal de salida está abierta.

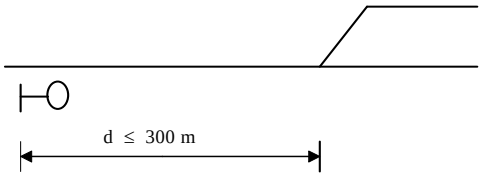
- La señal de salida está en verde - amarillo.

  La señal de salida es avanzada de la estación colateral.

- La señal de bloqueo es una señal permisiva.

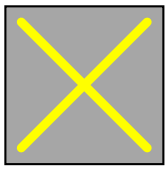
  Las señales de bloqueo se encuentran en el trayecto entre estaciones colaterales. Si un tren avanza en el trayecto y se encuentra una señal de bloqueo en rojo, teniendo ésta además un cartelón con la letra *P* (de permisiva), permanecerá parado ante esta señal un tiempo máximo de 3 minutos. Transcurrido este tiempo proseguirá su marcha a una velocidad no superior a 30 Km/h a fin de permitir su parada ante cualquier obstáculo.

  Por consigna se pueden poner señales indicadoras de velocidad en las señales. Estas indicadoras de velocidad permiten variar los límites establecidos por defecto.

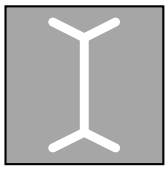


Se define la **señal absoluta** como aquella señal que se encuentra a una distancia no superior a 300m de la punta de la aguja. Estas señales están dotadas de un teléfono que permite realizar una comunicación verbal con el operador para autorizar el avance del tren cuando la señal está cerrada. Estas señales están dotadas de foco rojo, amarillo, verde y blanco.

Otras señales que aparecen en las líneas férreas son las señales ferroviarias de los pasos a nivel. Estas señales muestran al maquinista el estado del paso a nivel, indicando si se encuentra comprobando y la barrera está bajada o, por contra, el paso está averiado y se indica orden de parada. Esta señal luce como sigue:



Orden de parada



Paso a nivel comprobando

3.3. El concepto de seguridad

Las instalaciones que están relacionadas con la detección y avance de los trenes son de seguridad. Este concepto está relacionado con la necesidad de asegurar la respuesta del sistema que gobierna la instalación ante un error.

Los sistemas pueden manifestar averías o fallos. Las averías son causadas por la pérdida del punto de funcionamiento deseado para la instalación, por ejemplo una fusión de la lámpara de una señal. Las averías están bien definidas en el funcionamiento de la instalación y, por tanto, en el sistema que la gobierna. Los fallos originan la pérdida de supervisión y control del sistema y, por tanto, se podría dar un resultado no deseado; esto es, no definido. La instalación es de seguridad cuando al detectarse un fallo en el sistema éste reacciona evolucionando al estado de máxima protección. Este estado de protección va a depender del sistema: señales en rojo, actuar freno del tren, etc.

Cuando el sistema que gobierna una instalación ferroviaria reacciona ante un fallo en la manera descrita, se dice que es de seguridad intrínseca o “fail-safe”.

3.3.1. Instalaciones de seguridad

Los tipos de instalaciones de seguridad se pueden clasificar como sigue:

1. Instalaciones FIJAS, instaladas en la infraestructura para el control y supervisión de la vía y, que a su vez, quedan divididas en los grupos siguientes:

   - ENCLAVAMIENTOS, instaladas en las estaciones y que sirven para el accionamiento de los desvíos de las mismas a fin de formar y supervisar rutas de paso al tren por la estación. Históricamente se han desarrollado los siguientes tipos:

     - BOURÉ, que es la evolución de instalar cerraduras de este tipo, en lugar de candados, para poder mover las agujas.

     - MECÁNICOS, donde el accionamiento de los desvíos se hace mediante palancas que se encuentran centralizadas en una MESA DE MANDO. El mando de una de estas palancas provoca el movimiento del desvío mediante la transmisión de barras y cables hasta los aparatos de vía.

     - HIDRODINÁMICOS, de manera que la transmisión desde la mesa de mando se realiza por fluidos a presión que circulan por una red de tuberías.

     - ELÉCTRICOS, en los que los aparatos de vía disponen de un motor, cuyo mando se ejecuta sobre el CUADRO situado en el Gabinete de Circulación de la estación.

       Las incompatibilidades entre movimientos se realizan eléctricamente mediante asociaciones de relés que se agrupan constituyendo lo que se denominan “módulos” o “grupos”.

     - ELECTRÓNICOS, también llamados de ESTADO SÓLIDO, en los que el accionamiento de los aparatos de vía se consigue con motores eléctricos, pero las incompatibilidades entre movimientos se consiguen utilizando ordenadores.

       El mando de la estación se ejecuta sobre un puesto de operación dotado de monitor viderográfico.

   - BLOQUEOS, que se disponen para control de tráfico en el trayecto entre estaciones colaterales y que pueden agruparse en los siguientes tipos principales:

- Bloqueo Telefónico, abreviado como B.T., que asegura, tanto en vía única como en doble, el aviso de llegada de los trenes, así como la petición y la concesión de la vía mediante telefonemas.

  No existe en el trayecto ningún sistema de detección del tren y se repetirán las operaciones descritas para expedir cada tren entre las estaciones colaterales.

  En la figura se quiere esquematizar las señales que existen (salidas, entradas y avanzadas), así como la existencia de sistemas de detección sólo en cada estación y no en el trayecto.

- Bloqueo Eléctrico Manual, abreviado como B.E.M., done la circulación de los trenes entre estaciones colaterales se regula eléctricamente.

  La explotación se consigue ejecutando sobre los cuadros de mandos de las estaciones colaterales las operaciones necesarias para la toma de vía, su concesión, el establecimiento del bloqueo y la normalización del mismo. Al igual que en el caso anterior, en el trayecto no hay ningún dispositivo de detección del tren y, además, sólo se puede expedir un tren cada vez que se tome el bloqueo.

  En la figura se quiere esquematizar las señales que existen (salidas, entradas y avanzadas), así como la existencia de sistemas de detección sólo en cada estación y no en el trayecto.

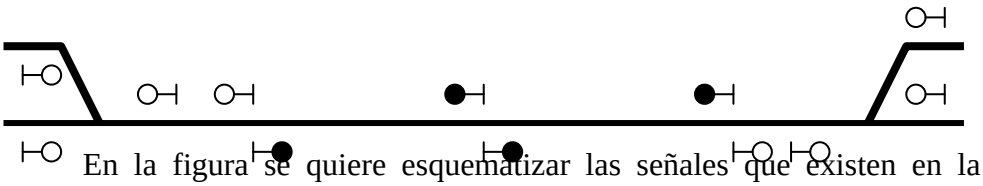
- Bloqueo Automático en vía única, abreviado como B.A.U., que comprende a los enclavamientos de estaciones colaterales y al trayecto entre las mismas.

  En esta instalación todo el trayecto dispone de equipos de detección del tren, de manera que se puede definir el concepto de sección de BAU como el tramo de vía comprendido entre dos señales sucesivas.

En el trayecto existen señales intermedias que, cuando el bloqueo no está establecido dan orden de parada. Sólo cuando el bloqueo está establecido estas señales darán el aspecto apropiado para la protección del tren que se disponga en la sección de vía asociada a la misma. De esta manera, al establecerse el bloqueo la señal puede autorizar el avance del tren; pero, al paso del primer eje, la señal dará orden de parada. La señal volverá a dar orden de avance al tren cuando TODA la circulación haya abandonado la sección de vía a la que dicha señal protege.

El establecimiento del bloqueo se hace por mando o a consecuencia de establecer un itinerario de salida al tren. La detección de la llegada del tren a la estación colateral conlleva, automáticamente, a la normalización del bloqueo cuando el trayecto quede libre.

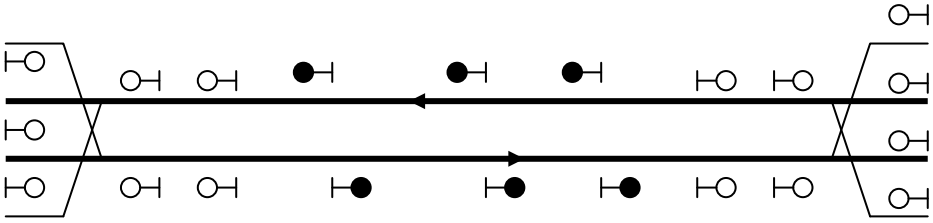
Dada la disposición de señales y de cantones de bloqueo es posible que exista sucesión de trenes en el trayecto en el que está establecido un BAU.



En la figura se quiere esquematizar las señales que existen en la estación (salidas, entradas y avanzadas, marcadas en blanco) e intermedias (marcadas en negro), así como la existencia de sistemas de detección tanto en la estación como en el trayecto.

- Bloqueo automático en vía doble, abreviado como B.A.D., que consiste en establecer un bloqueo de manera permanente en cada una de las vías de circulación y en el sentido normal de circulación de cada una de ellas.

  Existen sistemas de detección del tren en los trayectos y señales de bloqueo en cada una de las vías de circulación y que sólo afectan a los trenes que vayan en el sentido normal de circulación.



En la figura se quiere esquematizar las señales que existen en la estación (salidas, entradas y avanzadas, marcadas en blanco) e intermedias (marcadas en negro) que afectan sólo al sentido normal de circulación (que aparece marcado por una flecha), así como la

existencia de sistemas de detección tanto en la estación como en el trayecto.

- Bloqueo automático en vía doble banalizada, abreviado como B.A.B., que resulta de considerar la vía doble como dos vías dotadas de un B.A.U. cada una de ellas.

- ASFA, que es un sistema para provocar el frenado del tren. Para ello, en la vía y para cada señal se dispone de una baliza que recoge el aspecto de la misma. En el tren se dispone de una antena que capta la señal de la baliza y actúa el freno cuando el régimen de velocidad del tren no se adecúa a la señalización.

2. Instalaciones MÓVILES, instaladas en el material móvil; es decir, instalaciones en los trenes. Se dividen en equipos instalados en locomotoras, automotores, etc.

3. Señalización en cabina para conducción a alta velocidad.

Se requiere la captación de información en la cabina de las indicaciones que permitan la circulación del tren a la alta velocidad de la línea, con lo que el maquinista no depende de las condiciones de visibilidad exterior.

Por ejemplo, como sistemas de transmisión puntual se tiene el ASFA, donde la interacción TREN-VÍA se hace sólo en determinados puntos de la vía y que corresponden a los lugares donde hay instaladas señales. Este sistema, instalado en toda la red de RENFE, satisface las funciones de informe al maquinista del aspecto de la señal, de actuar sobre el freno cuando la marcha del tren no está conforme al aspecto de la señal de la que se ha recibido la información, así como de llevar un registro en el tacógrafo del vehículo. Por último, es posible anular el sistema cuando así se requiera por las condiciones de la explotación en ese momento.

Como sistema de transmisión continua, se requiere una interacción continua entre el TREN y la VÍA para conseguir hacer posible el eliminar la señalización lateral de la vía, si bien ésta pudiera coexistir como señalización de apoyo en determinados casos, por ejemplo si todos los trenes no están equipados con los dispositivos necesarios para captar la señal emitida desde la vía. Dentro de estos sistemas está el L.Z.B., que consiste en comunicar el punto donde debe iniciarse el frenado ante un obstáculo, fijo o móvil, para, con ello, exigir una reducción de la marcha hasta llegar a la parada del tren.

Para conseguir este propósito, el sistema LZB requiere un cable a lo largo de la vía, así como un equipo instalado en el vehículo para captar la información del cable. Además existe un centro de mando donde se encuentran recogidas las informaciones

de los elementos fijos del trayecto (topografía, pendientes, perfiles de velocidad, condiciones variables de la velocidad y derivadas del aspecto de las señales, posición de las agujas o de limitaciones transitorias de velocidad), además de recibir de manera continua las informaciones enviadas desde los trenes (longitud, tipo de frenado, situación, velocidad); de manera que, con estos datos, se determina la orden de marcha para cada tren.

4. El sistema europeo ERTMS.

El ERTM es un sistema de control automático de trenes que permita la circulación mediante un dispositivo que satisfaga una interoperabilidad en todas las redes férreas de Europa.

El criterio de interoperabilidad permite la libre circulación del los trenes por las diferentes líneas de cada administración ferroviaria, sin que por ello sea necesario cambiar de locomotora o de maquinista en las fronteras.

El sistema ERTMS se divide en tres niveles de implantación que, muy brevemente se resumen como sigue:

- NIVEL 1.

  El sistema ERTMS está superpuesto con la señalización existente, típicamente será la señalización lateral.

  La autorización de la circulación es suministrada desde las balizas ubicadas en la vía. Se define así un control continuo de la velocidad, pero con transmisión puntual.

  El cantonamiento es fijo, detectándose la presencia del tren, con ello la localización del mismo, mediante equipos de circuito de vía o contadores de ejes. Además, estos sistemas serán los que se utilicen para definir la integridad del tren.

  No se requiere un sistema vía-radio.

  El interfaz hombre-máquina en la cabina proporciona la interoperabilidad nacional.

- NIVEL 2.

  Puede ser instalado en superposición al sistema de señalización existente o de manera independiente al mismo.

  El cantonamiento es fijo, detectándose la presencia del tren, con ello la localización del mismo, mediante equipos de circuito de vía o contadores de

ejes. Además, estos sistemas serán los que se utilicen para definir la integridad del tren.
La comunicación entre el tren y la vía es continua, empleando la radio como medio de transmisión. Las balizas en vía sólo tienen como misión redefinir la odometría.

La autorización de marcha al tren se comunica por radio desde un centro de bloqueo de radio, puesto que toda la información está centralizada en este centro para permitir la supervisión de las secciones de vía.

El interfaz hombre-máquina es obligatorio y debe proporcionar toda la información necesaria parta llevar a cabo la conducción.

- NIVEL 3.

  Se instala sin señalización lateral y el cantonamiento puede ser fijo o móvil y la integridad del tren se consigue con los dispositivos instalados a bordo del tren.

  La localización del tren viene determinada por su posición relativa en la línea; proporcionándose esta información por el mismo tren, así como por la situación de los itinerarios.

  La comunicación entre el tren y la vía es continua, empleando la radio como medio de transmisión. Las balizas en vía sólo tienen como misión redefinir la odometría.

  La autorización de marcha al tren se comunica por radio desde un centro de bloqueo de radio, puesto que toda la información está centralizada en este centro para permitir la supervisión de las secciones de vía.

  El interfaz hombre-máquina es obligatorio y debe proporcionar toda la información necesaria parta llevar a cabo la conducción.

  A fin de permitir la circulación de los trenes equipados con ERTMS por líneas convencionales, los equipos instalados a bordo incluyen un sistema de interfaz específico a la señalización existente.

3.4. DISPOSITIVOS. AGUJAS, SEÑALES, ELEMENTOS AUXILIARES

3.4.1. Objetivos

Este apartado describe los sistemas que, dentro de la línea férrea, permiten la explotación de las circulaciones en la línea. Dentro del amplio concepto de "explotación", se trata de describir los dispositivos que van a encontrar los trenes en su movimiento y que se destinan múltiples funciones:

- En primer lugar, se describen las agujas entendiéndolas como aquello dispositivos que permiten encaminar a los trenes dentro de una estación o apartadero y que posibilitan la entrada o la salida de los trenes desde los mismos.

- En segundo lugar, las señales permiten indicar al maquinista todos aquellos aspectos relevantes a la marcha del tren; esto es, tanto en la regulación de la velocidad como en la marcación de los puntos singulares de la línea.

- Por último, encuadrados en este capítulo quedan aquellos dispositivos que se utilizan para el control y supervisión de los trenes y de las infraestructuras, además de prevenir o avisar incidencias que pudieran afectar al ferrocarril.

Encuadrado de esta manera el capítulo, se procede, en los apartados siguientes, a desarrollar las principales características de cada uno de estos dispositivos.

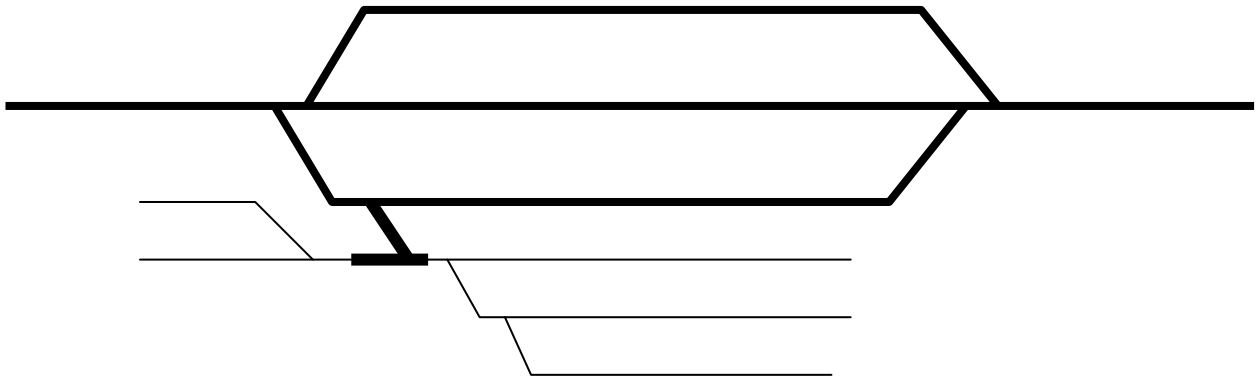
3.4.2. Agujas

El concepto de aguja se aplica a todo dispositivo cuyo objetivo es permitir el paso de una vía a otra.

De esta manera, la formación de rutas de entrada o de salida de los trenes requiere la intervención de uno o varios de estos dispositivos. Dado que el enclavamiento es la instalación encargada de asegurar las rutas a los trenes, quedarán incluidas en sus funciones las correspondientes al control y supervisión de las agujas que se incluyan en cada una de las rutas que se puedan establecer en dicha instalación.

Decir que debe haber un control y una supervisión de las agujas podría inducir, erróneamente, a entender que, asociado a toda aguja, existe algún tipo de dispositivo de mando y comprobación de gobierno automático. Pues bien, esta situación no se requiere siempre y dependerá del tipo de enclavamiento y, más aún, de las condiciones de explotación particulares a la instalación donde éste se aplique. ¿Qué es lo que se está haciendo ver? En primer lugar, que existirán agujas que formen parte de las rutas que pueden ser ordenadas desde el puesto de operación y agujas que servirán para encaminar los trenes y que se operarán localmente (a pie de vía) y que no forman parte de las rutas. Estas últimas suelen ser agujas que están en apartaderos no afectados por las vías principales de circulación.

Para ilustrar esta situación sirva la siguiente figura:



En esta figura se ha representado en trazo grueso cada una de las secciones de vías que intervienen en la formación de rutas de la estación; es decir, en los movimientos de entrada y de salida así como las maniobras centralizadas de la estación. Las agujas que se integran en ellas son mandadas desde el enclavamiento, el mando dependerá de la naturaleza de éste, y se requiere la comprobación de la posición de cada una de ellas para poder autorizar la marcha del tren sobre la ruta.

En esta misma figura, se han marcado con trazo fino aquellas vías que no se ven afectadas por las rutas de la estación. Por tanto, las agujas que quedan en estas vías son accionadas de manera manual y la verificación del estado de las mismas es responsabilidad de los operadores encargados de llevar a cabo la maniobra correspondiente.

Aunque las características mecánicas de las agujas se van a ir describiendo en sucesivos apartados, desde el punto de vista funcional es necesario hacer una clasificación para entender su integración dentro del enclavamiento:

- Aguja sin equipamiento de mando ni comprobación.

  El operador se encarga de mover de manera manual la aguja y de verificar que ésta adquiere la posición deseada. Con esta funcionalidad, esta aguja no puede integrarse en el enclavamiento para formar rutas.

- Agujas dotadas de comprobador.

  El operador se encarga de mover la aguja, pero existe información del estado de la misma. Por tanto, se pueden incluir condiciones sobre el estado de esta aguja en las rutas que pasen por ella.

- Agujas de mando por palanca.

  La aguja se manda desde el puesto de operación de la estación y, además, existe comprobación del estado de la misma. Por tanto, en un enclavamiento de concentración de palancas, estas agujas podrán intervenir en las rutas.

- Cerradura eléctrica.

La aguja dispone de comprobación de su posición y el mando es manual. Sin embargo, la aguja sólo puede ser manipulada previa autorización de operación desde el gabinete de circulación; es decir, desde el puesto de operación de la estación. La autorización permite extraer una llave para liberar la cerradura que impide el movimiento de la aguja.

Estas agujas, dado que se dispone de su posición, podrán incorporarse en las rutas permitidas por los enclavamientos de naturaleza eléctrica.

- Aguja dotada de motor.

Estas agujas disponen de un motor cuyo mando se da desde el enclavamiento, que será de naturaleza eléctrica/electrónica, y de un comprobador de la posición de la misma. Con ello, estas agujas quedan integradas en las funciones de aseguramiento de las rutas que pueden ser mandadas desde el enclavamiento.

Hecha la descripción funcional, a continuación se darán las definiciones mecánicas de las agujas, además de una evolución de las mismas.

3.4.2.1. Definición de componentes.

La aguja o desvío es un aparato de vía cuyo objetivo es el permitir el paso de los trenes de una vía a otra.

El caso más simple que existe es el que se conoce como sencillo y que permite el paso de una vía a otra, por consiguiente de dos vías. La primera vía recibe el nombre de directa y la segunda vía es la denominada vía desviada.

Las partes mecánicas constituyentes de una aguja son: aguja y contraguja. La aguja es la parte móvil excepto en su talón que es la parte más próxima al cruzamiento. La contraguja es fija y exterior a la aguja. Así mismo se expone que todo desvío está constituido por dos agujas y dos contragujas.

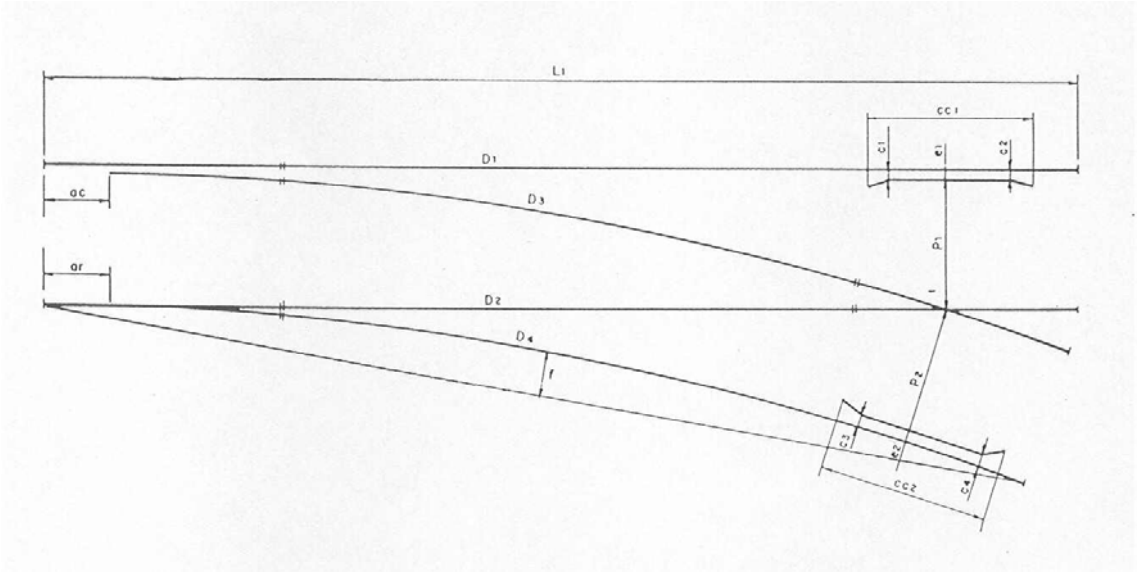
Respeto de su posición y según el sentido de la circulación, se dice que un cambio puede ser tomado de punta o de talón. En ambos casos y cuando la circulación va a pisarlo, la aguja que da continuidad a la vía por donde se ha de circular debe estar acoplada a su respectiva contraguja.

El concepto de punta o de talón es el siguiente:

- Aguja de punta, cuando la circulación entra al desvío por la parte de la punta del mismo.
- Aguja de talón, cuando la circulación entra al desvío por la parte del cruzamiento, o sea, por el talón de la misma.

Otro concepto a tener en cuenta y que viene dado como consecuencia de todo cuánto se ha expuesto con anterioridad es el concepto de posición normal o invertida de un desvío. La posición normal de un desvío es aquella en que la aguja se encuentra acoplada a su contraguja para el paso del tren por la vía directa. La posición invertida de un desvío es aquella en que la aguja se encuentra acoplada a su contraguja para el paso del tren por la vía desviada.

El siguiente dibujo es un esquema de concepto de la aguja.



3.4.2.2. Evolución de los mecanismos de control y supervisión para el cambio de posición de aguja.

Son los aparatos destinados a la maniobra para control y supervisión de las agujas de un desvío y se clasifican como sigue:

- Aparato local de maniobra de aguja (marmita)
- Palanca
- Accionamientos de aguja

Los aparatos utilizados para el control y supervisión de la posición de las agujas de un desvío, son los siguientes:

- Comprobador mecánico
- Comprobador eléctrico de posición de aguja
- Accionamiento hidráulico incluida en palanca
- Accionamiento eléctrico de aguja
- Accionamiento electrohidráulico de aguja

Seguidamente se da una breve descripción de cada uno de ellos.

3.4.2.2.1. Aparato local de maniobra de aguja (marmita)

Este apartado de maniobra es el más sencillo que existe y consta de dos partes:

- una fija, sobre las dos primeras traviesas de un desvío
- otra móvil, que consta de un vástago con su brazo de palanca y de todo en su extremo con un contrapeso.

Se acopla al tirante de maniobra mediante su correspondiente bulón y al tirante de conexión de las agujas del desvío o al cerrojo de uña.

3.4.2.2.2. Palanca

Este aparato, por lo general, montado en una mesa de enclavamiento mecánico aunque también puede montarse fuera de mesa.

La misión de este aparato es transmitir al accionamiento mecánico y mediante una transmisión bifilar formada por alambre de acero galvanizado de 5 mm. de diámetro, el esfuerzo necesario para la maniobra de las agujas componentes de todo desvío.

Los elementos componentes de la palanca son los siguientes:

- Empuñadura con una maneta o falleba

- Polea con su soporte de fijación

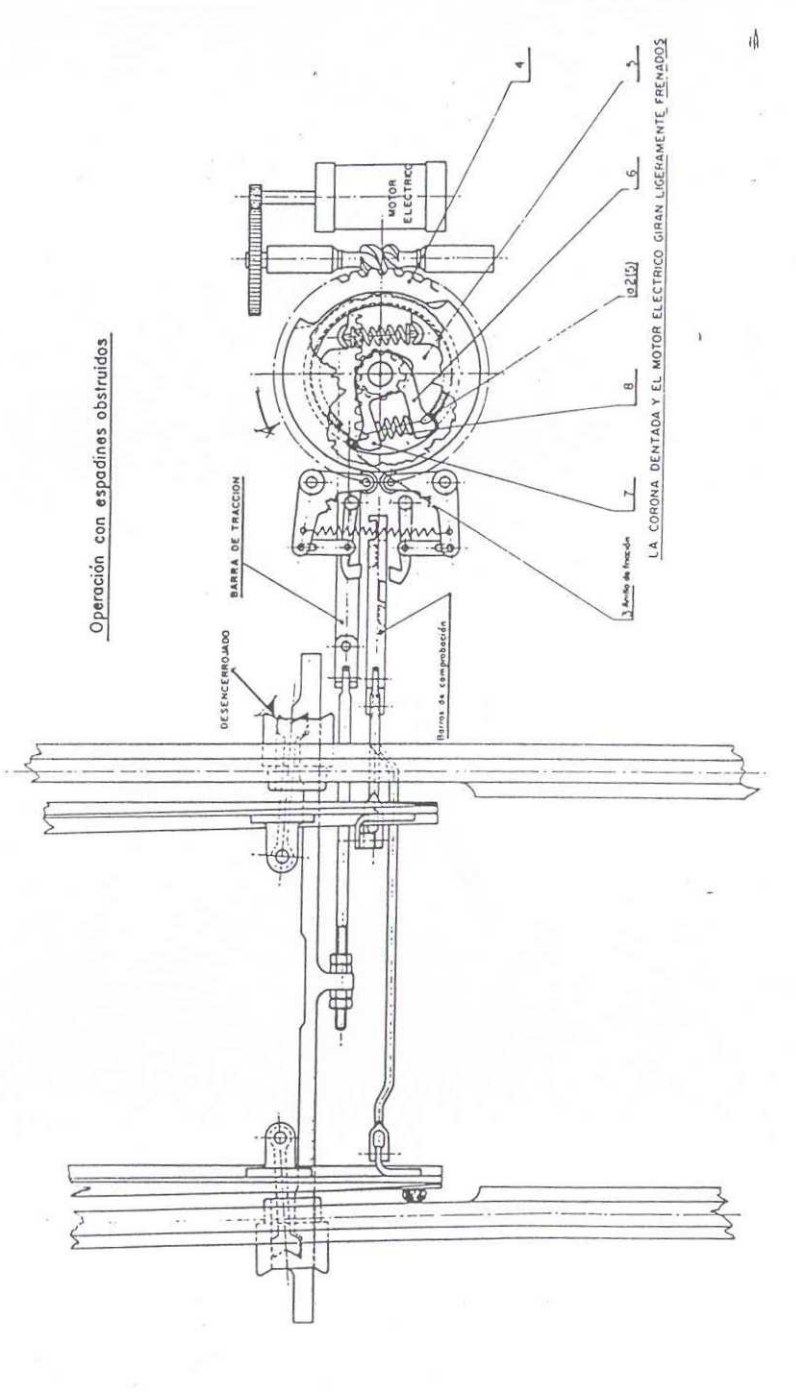
- Cerrojo de fijación al soporte tanto para la maniobra del desvío a posición normal como a posición invertida.

3.4.2.2.3. Accionamiento de aguja

El accionamiento de aguja es un dispositivo que aprovecha la energía que recibe con el fin de realizar el movimiento del correspondiente aparato.

Dependiendo de la energía empleada al aparato se denomina:

- Accionamiento mecánico cuando la energía recibida es mediante transmisión bifilar
- Accionamiento hidráulico cuando la energía recibida es hidráulica
- Accionamiento eléctrico cuando la energía recibida es eléctrica, sirviendo como esquema del mismo la figura adjunta.
- Accionamiento electrohidráulico cuando la energía recibida es hidráulica con parte eléctrica



Esquema de un accionamiento eléctrico

3.4.2.2.4. Aparatos para el control y supervisión de las agujas

Comprobador mecánico

Este aparato está destinado a la comprobación de la posición de las agujas componentes del desvío

Su funcionamiento es mediante transmisión bifilar y maniobrado con palanca.

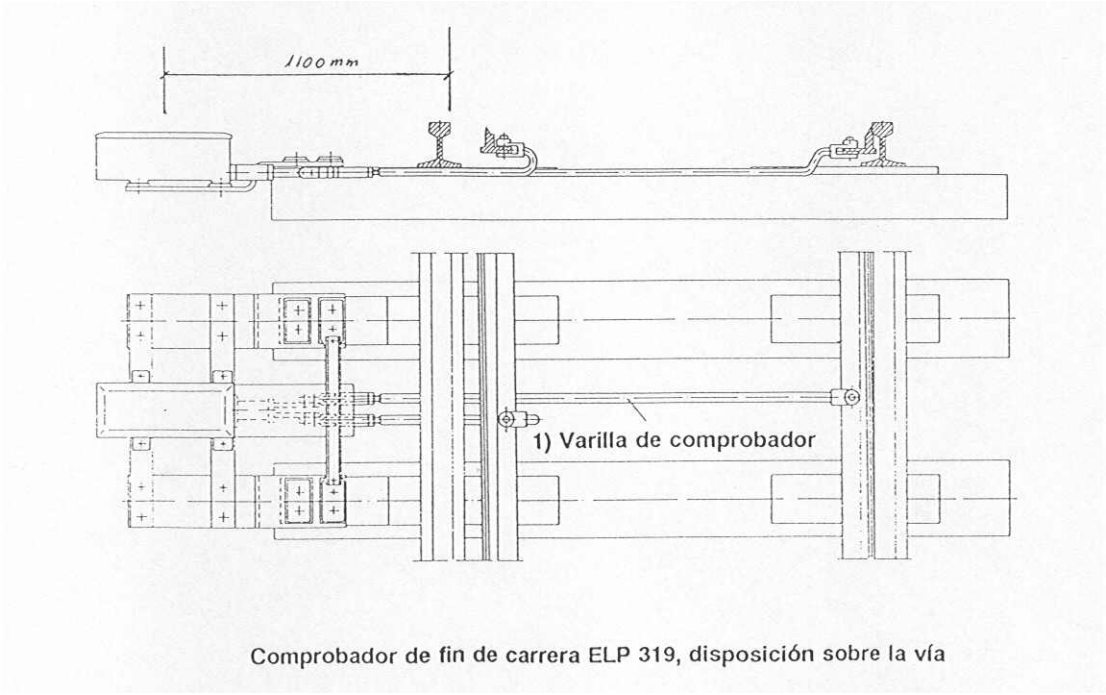
Caso de que las agujas a comprobar no cumplan las condiciones de acoplamiento y encerrojamiento correcto, el cerrojo de comprobación no puede cumplir la misión para la que está destinado.

Comprobador eléctrico de posición de aguja

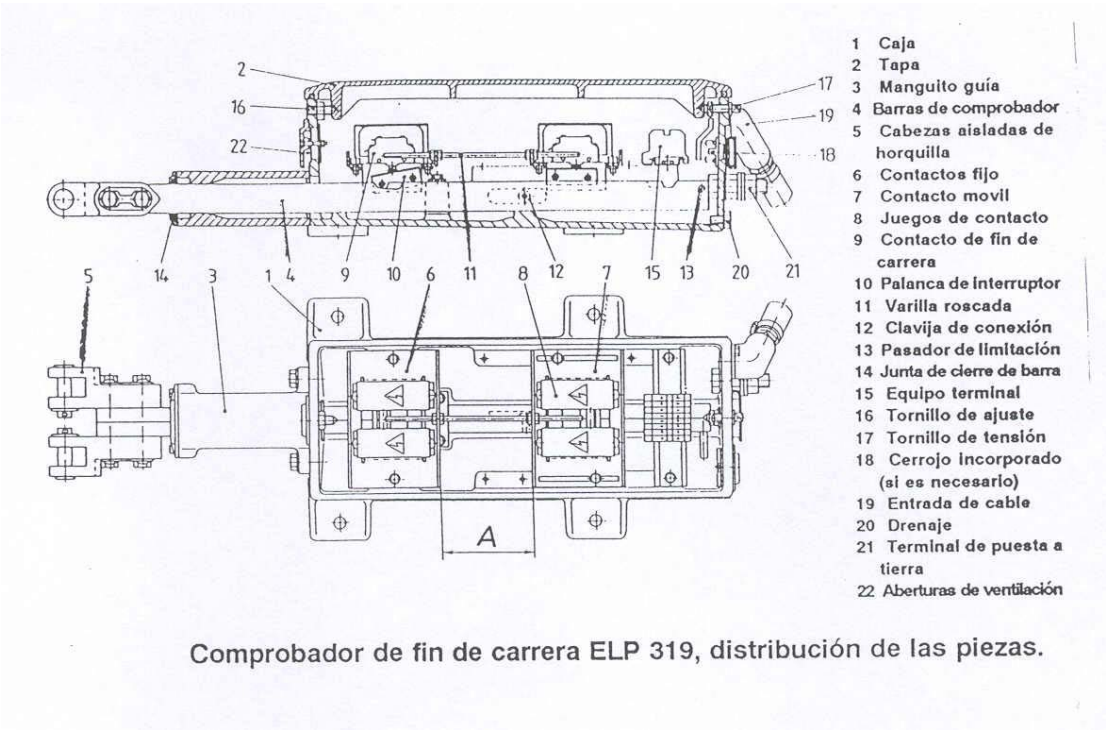
El comprobador eléctrico está destinado a realizar la comprobación de las agujas componentes del desvío.

El funcionamiento de este comprobador se basa en que el movimiento realizado por las agujas para su acoplamiento o desacoplamiento es transmitido a las bielas del comprobador por medio de los tirantes de comprobación, los cuales por medio de las levas y a través del rodillo respectivo, actúan sobre la palanca de accionamiento de contactos, comprobando dichos contactos la correcta posición de las agujas.

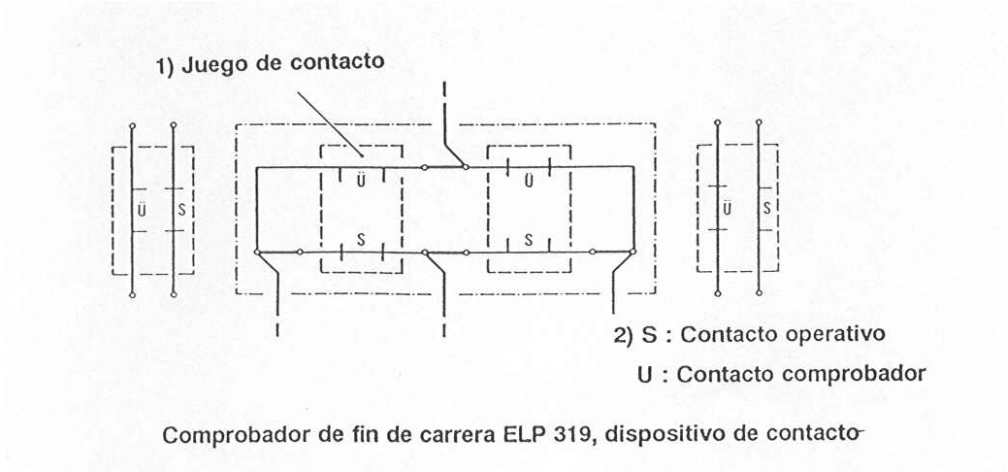
Las siguientes figuras esquematizan a este comprobador.



Comprobador de fin de carrera ELP 319, disposición sobre la vía



Comprobador de fin de carrera ELP 319, distribución de las piezas.



Comprobador de fin de carrera ELP 319, dispositivo de contacto

3.4.2.2.5. Accionamiento hidráulico incluida en su palanca

El elemento de comprobación de que consta un accionamiento hidráulico, es lo que se denomina el distribuidor de comprobación.

Su modo de funcionamiento se basa en el trasvase de un fluido a medida que la palanca se mueve, primero hasta los 45 ° y luego desde este ángulo hasta completar su recorrido.

3.4.2.2.6. Accionamiento eléctrico de aguja

Para la comprobación de la posición final de las agujas, el accionamiento eléctrico dispone de los siguientes elementos:

- Unas barras correderas en plano superior e inferior
- Un conjunto de guía
- Cerrojos
- Manivelas ajustables.

3.4.2.2.7. Accionamiento electrohidráulico de aguja

En este tipo de accionamiento es de aplicación todo cuánto ha sido expuesto en el apartado anterior.

3.4.2.2.8. Cerradura eléctrica

La cerradura eléctrica es utilizada en aquellos casos en que existen desvíos conjugados con calce y que se aseguran mediante cerraduras Bouré y su aparato de maniobra es con una marmita.

Constan de los siguientes elementos:

- Una caja metálica en cuyo interior se encuentra:
  - Una llave Bouré
  - Un visor de autorización con su conmutador
- Un mástil para la fijación de la caja

Su funcionamiento es el siguiente:

Una vez que ha sido autorizada por el jefe de circulación se tiene:

- Encendido del visor luminoso
- Extracción de la llave Bouré mediante giro de la misma.
- Con esta llave se procederá a la apertura de la cerradura Bouré de campaña instalada en la marmita y destinada a su inmovilización.

3.4.2.3. Talonamiento. Dispositivos talonables.

3.4.2.3.1. Talonamiento

El talonamiento viene dado como consecuencia de producir la inversión de las agujas de un desvío que se encuentran en posición distinta para la ruta en que están preparadas y siempre esperando a la circulación por el talón del desvío.

Cuando esto ocurre, no se provocan daños materiales o son de escasa consideración en el caso de que el accionamiento del desvío sea del tipo talonable. En caso de que el accionamiento sea de tipo no talonable los daños materiales son importantes e incluso provocar un descarrilo.

3.4.2.3.2. Dispositivos talonables o de llamada retardada

Son unos dispositivos de maniobra utilizados para el apartado de trenes en estaciones con vía única y sin necesidad de agentes encargados de la maniobra de los desvíos.

El campo de aplicación de estos dispositivos es en aquellos desvíos dotados de cerrojo de uña debido a que dicho cerrojo es talonable. Otras de las condiciones exigidas para los desvíos dotados con estos dispositivos son:

- La velocidad máxima del tren a su paso por el desvío es de 30 km/h.
- Los vehículos ligeros y pequeños no deben franquear dichos desvíos.
- En caso de que se produzca la detención de una circulación en la zona del desvío, la dirección de la marcha debe ser mantenida.

Los desvíos pueden hacerse manualmente, empleando para ello la marmita:

- El resorte de retardo deberá quedar sin efecto utilizando para ello una llave especial
- La función del dispositivo queda como tirante de maniobra y la aguja no puede ser talonada.

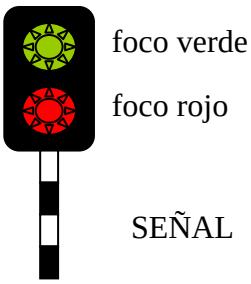
La mejor aplicación que posee este dispositivo es en las líneas de débil tráfico y complementando con la instalación de un comprobador eléctrico de posición de aguja y una señal alta de un solo foco con indicación blanco.

3.4.3. Señales

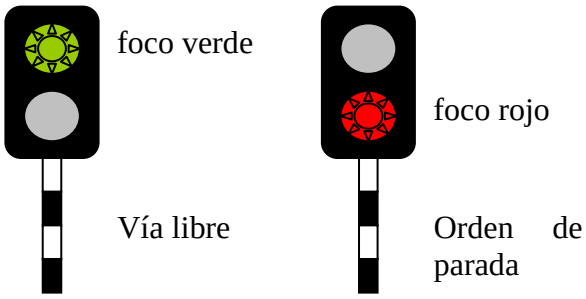
Las señales se disponen en la vía para anunciar al tren diferentes aspectos de la línea y que hacen que se deban definir diferentes grupos. Es difícil hacer una clasificación general, pues van a depender de cada administración ferroviaria y quedan recogidas en el Reglamento de Circulación propiedad de dicha administración.

- Señales que definen la autorización de marcha al tren conforme al estado de las instalaciones de seguridad. Estos aspectos, de manera simplificada, corresponden a las indicaciones de vía libre y a la de orden de parada, existiendo tantos aspectos intermedios como defina la correspondiente administración encargada de la explotación de la línea.

  Estas señales pueden ser luminosas o mecánicas, estas últimas tienden a desaparecer a medida que los enclavamientos pasan de ser de naturaleza mecánica a los eléctricos/electrónicos. Por tanto, nos centraremos en las señales luminosas, que se componen de un conjunto de focos, básicamente dos: verde, que indica vía libre, y rojo, para dar la orden de parada. Por ejemplo, la siguiente figura sería un ejemplo de este tipo de señal:



  Y, siguiendo con este ejemplo, los aspectos correspondientes a vía libre y a la orden de parada son los siguientes:



Los aspectos de las señales y las consecuentes imposiciones de velocidad al tren se definen en el reglamento de circulación de cada administración ferroviaria, tal y como se ha dicho al principio de este punto.

Las señales que se están recogiendo en este apartado son aquellas que se gobiernan desde el enclavamiento o desde los bloqueos; por tanto, son las señales que se encargan de autorizar movimientos de avance al tren, tanto en las estaciones como en los trayectos. Los tipos de señales van a depender de la administración ferroviaria y las autorizaciones a las que da lugar cada una de ellas quedará recogida en el reglamento de circulación, así como en la consigna de explotación particular a cada instalación Así pues y a modo de resumen generalizado, las señales que se definen aquí son:

- Señales de entrada, que permiten el avance de un tren hacia la zona de estacionamiento.

- Señales de salida, que autorizan el movimiento del tren desde la zona de estacionamiento hacia los trayectos a las estaciones colaterales.

- Señales de maniobra, que autorizan movimientos interiores en la estación y que están destinados a operaciones de maniobras de los trenes.

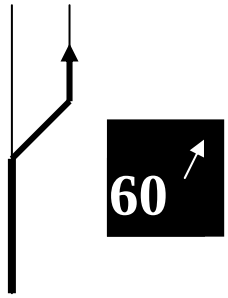
- Señales de avanzada, que sirven para representar el estado al que autoriza la señal de entrada a la que preceden.

- Señales repetidoras de la entrada, cuya misión es recordar el estado de la entrada durante el avance del tren en un recorrido correspondiente a la entrada a la estación.

- Señales de trayecto, que son las que autorizan la marcha del tren entre estaciones colaterales al amparo de un bloqueo entre las mismas.

- Señales de indicación del estado de los pasos a nivel, que dan al tren la información el estado de las protecciones del paso a nivel protegido; es decir, si las protecciones de la instalación a la carretera están actuando de manera correcta o, por el contrario, existe alguna incidencia en las mismas.

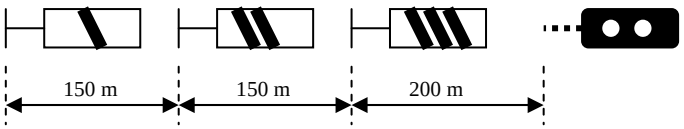
- Señales indicadoras, por ejemplo de la dirección que va a seguir el tren durante su ruta, de la posición de las agujas, de velocidad asociada al movimiento. Para clarificar estas señales, la siguiente figura ilustra la posibilidad de indicar tales situaciones:



En esta figura se indica que el movimiento marcado es autorizado con una velocidad de 60 Km/h.

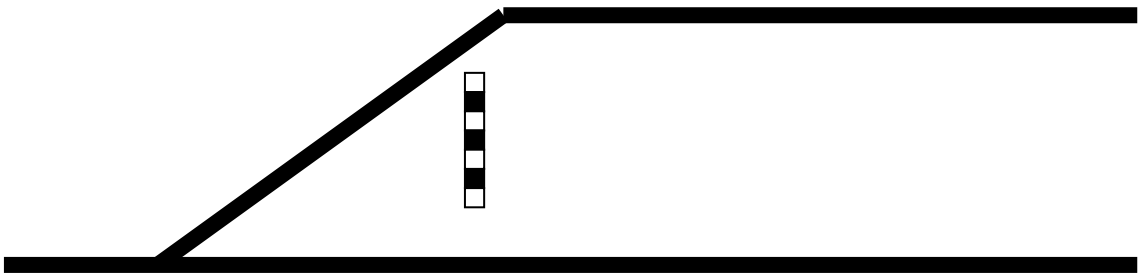
- Pantallas de proximidad, que son aquellas señales fijas que se disponen antes de una señal de avanzada o de una señal de paso a nivel cuando no hay visibilidad de la

misma a una cierta distancia. Esta distancia es recogida también en el reglamento de circulación que, para el caso particular de RENFE, es de 300 m.

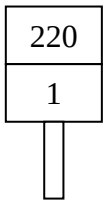


Estas pantallas se disponen de manera que se pueda parar un tren ante la entrada; contando esta distancia desde la última pantalla, que es la más cercana a la señal de avanzada.

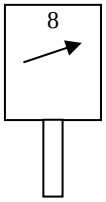
- Piquete de agujas, que es la señal que se dispone en la convergencia de dos vías, accesibles merced a la disposición de agujas, para anunciar el punto hasta el cual es compatible la circulación por dichas vías.



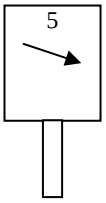
- Postes kilométricos, hectométricos e indicadores de rasante, que son postes fijos que se colocan a lo largo de la vía para indicar, bien la situación kilométrica (hectómetros en su caso), bien la rasante (rampa, horizontal o pendiente) en milímetros por metro a recorrer.



poste kilométrico



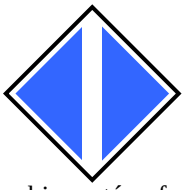
rampa de 8 mm en 850 m



pendiente de 5 mm en 400 m

- Señales indicadoras para la tracción eléctrica y que van orientadas a indicar parada, apertura o cierre del regulador, bajada o subida del pantógrafo y comienzo y final de la zona neutra.

bajar pantógrafo

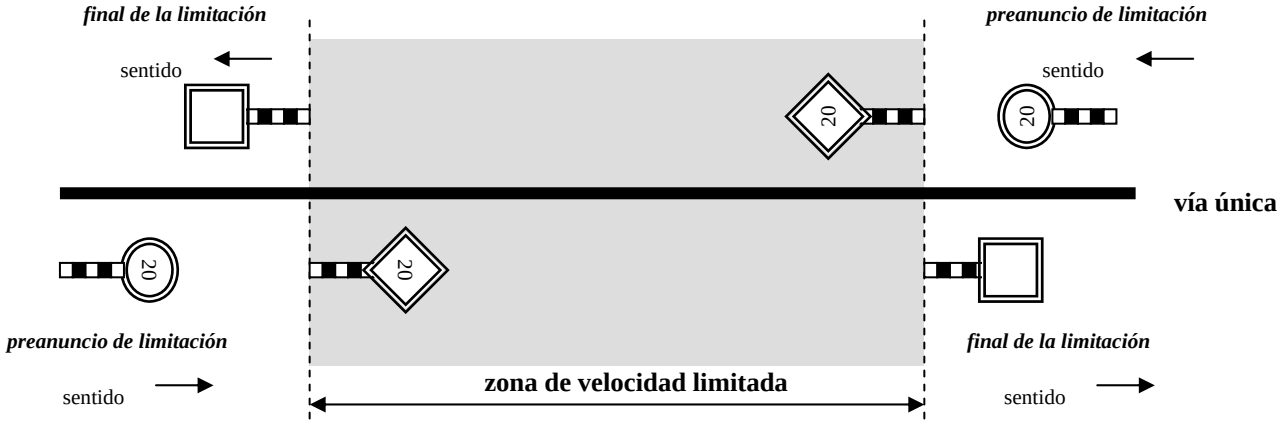


subir pantógrafo

- Cartelones, que indican al maquinista órdenes o indicaciones independientes del resto de las señales fijas. Llevan inscritas letras, palabras, números o figuras.

S	P.N. Km 123 a 500 m	T
Silbido de atención (obras, PN guardado, PN sin guardar,…)	Proximidad en metros de un paso a nivel	Indica que hay emplazado un teléfono

- Señales de limitación de velocidad, que indican un preanuncio de la reducción de velocidad, una limitación de velocidad y un final de la limitación de la velocidad.



Tal y como se observa en la figura, la zona afectada por la limitación de velocidad, marcada y sombreada en la misma, dispone de una señal, por cada sentido de circulación, que da la orden de mantener la marcha a 20 Km/h en toda la zona. Previa a cada señal de limitación, se dispone una señal por cada sentido de marcha para anunciar la reducción de la velocidad. Por último, por

cada sentido de circulación, se dispone de una señal que indica el final de la limitación de velocidad.

3.4.4. Dispositivos auxiliares

Los diferentes dispositivos que se van a estudiar en este apartado se caracterizan porque han de satisfacer un doble objetivo:

- Por un lado, el control y la vigilancia del buen funcionamiento de los trenes y de las infraestructuras dispuestas para el tránsito de los mismos.
- Por otro lado, deben servir como sistemas de prevención y aviso de incidentes que puedan afectar al funcionamiento.

Esta doble necesidad hace, en sí misma, una clasificación natural de estos dispositivos.

Existe una gran cantidad y variedad de dispositivos, de los cuales, sólo los más representativos van a ser descritos en términos muy generales.

3.4.4.1. Dispositivos en trenes e infraestructura

En este apartado se va a hacer una doble clasificación, según se instalen estos dispositivos en los trenes o en las infraestructuras.

3.4.4.2. Dispositivos en la infraestructura

- Detector de cajas calientes.

El objetivo es encontrar elementos calientes en la composición del tren. De esta manera, se instalan dispositivos en la vía que disponen de sensores térmicos para detectar “ejes” que están sometidos a un calentamiento y que, de seguir el tren la marcha, podrían provocar una rotura en dicho eje y, con ello, un accidente.

La información de estos dispositivos es centralizada en algún puesto de operación, de manera que la detección del fallo impedirá que se dé orden de avance al tren, aún cuando existieran todas las condiciones para ello.

Los inconvenientes a este sistema son los siguientes:

- Sólo detectan averías al paso del tren por el dispositivo; es decir, el fallo sólo se detecta en el instante de paso por el detector.
- No se detecta el fallo cuando éste ocurre en un punto sin detector.
- Se generan falsas alarmas que, en general, tienen su origen en los frenos.

- Detector de planos en las ruedas, sobrecargas, pesos y carga uniforme.

  En España es más conocido como Detector de Impactos de Vía (DIV) y tiene una múltiple funcionalidad.

  El sistema se basa en un conjunto de extensómetros, ubicados entre traviesas, soldadas al alma de los carriles y en número que va a depender de la cantidad de ruedas que se quiera detectar. El sistema sólo funciona a velocidades superiores a los 25 Km/h.

  La información sirve para determinar el tamaño y la frecuencia de los impactos dañinos, para medir el peso de los ejes y, por tanto, los trenes y el tráfico de la vía; además, se utiliza este dispositivo para hacer una detección de los ejes en términos de sobrecarga o de desequilibrio en los mismos y también es de utilidad para medir planos en las ruedas.

Con la información extraída es posible definir una base de datos de “ejes de trenes”, cuyo objetivo es ver la evolución de los defectos en las circulaciones. También es posible utilizar esta modificación para modificar la velocidad máxima de un tren que lleva dañada una cierta rueda y, llegado al caso, retirar el tren de la circulación.

- Medidor de perfil de ruedas.

  Estos equipos se diseñan para control del mantenimiento de los trenes, particularmente la importancia de tener las ruedas de un perfil correcto cuando van a circular a gran velocidad.

  El dispositivo es autónomo y se ha de instalar en algún punto donde el tren circule a poca velocidad; por ejemplo, talleres, túneles de lavado, etc. En sí mismo, el sistema está basado en técnicas de visión artificial que recogen la imagen de la rueda y la contrastan con unos patrones disponibles en una base de datos. El objetivo es detectar el desgaste de la rueda y determinar el momento en el que se ha de realizar el mantenimiento o, si fuera el caso, recomendar velocidades máximas en función del estado de las ruedas.

- Detector de la dinámica del pantógrafo.

  Aunque cada pantógrafo se calibra en el taller de mantenimiento, este calibrado es estático y, por tanto, el comportamiento dinámico puede diferir.

Estos sistemas miden el levantamiento del hilo de la catenaria al paso del tren, teniendo en cuanta la velocidad del tren y la dirección del viento. El objetivo es crear una base de datos para cada pantógrafo para evitar situaciones peligrosas que pudieran conducir a un enganchón de la catenaria.

- Detector del estado del pantógrafo.

Este sistema es complementario del anterior y cuyo objetivo es medir la existencia de carbones o de superficies que pudieran acelerar el desgaste del hilo de la catenaria o conducir a un enganchón del mismo. Para ello se aplican técnicas de visión artificial.

El sistema se instala en lugares donde el tren pase a baja velocidad, como es el caso de túneles de lavado para locomotoras.

3.4.4.3. Equipos instalados en el tren.

Estos sistemas se conciben como complementarios a los instalados en las infraestructuras y su objetivo es controlar los parámetros de los trenes.

Así pues, se encuadran en este conjunto de dispositivos los siguientes sistemas:

- Detector de puntos duros de la catenaria.

  El sistema se basa en disponer de un pantógrafo instrumentado a fin de medir la aceleración del pantógrafo al paso por la vía; generando una base de datos que permita definir los puntos duros, el control de su evolución y la efectividad de las tareas de mantenimiento.

- Control geométrico de la vía.

  Para este control se ha de disponer de un coche especializado para incluir medidas de desgaste lateral y superficial de los carriles, auscultación de la catenaria y medidas de gálibo como, por ejemplo, distancias entre carriles y obstáculos, andenes, postes, etc.

- Control de calidad de la rodadura.

Este sistema se concibe como complementario al anterior y el objetivo es medir la comodidad de los viajeros, expresada ésta en términos de calidad percibida por los mismos.

- Detector de la adherencia de la rueda al carril.

Medir la adherencia entre el carril y la rueda es importante porque va depender de las características del medio. Así, tanto la humedad, como la lluvia, así como las caídas de las hojas de los árboles provocan variaciones de la adherencia que afecta a la aceleración y a las distancias de frenado de los trenes.

3.4.4.4. Prevención y aviso de accidentes

Se recogen en este apartado a aquellos que captan información ajena al ferrocarril, pero que pueden provocar situaciones de peligro sobre el mismo.

Dentro de estos sistemas se agrupan los siguientes:

- Detector de corrimientos de tierras.
- Detector de movimientos sísmicos.
- Detector de caída de vehículos y objetos en puentes que se encuentren sobre la vía.
- Detector de desprendimientos o piedras a la vía.
- Detector de condiciones medioambientales; dentro de los cuales se integran dispositivos de medidas diversas como vientos elevados, condensación y humedad,

bajas temperaturas que pudieran provocar la congelación de elementos vitales (desvíos por ejemplo), temperaturas elevadas que provocaran deformaciones en los carriles o afectasen a los desvíos, inundaciones, etc.

- Detector de integridad de los túneles.

- Detector de choques de vehículos de carretera contra puentes.

4. Arquitectura SIMENC

4.1. Contexto del Sistema

4.1.1. Definiciones y Acrónimos

Acrónimo	Significado
TAS	Soluciones de Automaticación del Transporte.
Alcatel-TSD	División de Soluciones de Automatización del Transporte de Alcatel.
ADD	Documento de Diseño de la Arquitectura.
DDD	Documento de Diseño Detallado.
DocP	Plan de Documentación.
DPT	Herramienta de Preparación de Datos.
FEC	Controlador de los Elementos de Campo.
IFB	Placa de Interfaz.
CAN	Controlador del área de red.
EC	Controlador de Elemento.
PT	Elemento de Punto.
IM	Módulo de Enclavamiento.
IP	Protocolo de Internet. Protocolo de la capa de red (capa 3) que contiene información de direccionamiento y cierta información de control que permite el envío de paquetes.
TCP	Protocolo de Control de Trasmisión. Proporciona una transmisión fiable de datos en un entorno IP. TCP corresponde a la capa de transporte (capa 4) del modelo de referencia OSI. Entre los servicios que presta se encuentran: el flujo TCP de datos de la transferencia, la fiabilidad, control de flujo eficiente, operación full-dúplex, y la multiplexación.
MMI	Man Machine Interface.
UML	Lenguaje de Modelo Unificado.
argoUML	Herramienta UML de la Universidad de California (http://argouml.tigris.org).
DS	Sistema de Diagnóstico.
IFB-SIMCAM	Subsistema que ofrece en el sistema de simulación la misma interfaz y el comportamiento de la FEC como el IFB en un sistema real.
SIMENC	Subsistema que proporciona el MMI para el Sistema de Simulación.
CAN Bus Monitor	Componente GUI en SIMENC que proporciona información y añade funciones del simulador IFB.

Acrónimo	Significado
SRS	Especificación de Requisistos del Sistema.
ValP	Plan de Validación.
VerP	Plan de Verificación.
VGR	Representación gráfica.
Application Program	El software diseñado para satisfacer los requerimientos funcionales del usuario. Para desarrollar funciones de librerías de baja complejidad usados por esta aplicación se ha usado lenguaje de alto nivel.
Bitmap	Símbolo básico usado por SIMENC para la representación gráfica de elementos.
Interface	Término genérico que designa la frontera compartida entre dos unidades funcionales. Una interfaz puede ser funcional o física o ambas.
Software (SW)	Todo el código (fuente y código objeto), la documentación y herramientas de desarrollo de software que son necesarios para regular y controlar el funcionamiento de un sistema de procesamiento de datos. A menos que se indique lo contrario, la palabra de Software deberá describir el software, firmware o Software en Sistemas Embebidos.
System or Sub-system	Un complemento del hardware y del software funciona como un sistema integrado. Este incluirá todo el software para ser operativo en su hardware. El sistema deberá incluir todas las versiones de software desarrollado y controlado en la ejecución de este Contrato, todas las herramientas y equipos utilizados para el desarrollo, mantenimiento, operación y capacitación para el uso del sistema.
Use Case	Una forma específica de usar el sistema desde una perspectiva del usuario (actor).
NI	National Instruments

TABLA: LISTA DE ACRÓNIMOS

El Simulador IFB está compuesto por los siguientes subsistemas:

- Subsistema **SIMENC**, que proporciona una interfaz hombre-máquina para la Simulación.
- Subsistema **IFB-SIMCAM**, que permite la comunicación de ECs a través de una tarjeta CAN, simula el comportamiento de IFBs y se comunica con SIMENC.

El conjunto completo del Simulador IFB está representado en la siguiente figura:

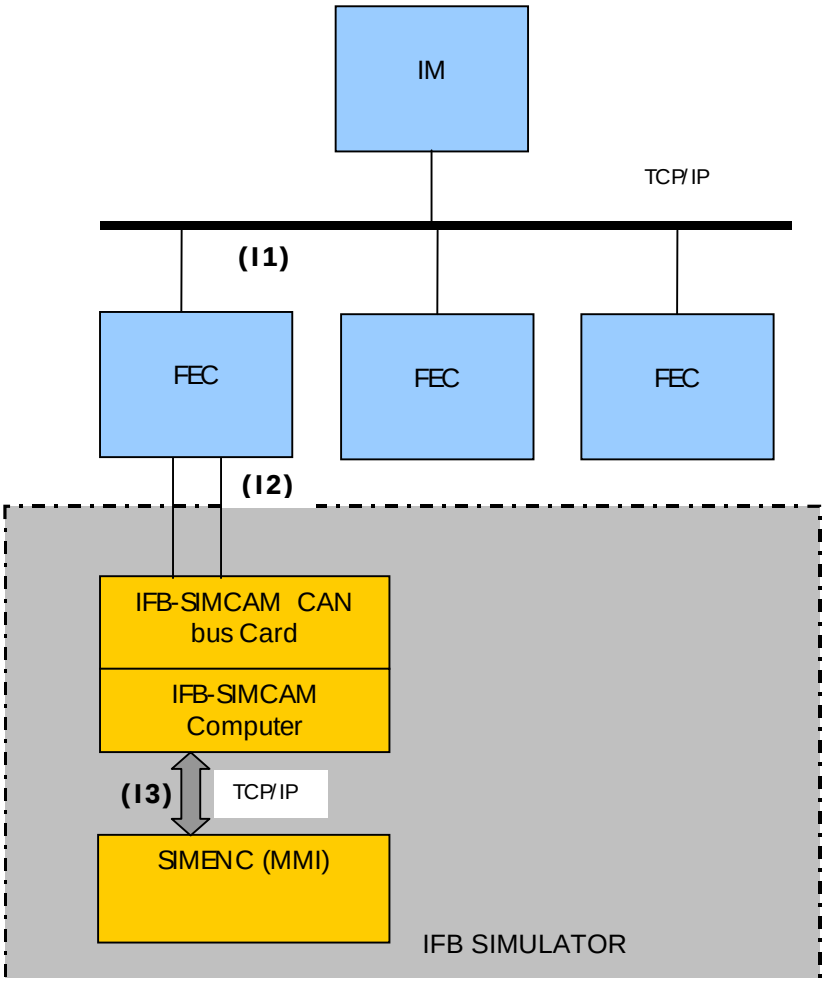


FIGURA: DIAGRAMA DEL SIMULADOR IFB

Las características de las interfaces del simulador de campo, mostradas en el diagrama anterior, son:

Interfaz	Protocolo			Entidades comunicadas	
	Físico	Datos	Red		
I1	Cat 5	Ethernet	TCP/IP	IM	FEC
I2	CAN	CAN	CAN	FEC	IFB-SIMCAM
I3	Cat 5	Ethernet	TCP/IP	SIMENC	IFB-SIMCAM

TABLA: PROTOCOLOS E INTERFACES

4.2. Diseño General del Simulador IFB

El Simulador IFB ha sido desarrollado siguiendo la metodología de *Rational Unified Process*, usando UML y la herramienta argoUML. Este documento está basado en estas herramientas y muchos de los diagramas han sido importados de argoUML.

- El software ha sido desarrollado para Microsoft Windows NT/2000/XP.
- El software ha sido desarrollado en C++ y Microsoft Visual C++ 6.0 SP5, usando las siguiente librerías:
  - Biblioteca de Plantillas Estándar, para las estructuras dinámicas.
  - OpenGL para las vistas gráficas en 3D.
  - MFC para el manejo de ventanas.
- La comunicación TCP/IP entre los sub-sistemas distribuidos sigue la arquitectura cliente-servidor.
  - La cantidad de datos trasmitida entre IFBSimCam y SimEnc es la mínima posible para evitar exceso de tráfico en la red.

4.3. Paquetes de Software del Simulador IFB

El software del simulador IFB está dividido en los cuatro paquetes siguientes:

- **Common**: Incluye todas las clases que proporcionan funcionalidad común para las aplicaciones de IFBSimCam y SimEnc, así como las clases usadas para la transferencia de datos entre estos subsistemas.
- **IFBSimCam:** Incluye todas las clases específicas del componente IFBSimCam. Estas clases están a su vez clasificadas en dos paquetes: clases funcionales y clases del interfaz.
- **SimEnc:** Incluye todas las clases específicas de la versión anterior de SimEnc. También incluye métodos necesarios para establecer la comunicación con IFBSimCam.

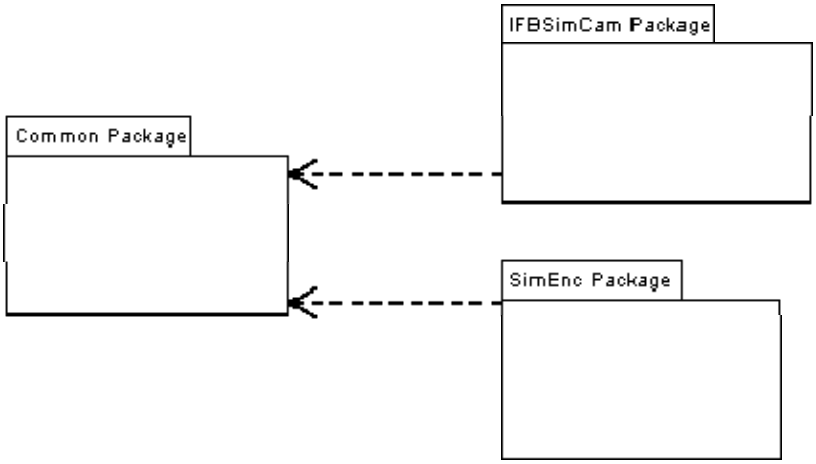


FIGURA: PAQUETES DE SOFTWARE DEL SIMULADOR IFB

4.3.1. Paquete Common

4.3.1.1. Sub-paquetes

Este paquete está subdividido en los siguientes sub-paquetes:

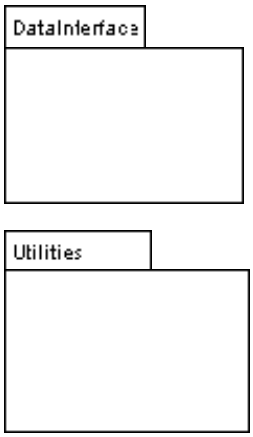


FIGURA: COMMON/SUB-PAQUETES (1)

4.3.1.1.1. Sub-paquete Data Interface

Varias clases destinadas a la transferencia de datos de configuración comunes entre los subsistemas cuando se inicia la conexión. Estos datos se almacenan por emisor y receptor y se utilizan desde ese momento para simular el comportamiento de IFB.

La principal característica de las clases de este paquete es que todos ellos deben aplicar serializar y deserializar funciones para transmitir los datos. Las clases incluyen en este paquete están representados en el diagrama de clases siguiente:

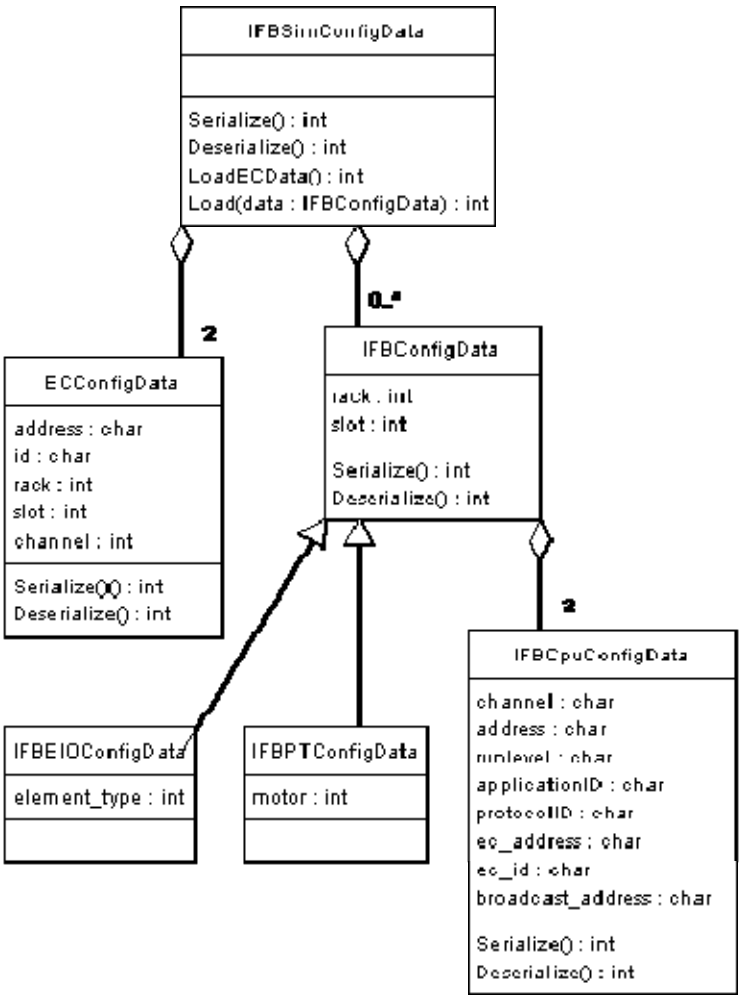


FIGURA: COMMON/ SUB-PAQUETES (2)

4.3.1.1.2. Clase IFBSimConfigData

Propósito: Mantener y Transmitir (de SimEnc a IFBSimCam) información acerca de la configuración estática del simulador IFB. Un objeto IFBSimConfigData contiene toda la información relevante de todos los IFBs que estén siendo simulados y todos los ECs reales a los que el simulador debe conectarse.

Atributos relevantes:

std::vector<IFBConfigData> v_ifbs_data;	Contenedor de la información de todas las IFBs
ECConfigData ec_a_data;	Datos de configuración de la conexión por canal A al EC

ECConfigData ec_b_data;	Datos de configuración de la conexión por canal B al EC

Funciones relevantes:

int LoadECData(const char* cad);	Carga información sobre los ECs reales desde un archivo de texto
int Load (IFBConfigData* pData);	Carga información sobre los IFBs
void Clear();	Borra todos los datos de la clase
int Serialize(char* cad,int& l,int size) const;	Rellena un byte string con información de la clase
int DeSerialize(char* cad,int& l,int size);	Obtiene datos de la clase de un byte string serializable

Relaciones relevantes:

Contiene dos objetos de la clase ECConfigData.

Contiene 0..36 objetos de la clase IFBConfigData.

4.3.1.1.3. Clase ECConfigData

Propósito: Mantener y transmitir (de IFB SimEnc a IFBSimCam) información acerca de la configuración estática de un EC.

Atributos relevantes:

unsigned char address;	Dirección del Protocolo *XFibuss* de EC
unsigned char id;	ID del Protocolo de EC
int rack;	Rack de EC
int slot;	Slot de EC
int channel;	Canal Can Bus conectado a este EC

Operaciones relevantes:

void Clear();	Borra todos los datos de la clase
int Serialize(char* cad,int& l,int size) const;	Rellena un byte string con información de la clase
int DeSerialize(char* cad,int& l,int size);	Obtiene datos de la clase de un byte string serializable.
void SetAddress(int r, int s, int cha);	Selecciona el rack, slot y canal y obtiene la dirección y el id a partir de estos valores.

Relaciones relevantes: Ninguna.

4.3.1.1.4. Clase IFBConfigData

Propósito: Mantener y transmitir (de SimEnc a IFBSimCam) información acerca de la configuración estática de un IFB. Es una clase base a partir de la cual otras clases como IFBPTConfigData o IFBEIOConfigData heredan.

Atributos relevantes:

int rack;	Rack de IFB
int slot;	Slot deIFB
int motor;	Índice del motor controlado por esta IFB
IFBCpuConfigData cpu_a_data;	Datos de Configuración de la CPU A
IFBCpuConfigData cpu_b_data;	Datos de Configuración de la CPU B

Operaciones relevantes:

int Serialize(char* cad,int& l,int size) const;	Rellena un byte string con información de la clase
int DeSerialize(char* cad,int& l,int size);	Obtiene datos de la clase de un byte string serializable.

Relaciones relevantes:

Contiene dos objetos de la clase IFBCpuConfigData, uno por cada IFB CPU. Toda configuración de datos de IFB contiene la configuración de datos de cada una de sus CPUs.

4.3.1.1.5. Clase IFBCpuConfigData

Propósito: Mantener y transmitir (de SimEnc a IFBSimCam) información acerca de la configuración estática de un CPU IFB.

Atributos relevantes:

unsigned char channel;	Canal Can Bus Channel al que esta CPU está conectado.
unsigned char address;	Direccion del protocolo *XFibuss* de CPU.
unsigned char id;	ID del Protocolo de CPU.
unsigned char runlevel;	Runlevel.
unsigned char mode;	Modo.
unsigned char applicationID;	ID de la aplicación .

unsigned char protocolID;	ID del Protocolo.
unsigned char ec_address;	Dirección del protocolo *XFibuss* conectado a este canal.
unsigned char ec_id;	ID del Protocolo conectado a este canal.
int response_time;	Milisegundos que debe esperar la CPU para reconocer al EC.

Operaciones relevantes:

int Serialize(char* cad,int& l,int size) const;	Rellena un byte string con información de la clase
int DeSerialize(char* cad,int& l,int size);	Obtiene datos de la clase de un byte string serializable.
void SetAddress(int r, int s, int cha);	Selecciona el rack, slot y canal y obtiene la dirección y el id a partir de estos valores.

Relaciones relevantes: Ninguna.

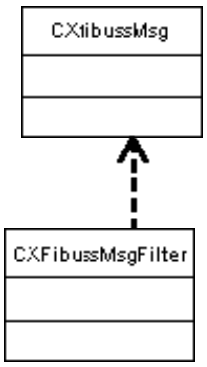


FIGURA: COMMON/SUB-PAQUETES (3)

4.3.1.1.6. Clase CXfibussMsg

Propósito: Mantener y transmitir (de IFB SimCam a IFBSimEnc) información acerca del protocolo de mensajes *XFibuss*.

Atributos relevantes:

CTimeStamp time_stamp;	Time_stamp del mensaje.
CSignature signature;	La firma del mensaje Xfibuss.
int length;	Tamaño del mensaje en bytes.
unsigned char data[MAX_LENGTH];	Datos del mensaje.

Funciones relevantes:

int Serialize(char* cad,int& l,int size) const;	Rellena un byte string con información de la clase
int DeSerialize(char* cad,int& l,int size);	Obtiene datos de la clase de un byte string serializable.
void Print(int size,char* cad,…) const;	Rellena un char string con el contenido del mensaje.
bool Match(const StringFilter& f);	Compara el contenido del mensaje con un filtro de mascara del usuario.

Relaciones relevantes:

Contiene objetos de CTimeStamp y CSignature (del paquete *Utilities*). Todo mensaje *Xfibuss* tiene su propio *time stamp* y su propia firma.

4.3.1.1.7. Clase CXfibussMsgFilter

Propósito: Mantener y transmitir (de IFB SimCam a IFBSimEnc) información acerca del protocolo de un sólo mensaje *XFibuss*.

Atributos relevantes:

int address_a;	Dirección XFibuss del canal A de los mensajes que serán modificados (0=todos).
int address_b;	Dirección XFibuss del canal B de los mensajes que serán modificados (0=todos).
StringFilter filter;	Sólo los mensajes que se correspondan con esta máscara serán modificados.
StringFilter change;	El nuevo valor, aleatorio, usuario o automático (no modifcado)
int modify_num;	Número de mensajes a modificar, 0 para indefinido

Operaciones relevantes:

int Serialize(char* cad,int& l,int size) const;	Rellena un byte string con información de la clase.
int DeSerialize(char* cad,int& l,int size);	Obtiene datos de la clase de un byte string serializable.
bool Filter(CXfibussMsg& msg);	Modifica el parámero “msg” si es igual que la configuración del filtro.

Relaciones relevantes:

Dependencia de CXfibussMsg. CXfibussMsgFilter puede modificar mensajes Xfibuss.

Contiene dos objetos StringFilter (del paquete Utilities). Uno contiene información del filtro para seleccionar que mensajes han de ser modificados. El otro, contiene información de las modificaciones que serán hechas.

4.3.1.1.8. Sub-paquete Utilities

Se han usado varias clases comunes con diferentes objetivos:

- Gráficos: representación gráfica 3D
- Comunicaciones: Comunicación socket TCP/IP.
- Verificación y producción: clases para desarrollar la verificación de archivos y el formato de salida.
- Datos: clases que contiene datos comunes, pero que no necesariamente serán trasmitidos de un subsistema a otro.

Las clases contenidas en este paquete están representadas en la figura que se muestra a continuación:

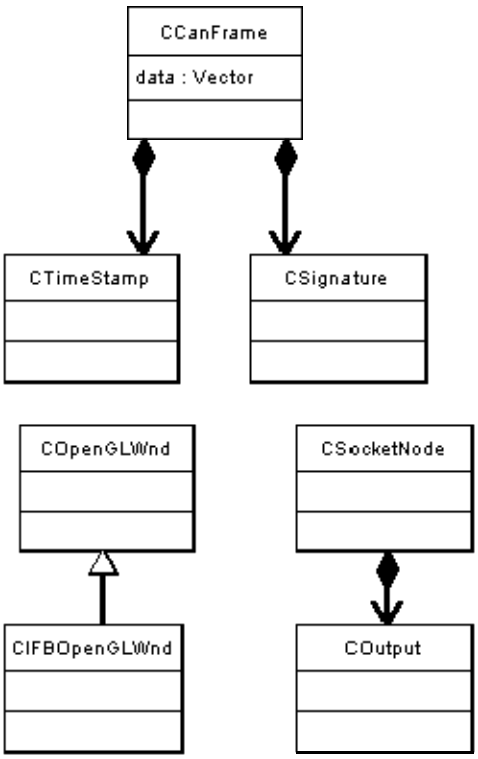


FIGURA: SUB-PAQUETE DE UTILITIES

La relación de este paquete con la Interfaz de Datos se muestra en la siguiente figura:

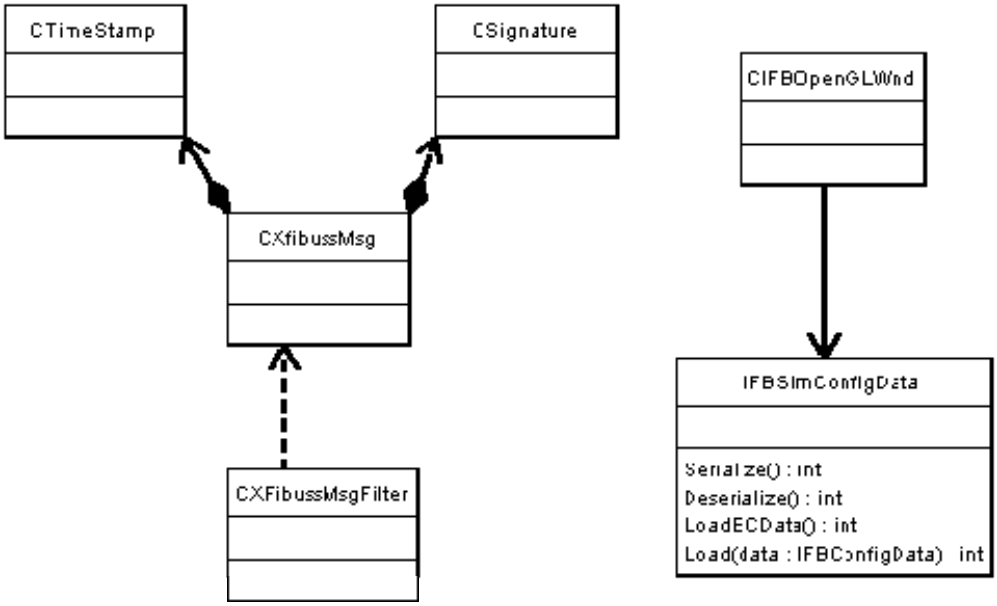


FIGURA: RELACIÓN DE PAQUETES DE UTILITIES E INTERFAZ DE DATOS

4.3.1.1.9. Clase CCanFrame

Propósito: Mantener información del marco de datos del protocolo CanBus, con varias operaciones que relacionen la información con el marco de información del protocolo XFibuss.

Relaciones Relevantes: Cada CCanFrame contiene una instancia de la clase CTimeStamp, y otra de la clase CSignature. La firma XFibuss de cada mensaje XFibuss está codificada en el arbitraje ID de los Can Frames de los marcos correspondientes a ese mensaje XFibuss, por lo que estos datos se mantienen en la clase para mayor comodidad.

4.3.1.1.10. Clase CTimeStamp

Propósito: Mantener información de un mensaje *time stamp*, en segundos y milisegundos.

Relaciones Relevantes: ninguna

4.3.1.1.11. Clase CSignature

Propósito: Mantener información de la firma de un mensaje XFibuss: origen y dirección de destino, prioridad y puerto.

Relaciones Relevantes: ninguna

4.3.1.1.12. Clase CSocketNode

Propósito: Encapsular la librería de sockets y gestionar, de manera transparente, todas las conexiones y desconexiones. Puede actuar tanto como servidor como cliente en una arquitectura cliente-servidor. Está orientada a mensajes en el sentido de que los datos son enviados y recibidos en paquetes de usuario, de modo que le usuario no tiene que separar mensajes. Se trata de una clase "alive", con un hilo interno para comunicarse sin bloquear el subproceso que usa esta clase a causa de las llamadas bloqueantes del socket, y minimizando la carga del procesador y los retardos propios de las comunicaciones.

Relaciones Relevantes: Contiene un objeto de la clase COutput para registrar y sacar la actividad interna de esta clase.

4.3.1.1.13. Clase COutput

Propósito: Clase para registrar y facilitar la salida con formato a archivo, a salida estándar y/o a cadena en memoria para ser representada por aplicaciones gráficas.
Relaciones Relevantes: ninguna

4.3.1.1.14. Clase COpenGLWnd

Propósito: Encapsula un render OpenGL en un componente de Windows (CWnd) que puede ser utilizado en cualquier aplicación basada en MFC. Esta clase gestiona todas las funciones de dibujo, cambio de punto de vista, colores, perspectiva, etc., y puede ser usada para dibujar cualquier escena.

Relaciones Relevantes: Hereda de la clase MFC CWnd.

4.3.1.1.15. Clase CIFBOpenGlWnd

Propósito: Encapsula una vista 3D OpenGL de un armario de control de simulación que contiene las IFB. Esta clase sirve para ver el estado de las IFB y las indicaciones de

las IFB reales, realizar una selección de usuarios sobre un IFB específico y producir acciones sobre un IFB específico.

Relaciones Relevantes: Hereda de la clase COpenGLWnd. Así, esta clase es una clase derivada de MFC para aplicaciones basadas en MFC. Conectado a un objeto IFBSimConfigData externo a través de un puntero, el acceso a los datos de ese objeto para representar adecuadamente la configuración del armario.

4.3.2. Paquete IFBSimCam

4.3.2.1. Sub-paquetes

Este paquete está subdividido en los siguientes sub-paquetes:

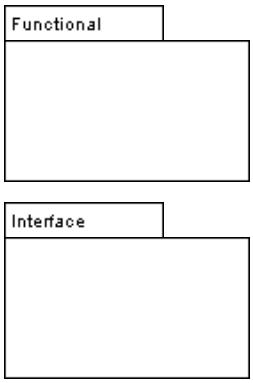


FIGURA: SUB-PAQUETES DE IFBSIMCAM

- Funcional: Varias cases diseñadas para calcular la simulación de las IFBs, y realizar otras funciones:
  - Clases de simulación de IFB, simulación IFBs, CPUs, Can Bus, protocolo XFibuss.
  - Comunicación: permitir a este paquete comunicarse con SimEnc y con ECs reales.
- Interfaz: Varias clases de interfaz de usuario usadas por distintas tareas:
  - Diálogos, representación gráfica en 3D.

4.3.2.1.1. Diagrama de Paquete de Clases

Las clases de este paquete están representadas en el diagrama siguiente:

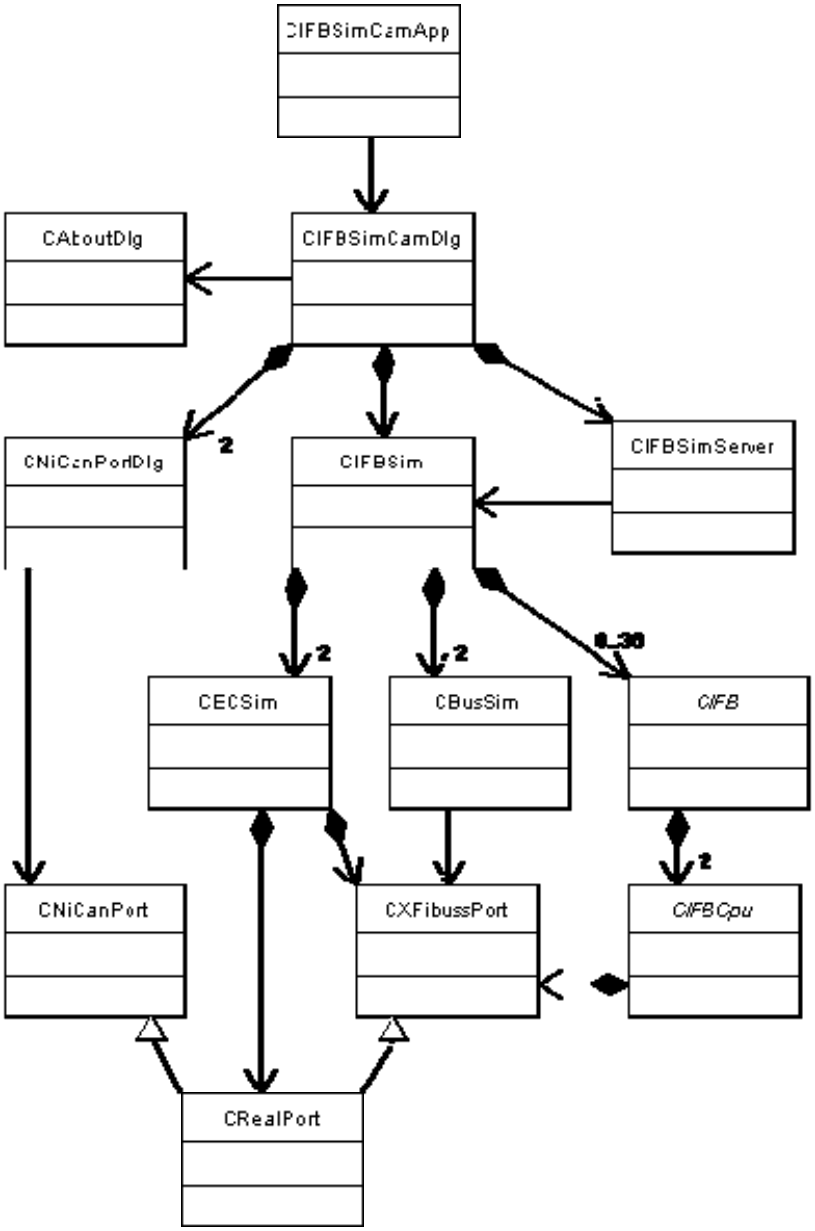


FIGURA:CLASES DEL PAQUETE IFBSIMCAM (1)

Para facilitar la comprensión de este diagrama, el siguiente diagrama nos da información sobre el paquete:

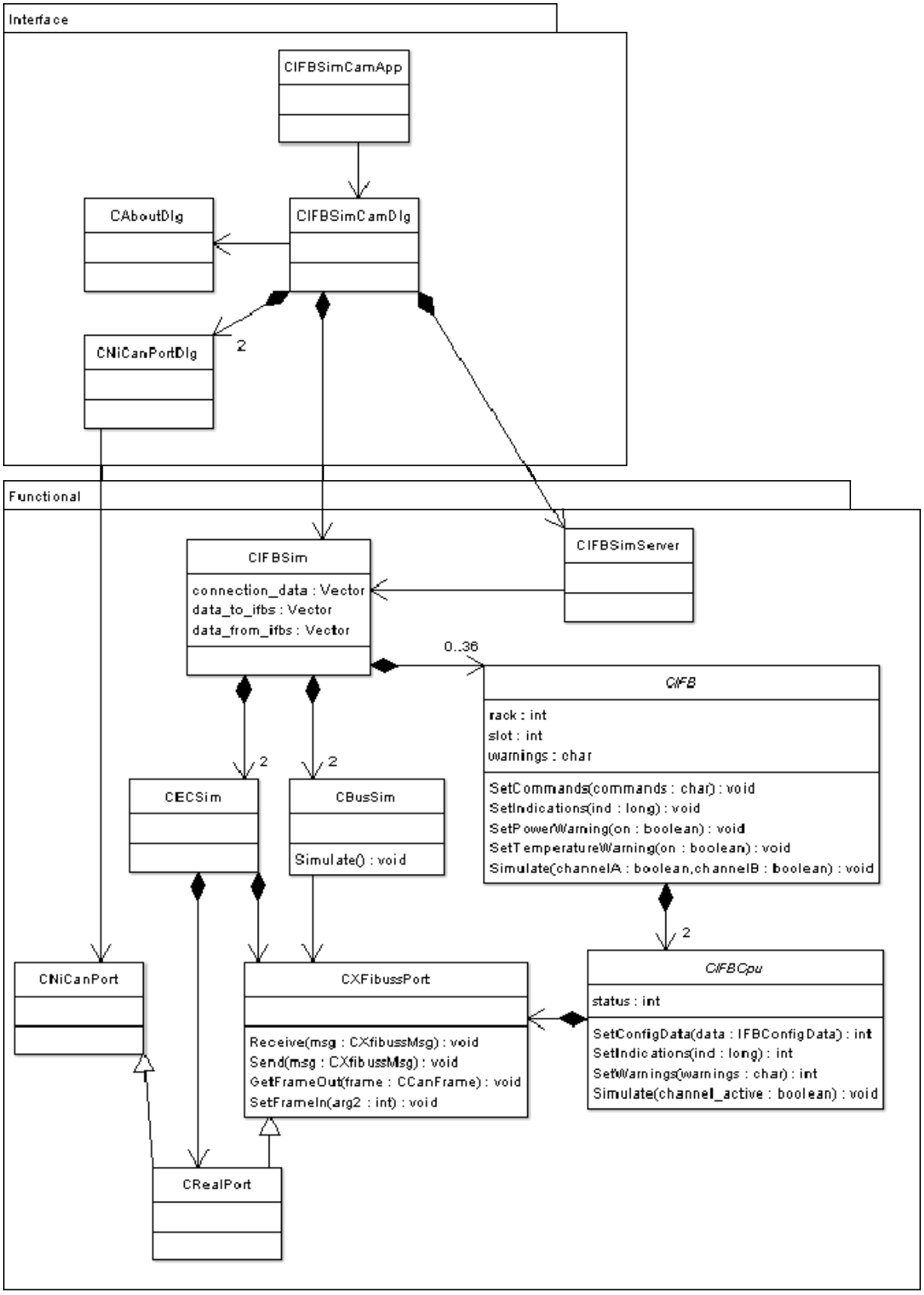


FIGURA: CLASES DEL PAQUETE IFBSIMCAM (2)

Este diagrama no ha sido mostrado por completo en las figuras anteriores. Como podemos ver, por la notación en cursiva, tanto la clase CIFBCpu como la clase CIFB han sido etiquetadas como clases abstractas. Las clases derivadas se puede observar en los siguientes sub-diagramas:

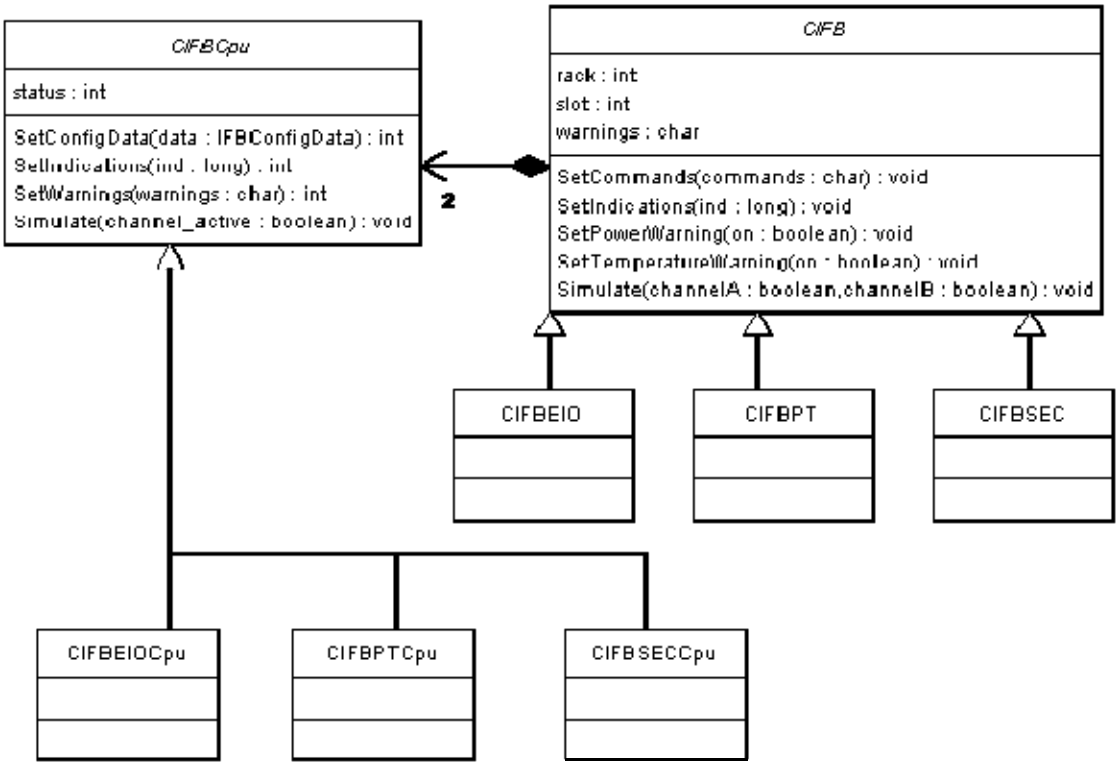


FIGURA: CLASES DEL PAQUETE IFBSIMCAM (3)

Las clases más relevantes desde el punto de vista arquitectónico se describen brevemente en las siguientes subsecciones:

4.3.2.1.2. Clase CIFBSimCamDlg

Propósito: El cuadro de diálogo principal de la aplicación.

Relaciones Relevantes:

Contiene un objeto de la clase CIFBSim, llamado ifb_simulator, que es el que realmente lleva a cabo todas las simulaciones de las IFBs.

Contiene un objeto de la clase CIFBSimServer llamado servidor que es el enlace de comunicación socket con IFB SimEnc.

Contiene dos objetos de la clase CNiCanPortDlg que permiten ver y registrar las trazas de estructura can.

4.3.2.1.3. Clase CIFBSimServer

Propósito: El punto de conexión de socket con la aplicación SimEnc IFB. Actúa como servidor en la arquitectura cliente-servidor en IFBSimCam y SimEnc.

Relaciones Relevantes:

Hereda de la clase CSocketNode (de *utilities*).

Conectar a un objeto CIFBSim a través de un puntero al objeto ifb_simulator CIFBSim contenido por CIFBSimCamDlg.

4.3.2.1.4. Clase CIFBSim

Propósito: Realizar la simulación de un armario IFB. Como el armario en sí mismo no realiza ninguna acción, actúa como el armario físico, un contenedor de otros objetos que en realidad contienen procesadores que deberán ser simulados. Su tarea principal es la gestión de la simulación y la interacción entre sus objetos secundarios, así como enrutar, filtrar y gestionar toda la información y las órdenes que vienen de CIFBSimServer, recibidas por la aplicación SimEnc.

Relaciones Relevantes:

Contención de los dos objetos de la clase CBusSim, que simulan el comportamiento físico del can bus de comunicación, uno por canal.

Contención de los dos objetos de la clase CECSim, que representan la EC real conectada a cada canal.

Contención de un vector de objetos (de 1 a 36) de la clase CIFB, que representan cada IFB en el armario.

4.3.2.1.5. Clase CECSim

Propósito: Representar la CE conectada a cada canal. Esta clase es un punto de conexión entre la simulación y la CE real, y el envío y enrutamiento de mensajes.

Relaciones Relevantes:

La contención de un objeto de clase CXfibussPort, que es la conexión a la simulación a través de CBusSim, que simula el comportamiento físico del can bus de comunicaciones.
La contención de un objeto de clase CRealPort (que hereda de CXfibussPort) que realmente es capaz de enviar y recibir datos de la CE real.

4.3.2.1.6. Clase CBusSim

Propósito: Simular la capa de comunicación física (cable) del protocolo CAN bus. Esta clase puede "conectar" varios objetos de la clase CXfibussPort, y enrutar y enviar *can frames* entre estos objetos. Aunque no está implementada actualmente, esta clase está destinada a simular completamente la transmisión y recepción de *can frames*, simulando la prioridad del bus, los retrasos en la comunicación, la retransmisión, etc. Esta clase puede ser configurada para una simulación de una comunicación completa entre los elementos (es decir, cada mensaje enviado por una IFB simulada, será enviada a otras IFB simuladas, como en la realidad), o puede trabajar en modo simple (los *can frames* no se envían entre las IFBs simuladas).

Relaciones Relevantes:

Un vector de punteros a los objetos de la clase CXfibussPort, a la que se puede enviar can frames y desde la que se puede recibir can frames.

4.3.2.1.7. Clase CIFB

Propósito: Representar un IFB físico. No realiza el procesamiento de mensajes, sino que actúa como contenedor de dos CPUs que en realidad procesan esta información. En su lugar, gestiona y contiene información para ambas CPUs, establece el vínculo entre ambas y enruta los mensajes y la información procedente de CIFBSim.

Relaciones Relevantes:

Contención de dos objetos de la clase CIFBCpu, que simulan cada una de las CPUs IFB.
NOTA: No tiene relación directa con los elementos de campo. Sólo se envía información codificada a SimEnc, que a su vez distribuye la información entre los elementos correspondientes.

4.3.2.1.8. Clase CIFBCpu

Propósito: Esta es la unidad de procesamiento del núcleo del simulador. Esta clase es capaz de enviar y recibir mensajes de XFibuss a través de un CXfibussPort contenido. Esta clase simula el comportamiento de una CPU IFB real :

- Gestiona la conexión y reconexión a ECS.
- Construye e interpreta los mensajes de protocolo FEC-IFB.
- Construye y comprueba los CRC's de los mensajes del protocolo FEC-IFB.
- Maneja el número de secuencia de mensajes del protocolo FEC-IFB.
- Puede simular la desconexión, los fallos, el reinicio o el retraso en responder a la EC.
- Simula los tiempos de espera en la conexión y las solicitudes de estado de la EC.
- Se conecta a otro subsistema, `simulando un comportamiento 2o2.
- Su reacción automática a los mensajes de protocolo FEC-IFB puede ser configurada.

Relaciones Relevantes:

Contención de un CXfibussPort, que es el punto de conexión de la CPU a la red simulada CBusSim.

Puntero al otro objeto de subsistema de la CPU CIFBCpu, simulando el enlace de comunicación entre ellos (sistema 2o2). Extendido por CIFBEIOCpu, CIFBPTCpu y CIFBSECCpu. Nunca deberían existir instancias directas de CIFBCpu, sino sólo instancias de las clases extendidas dentro de IFBSimCam.

4.3.2.1.9. Clase CNiCanPort

Propósito: Encapsular las llamadas de la librería NI Can Card para un canal Can Bus. Es el punto de conexión para el Can Bus real y la aplicación. Esta clase es capaz de enviar y recibir instancias de la clase CCanFrame (como can frames) a través del puerto real de la tarjeta can bus.

Relaciones Relevantes: ninguna

4.3.2.1.10. Clase CNiCanPortDlg

Propósito: Implementar un cuadro de diálogo que sirva para monitorizar y registrar los can frames transmitidos y recibidos por un objeto CNiCanPort.

Relaciones Relevantes: puntero a un objeto CNiCanPort cuyos can frames están siendo monitorizados.

4.3.2.1.11. Clase CXfibussPort

Propósito: Esta clase implementa el protocolo XFibuss, convirtiendo los mensajes de XFibuss en *can frames* y *can frames* en mensajes XFibuss. También se puede simular la desconexión física del cable conectado a ese puerto. Su objetivo es ser instanciado por cualquier procesador que requiera una conversión entre estas dos capas de comunicación.

Relaciones Relevantes: ninguna.

4.3.2.1.12. Clase CRealPort

Propósito: Esta clase hereda tanto de CXfibussPort, proporcionando así la funcionalidad de traducción *can frames* - Xfibuss, como de CNiCanPort, proporcionando así la capacidad de comunicación real con un EC. Esta clase es la responsable de modificar los mensajes salientes Xfibuss (simulación de IFB-> ECs reales) definidos y establecidos por el usuario, con lo que contiene un objeto CXfibussMsgFilter para ese fin. Esta clase es también la responsable del registro de las trazas de los mensajes Xfibuss que son enviados a IFB SimEnc.
Relaciones Relevantes: hereda de CXfibussPort y CNiCanPort.

Contiene un objeto CXfibussMsgFilter que define los mensajes Xfibuss modificando y filtrando máscaras y patrones.

4.3.2.1.13. Diagrama de colaboración funcional

Las clases funcionales de IFBSimCam se organizan en los objetos representados en la figura siguiente. La presentación de los objetos dentro del simulador trata de asemejarse a los objetos reales en el sistema real.

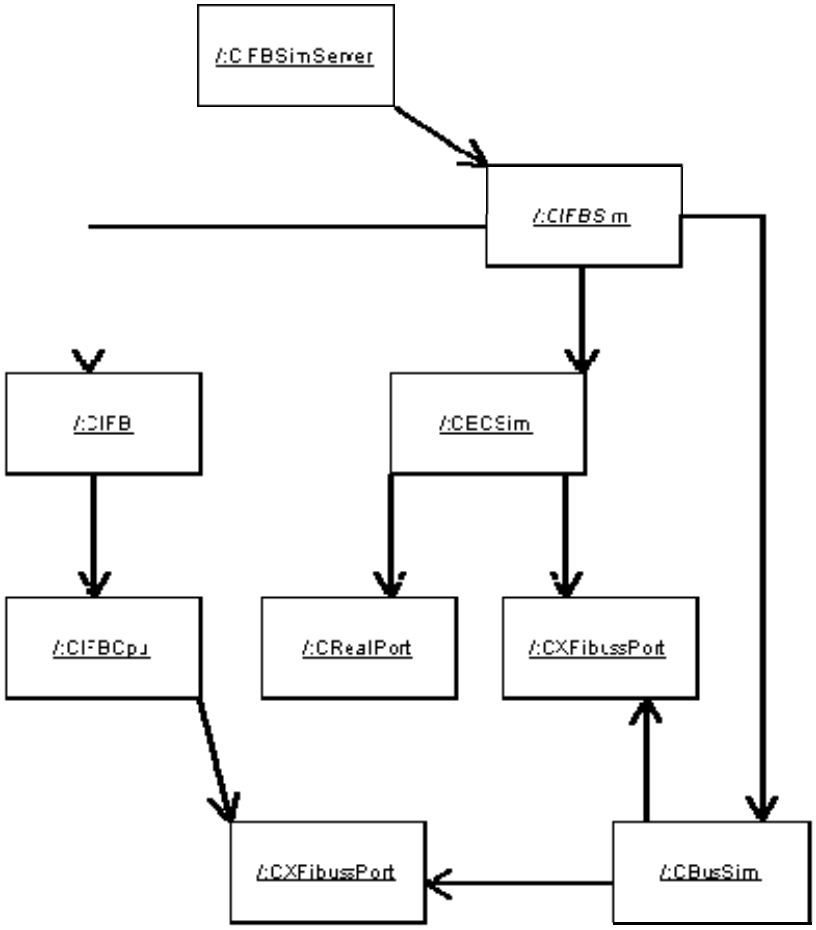


FIGURA: DIAGRAMA DE COLABORACIÓN FUNCIONAL

Del diagrama podemos destacar los siguientes puntos:

- El diseño de simulación sigue una estructura de árbol, con un flujo de control orientado de padres a hijos.
- Los objetos CIFBCpu envían y reciben mensajes XFibuss a través del objeto hijo CXfibussPort que traduce estos mensajes a *Can frames*.
- La transmisión de datos en la simulación del bus CBusSim se lleva a cabo en Can Frames, no en mensajes XFibuss.
- Los objetos CECSim reciben *can frames* de la simulación a través del objeto hijo CXfibussPort, que los convierte en mensajes XFibuss. Estos mensajes son modificados si se define un filtro de modificación del usuario, se registran las trazas para ser mostradas por IFBSimCam, y a continuación, los mensajes XFibuss mencionados se encaminan hacia el objeto hijo CRealPort. Este objeto convierte de nuevo el mensaje en *Can frames* y los entrega a la EC real.
- Los objetos CECSim reciben mensajes XFibuss de la EC a través del objeto hijo CRealPort, que recibe *can frames* y los convierte en mensajes XFibuss. Estos

mensajes se registran para que se muestren por IFBSimCam, y luego se convierten de nuevo a *can frames* por el objeto hijo CXfibussPort, para ser enviados a los IFBs simulados a través del objeto CBusSim.

- El CIFBSimServer sólo interactúa con el objeto CIFBSim, que es el responsable de manejar y entregar comandos e información a sus objetos hijos.

4.3.2.1.14. Diagrama de secuencia funcional

El flujo de información del paquete funcional de IFBSimCam puede describirse de manera general por el siguiente diagrama de secuencia:

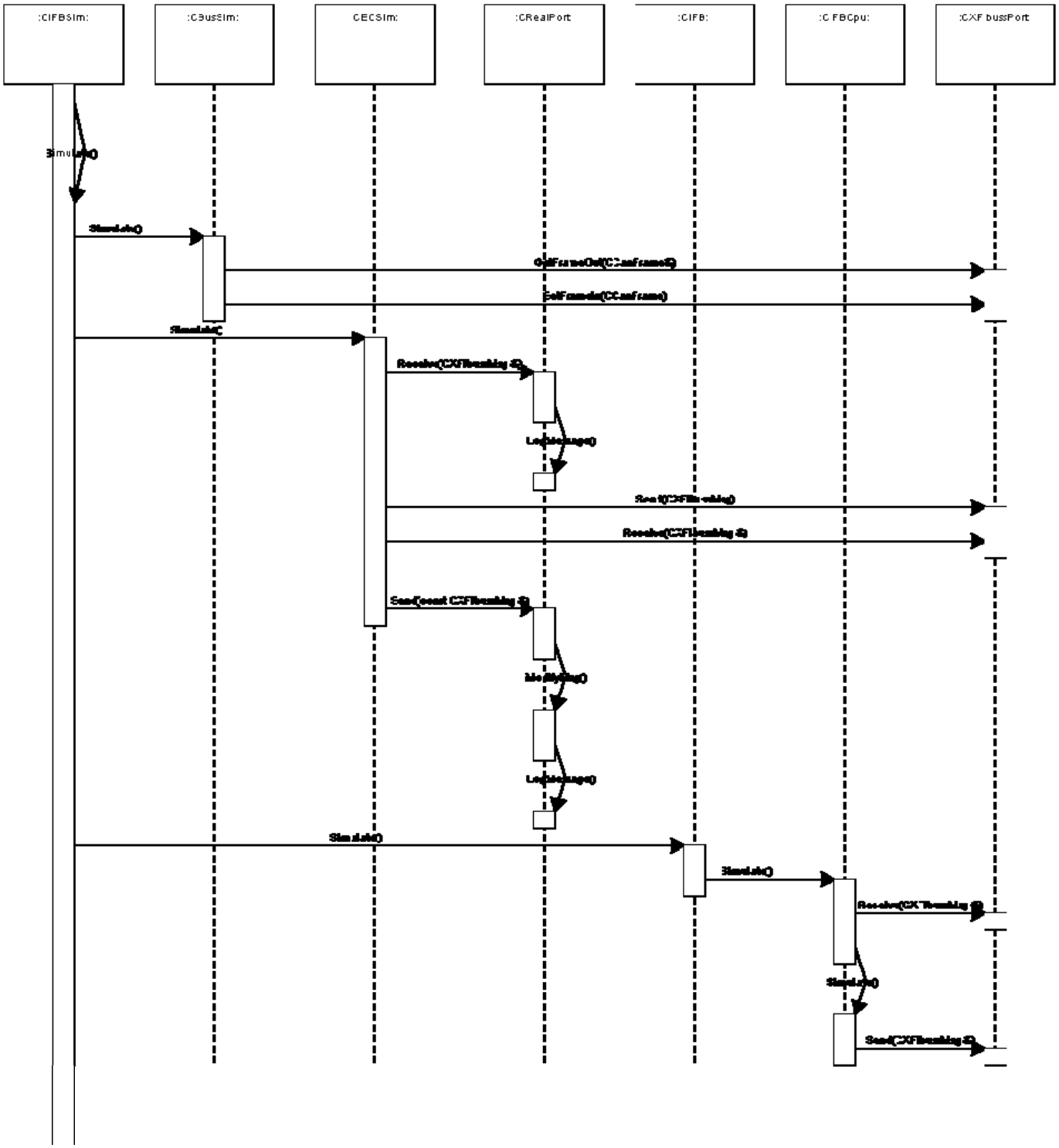


FIGURA: DIAGRAMA DE SECUENCIA FUNCIONAL

De este diagrama podemos destacar los siguientes puntos:

- El CIFBSim inicia la simulación (llamado por un hilo dedicado).
- El objeto CIFBSim ordena al objeto CBusSim simular. El objeto CBusSim a continuación, envía y recibe *can frames* entre los CXfibussPort del simulador.
- El objeto CIFBSim ordena al objeto CECSim simular. El objeto CECSim enruta los mensajes XFibuss desde el CRealPort hacia el CXfibussPort.
- El objeto CRealPort registra las trazas del mensaje y modifica el mensaje (si hay modificaciones de usuario definidas).
- El objeto CIFBSim ordena la simulación a cada CIFB, que manda simular a cada una de sus CPUs.
- En la simulación de la CIFBCpu, los mensajes XFibuss se reciben del CXfibussPort hijo, se procesan y son entregados al CXfibussPort.

4.3.2.1.15. Sub-paquete de Relación con la Interfaz de Datos (de Common)

Este paquete tiene las siguientes relaciones con las clases del paquete de la Interfaz de Datos (de Common).

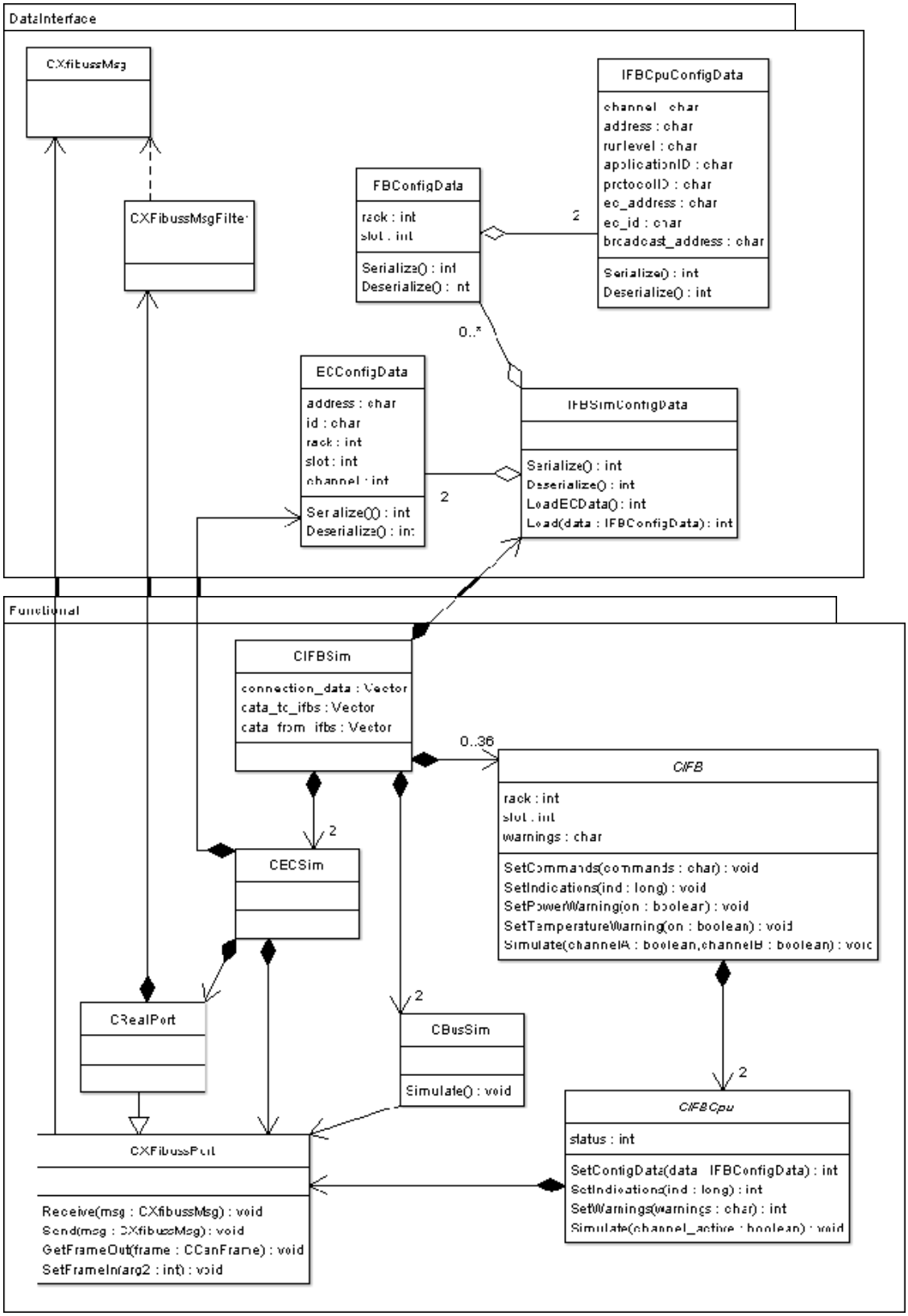


FIGURA: RELACIÓN DEL PAQUETE IFBSIMCAM CON EL SUB-PAQUETE DE LA INTERFAZ DE DATOS (COMMON)

De este diagrama podemos destacar los siguientes puntos:

- Cada uno de los componentes funcionales CIFBSim, CECSim y CIFBCpu contienen un objeto de la correspondiente clase de datos (de Data Interface).
- La relación con CXfibussMsg y el filtro CXfibussMsg ha sido ya explicada y está mejor representada en el diagrama de secuencia.
- El flujo de la información contenida en las clases Data Interfaz dentro de la estructura del simulador de IFB sigue una estructura de árbol, con orientación de padres a hijos.

4.3.2.1.16. Flujo de la Interfaz de Datos dentro de los elementos Funcionales

El flujo de información contenido en las clases Data Interfaz dentro de la estructura del simulador IFB sigue una estructura de árbol, con un flujo de control orientado de padres a hijos. El procedimiento de transmisión de los datos procedentes de SimEnc está representado en el siguiente diagrama de secuencia:

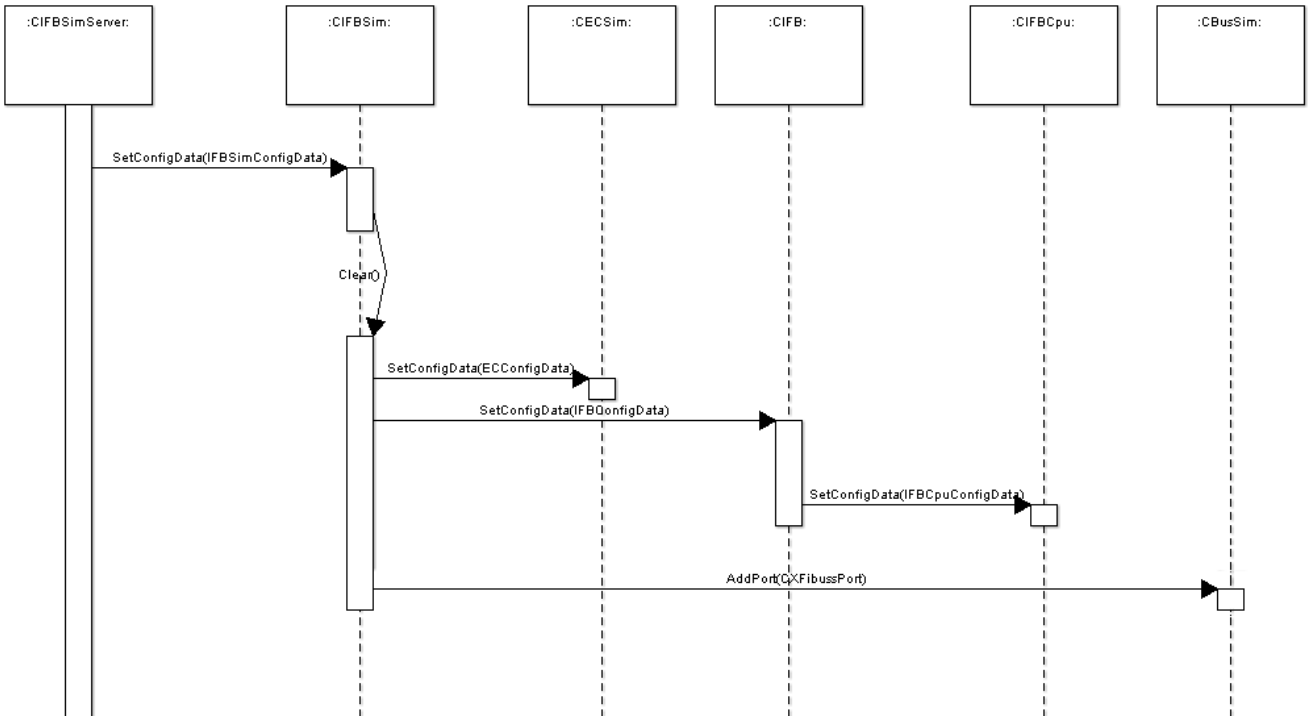


FIGURA: FLUJO DE LA INTERFAZ DE DATOS DENTRO DE LOS ELEMENTOS FUNCIONALES

De este diagrama podemos destacar los siguientes puntos:

- IFBSimConfigData se recibe en CIFBSimServer, conteniendo toda la información requerida.
- Estos datos son enviados a CIFBSim, que elimina automáticamente los datos de configuración anteriores.
- El CIFBSim enruta la información a cada uno de sus objetos hijos, enviando la ECConfigData de la IFBSimConfigData recibida a CECSim, la IFBConfigData a CIFB, que son creados dinámicamente.
- Todo CIFB enruta la CIFBCpuConfigData a sus CIFBCpus.
- Dependiendo de la naturaleza de cada IFBs, serán usadas distintas subclases de IFBConfigData. Sin embargo, para el propósito de este análisis sólo se considera la clase padre.
- CIFBSim conecta los objetos CBusSim a los objetos hijos CIFBCpus recién creados del CXfibussPort.

4.3.2.1.17. Relación con el sub-paquete Utilities (de Common)

Este paquete tiene las siguientes relaciones con las clases del sub-paquete Utilities (de Common):

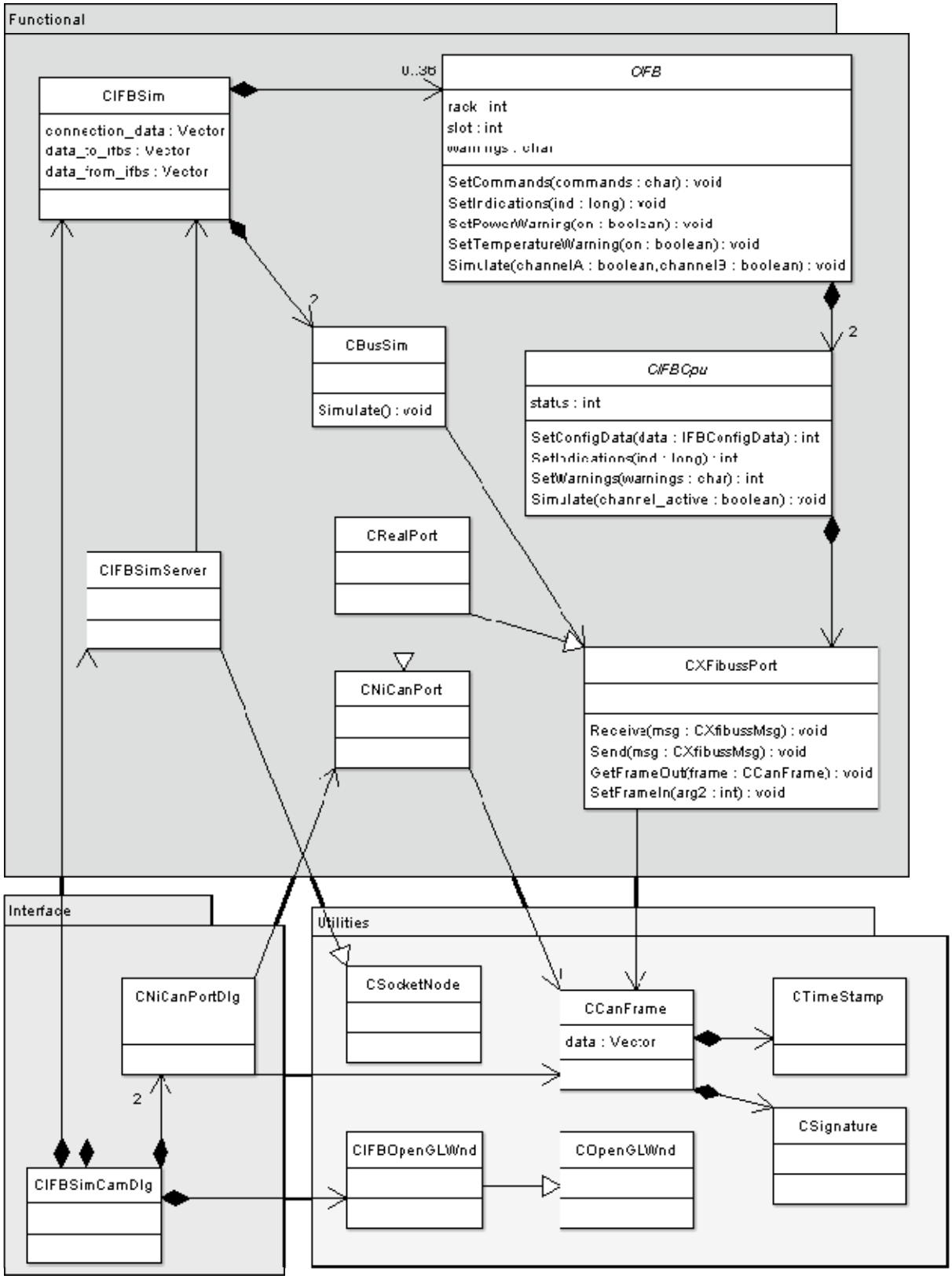


FIGURA: RELACIÓN DEL PAQUETE IFBSIMCAM CON EL SUB-PAQUETE UTILITIES (COMMON) (1)

De este diagrama podemos destacar los siguientes puntos:

- CIFBSimCamDlg contiene un CIFBOpenGLWnd que representa gráficamente la información del simulador.
- CIFBSimServer hereda de CSocketNode, por lo que contiene un objeto COutput para propósitos de registro.
- La CIFBCpu también contiene un objeto COutput para propósitos de registro.
- CXfibussPort y CNiCanPort son capaces de enviar y recibir objetos CCanFrame (dentro del simulador y de los EC reales, respectivamente).

Estas relaciones pueden ser traducidas en el siguiente diagrama de colaboración:

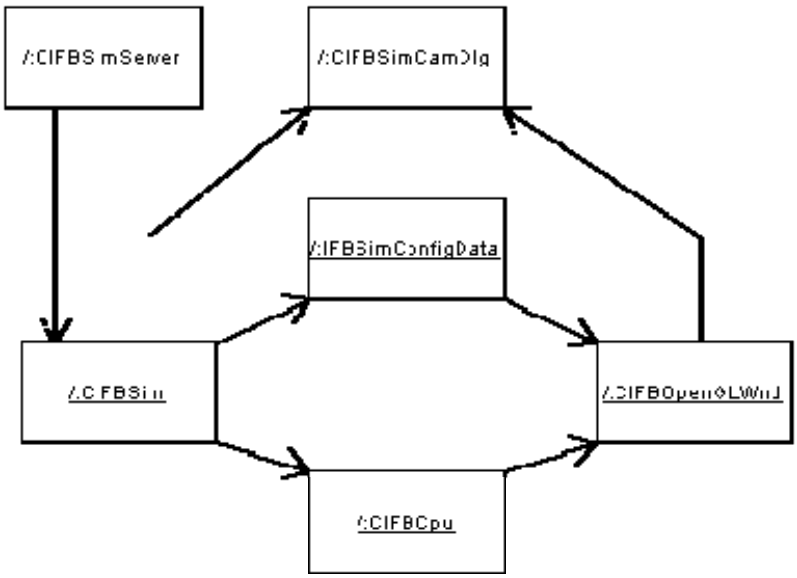


FIGURA: COLABORACIÓN DEL PAQUETE IFBSIMCAM CON EL SUB-PAQUETE UTILITIES (COMMON) (2)

De este diagrama podemos destacar:

- CIFBOpenGLWnd extrae información de las clases Data Interface contenidas y gestionadas por CIFBSim para propósitos de representación gráfica.
-

4.3.3. Paquete SimEnc

El paquete SimEnc sigue siendo prácticamente el mismo que el de Simenc_3.01, y su software de diseño interior no se describe aquí. Sólo las relaciones de SimEnc con el paquete de IFBSimCam, además de tres nuevas clases necesarias.

4.3.3.1. Diagrama de Clases

La relación de SimEnc 4.02 con el paquete IFBSimCam pretende permitir una comunicación sencilla entre IFBSimCam y SimEnc 4.02. El diagrama de clases que representa estas relaciones es el siguiente, con las siguientes características:

- La comunicación entre IFBSimCam y SimEnc se logra mediante la contención de la clase CIFBSimThread en SimEnc 4.02.
- Un *gannu_FB_element* de clase se utiliza para representar el estado del punto en la arquitectura SimEnc. Esta clase accede a los motores situados en CIFBSimEncDlg para llevar a cabo la simulación del motor.
- Un *gannu_vgr_fb* de clase se utiliza para representar al gannu_FB_element en la vista principal de Simenc.

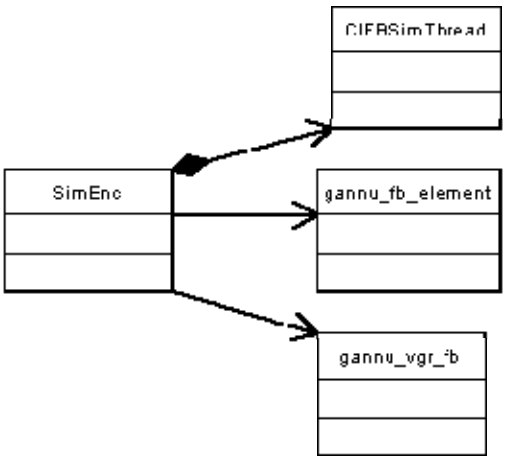


FIGURA:DIAGRAMA DE CLASE DEL PAQUETE SIMENC

4.3.4. Relación entre IFBSimCam y SimEnc

IFBSimCam y SimEnc están conectados a través de CIFBSimServer y CIFBSimThread respectivamente. La primera hereda de CSocketNode, mientras que la segunda es una clase muy similar a otras clases anteriores de SimEnc (SimTrainThread, CReaderThread), que permite una mejor integración de cliente. Todos los datos de configuración, las trazas de mensajes y los comandos de usuario se transmiten a través de una única instancia de estas clases, una por cada aplicación.

El panorama general del flujo de las operaciones y la información entre ambos sistemas está representado en el siguiente diagrama de secuencia:

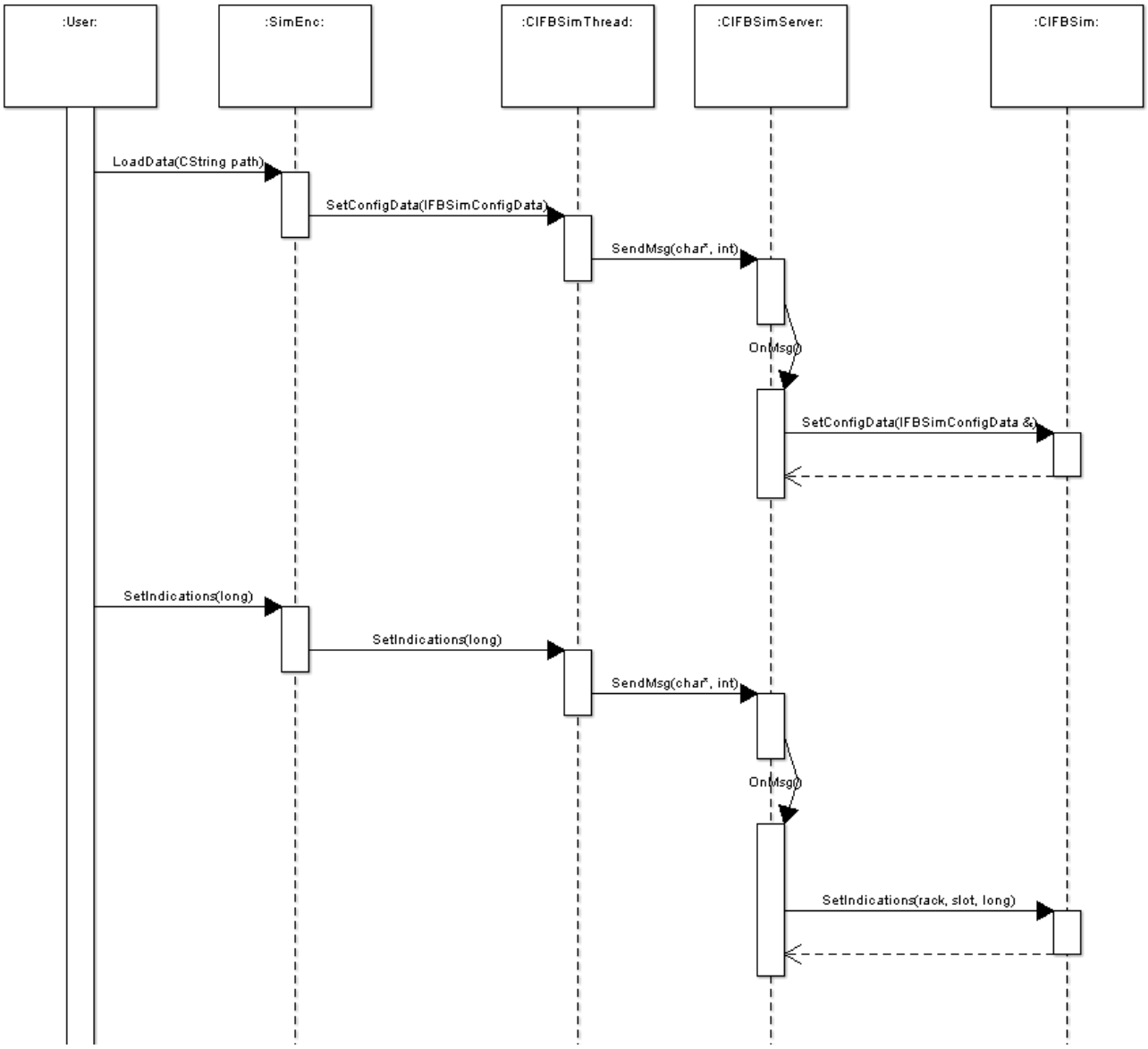


FIGURA: RELACIÓN IFBSIMCAM – SIMENC (1)

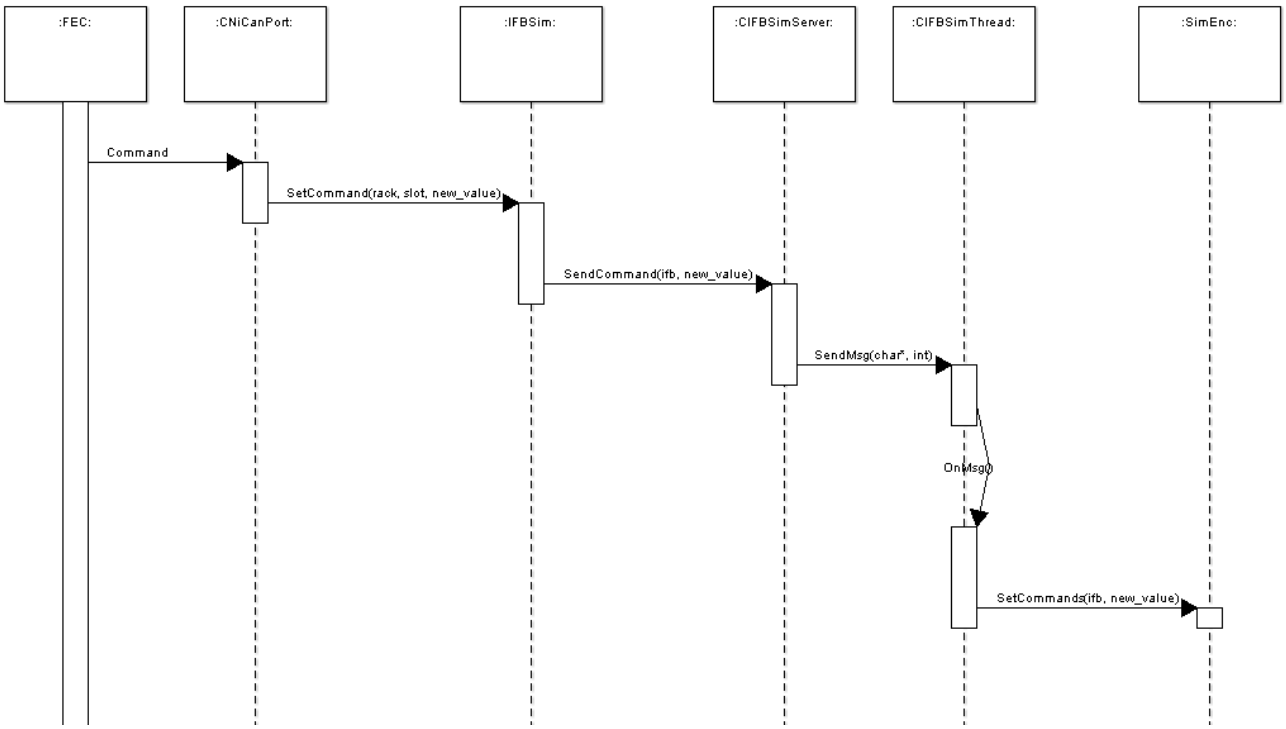


FIGURA:RELACIÓN IFBSIMCAM – SIMENC (2)

Sobre estos esquemas podemos hacer las siguientes observaciones:

- Nunca hay mensajes de confirmación intercambiados a nivel de aplicación. La comunicación se basa en el protocolo TCP/IP. Los mecanismos de alerta por desconexión de socket son usados para ese fin.
- El envío de mensajes se realiza de forma asíncrona, lo que reduce la sobrecarga de la red a un mínimo. Sólo los mensajes necesarios se envían. No se utilizan mensajes innecesarios.
- La comunicación con IFBSimCam está estrechamente integrada en SimEnc 4.02, por medio de CIFBSimThread, una clase muy similar a las clases anteriormente utilizadas en SimEnc para las comunicaciones.

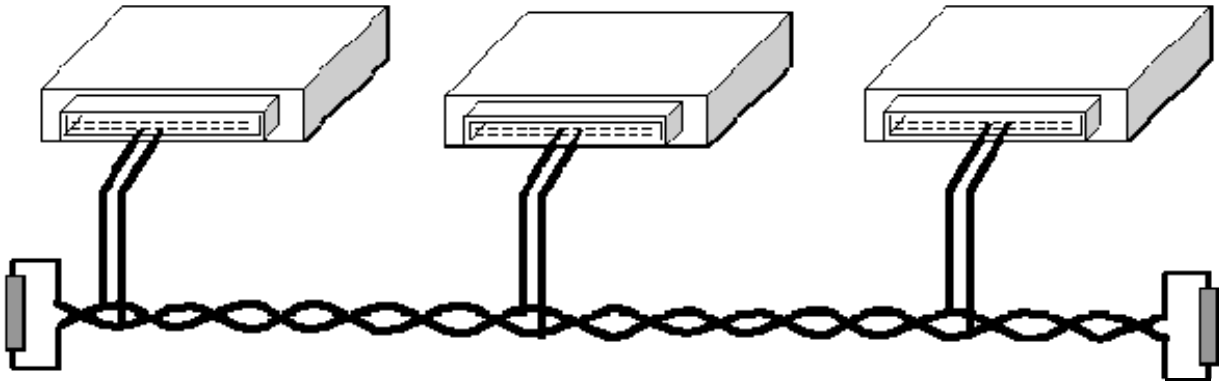
5. Mecanismo de comunicación

5.1. CAN BUS

Can significa Controller Area Network (Red de área de control) y Bus, en informática, se entiende como un elemento que permite transportar una gran cantidad de información.

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.



5.1.1. Principales características del protocolo CAN

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

Prioridad de mensajes.

Garantía de tiempos de latencia.

Flexibilidad en la configuración.

Recepción por multidifusión (*multicast*) con sincronización de tiempos.

Sistema robusto en cuanto a consistencia de datos.

Sistema multimaestro.

Detección y señalización de errores.

Retransmisión automática de tramas erróneas

Distinción entre errores temporales y fallas permanentes de los nodos de la red, y desconexión autónoma de nodos defectuosos.

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

El flujo de la información que se transmite gracias a este protocolo es:

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

- Uno de esos campos actúa de identificador del tipo de dato que se transporta, de la unidad de mando que lo trasmite y de la prioridad para trasmitirlo respecto a otros. El mensaje no va direccionado a ninguna unidad de mando en concreto, cada una de ellas reconocerá mediante este identificador si el mensaje le interesa o no.

- Todas las unidades de mando pueden ser trasmisoras y receptoras, y la cantidad de las mismas abonadas al sistema puede ser variable (dentro de unos límites).

- Si la situación lo exige, una unidad de mando puede solicitar a otra una determinada información mediante uno de los campos del mensaje (trama remota o RDR).

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

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

5.1.2. Capas de Can-Bus

CAN es un protocolo de comunicaciones serie que soporta control distribuido en tiempo real con un alto nivel de seguridad y multiplexación.

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

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

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

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

El método de acceso al medio utilizado es el de Acceso Múltiple por Detección de Portadora, con Detección de Colisiones y Arbitraje por Prioridad de Mensaje (CSMA/CD+AMP, *Carrier Sense Multiple Access with Collision Detection and Arbitration Message Priority*). De acuerdo con este método, los nodos en la red que necesitan transmitir información deben esperar a que el bus esté libre (detección de portadora); cuando se cumple esta condición, dichos nodos transmiten un bit de inicio

(acceso múltiple). Cada nodo lee el bus bit a bit durante la transmisión de la trama y comparan el valor transmitido con el valor recibido; mientras los valores sean idénticos, el nodo continúa con la transmisión; si se detecta una diferencia en los valores de los bits, se lleva a cabo el mecanismo de arbitraje.

CAN establece dos formatos de tramas de datos (*data frame*) que difieren en la longitud del campo del identificador, las tramas estándares (*standard frame*) con un identificador de 11 bits definidas en la especificación CAN 2.0A, y las tramas extendidas (*extended frame*) con un identificador de 29 bits definidas en la especificación CAN 2.0B.

Para la transmisión y control de mensajes CAN, se definen cuatro tipos de tramas: de datos, remota (*remote frame*), de error (*error frame*) y de sobrecarga (*overload frame*). Las tramas remotas también se establecen en ambos formatos, estándar y extendido, y tanto las tramas de datos como las remotas se separan de tramas precedentes mediante espacios entre tramas (*interframe space*).

En cuanto a la detección y manejo de errores, un controlador CAN cuenta con la capacidad de detectar y manejar los errores que surjan en una red. Todo error detectado por un nodo, se notifica inmediatamente al resto de los nodos.

Capa de supervisor: La sustitución del cableado convencional por un sistema de bus serie presenta el problema de que un nodo defectuoso puede bloquear el funcionamiento del sistema completo. Cada nodo activo transmite una bandera de error cuando detecta algún tipo de error y puede ocasionar que un nodo defectuoso pueda acaparar el medio físico. Para eliminar este riesgo el protocolo CAN define un mecanismo autónomo para detectar y desconectar un nodo defectuoso del bus, dicho mecanismo se conoce como aislamiento de fallos.

Capa de aplicación: Existen diferentes estándares que definen la capa de aplicación; algunos son muy específicos y están relacionados con sus campos de aplicación. Entre las capas de aplicación más utilizadas cabe mencionar CAL, CANopen, DeviceNet, SDS (*Smart Distributed System*), OSEK, CANKingdom.

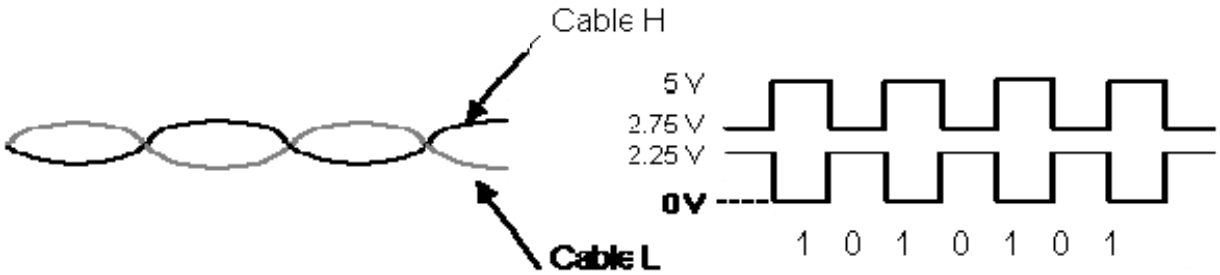
5.1.3. Elementos que componen el sistema Can-Bus

5.1.3.1. Cables

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

En un cable los valores de tensión oscilan entre 0V y 2.25V, por lo que se denomina cable L (Low) y en el otro, el cable H (High) lo hacen entre 2.75V. y 5V. En caso de que se interrumpa la línea H o que se derive a masa, el sistema trabajará con la señal de Low con respecto a masa, en el caso de que se interrumpa la línea L, ocurrirá lo contrario. Esta situación permite que el sistema siga trabajando con uno de los cables cortados o comunicados a masa, incluso con ambos comunicados también sería posible el funcionamiento, quedando fuera de servicio solamente cuando ambos cables se cortan.

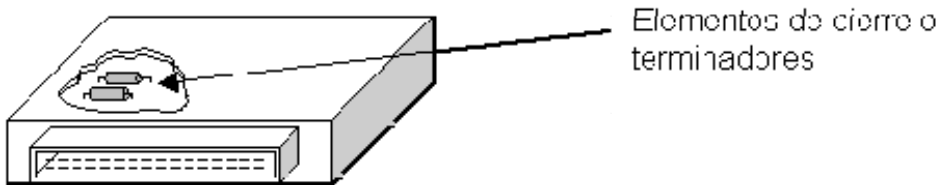
Es importante tener en cuenta que el trenzado entre ambas líneas sirve para anular los campos magnéticos, por lo que no se debe modificar en ningún caso ni el paso ni la longitud de dichos cables.



5.1.3.2. Elemento de cierre o terminador

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

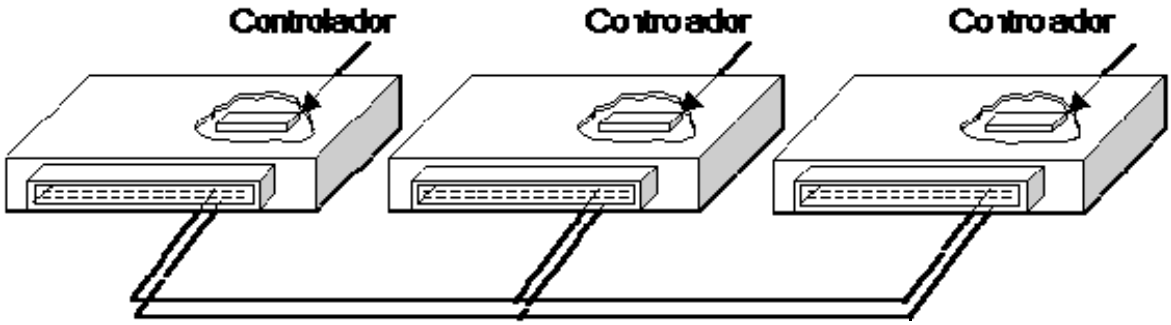
Estas resistencias están alojadas en el interior de algunas de las unidades de control del sistema por cuestiones de economía y seguridad de funcionamiento



5.1.3.3. Controlador

Es el elemento encargado de la comunicación entre el microprocesador de la unidad de control y el trasmisor-receptor. Trabaja acondicionando la información que entra y sale entre ambos componentes.

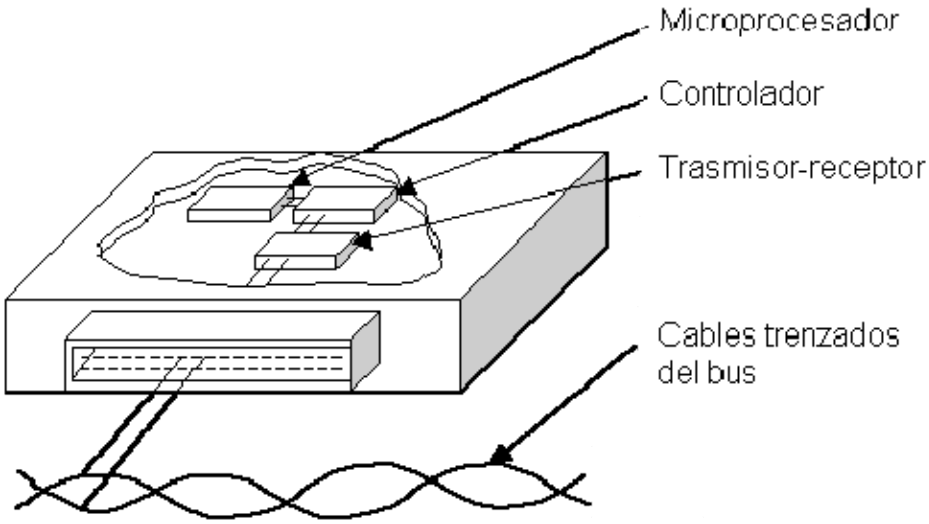
El controlador está situado en la unidad de control, por lo que existen tantos como unidades estén conectadas al sistema. Este elemento trabaja con niveles de tensión muy bajos y es el que determina la velocidad de trasmisión de los mensajes, que será más o menos elevada según el compromiso del sistema. Así, en la línea de Can-Bus del motor-frenos-cambio automático es de 500 K baudios, y en los sistema de confort de 62.5 K baudios. Este elemento también interviene en la necesaria sincronización entre las diferentes unidades de mando para la correcta emisión y recepción de los mensajes.



5.1.3.4. Transmisor / Receptor

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

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



5.1.4. Funcionamiento del sistema Can-Bus

Las unidades de mando que se conectan al sistema Can-Bus son las que necesitan compartir información, pertenezcan o no a un mismo sistema.

El sistema Can-Bus está orientado hacía el mensaje y no al destinatario. La información en la línea es trasmitida en forma de mensajes estructurados en la que una parte del mismo es un identificador que indica la clase de dato que contiene. Todas las unidades de control reciben el mensaje, lo filtran y solo lo emplean las que necesitan dicho dato. Naturalmente, la totalidad de unidades de control abonadas al sistema son capaces tanto de introducir como de recoger mensajes de la línea. Cuando el bus está libre cualquier unidad conectada puede empezar a trasmitir un nuevo mensaje.

En el caso de que una o varias unidades pretendan introducir un mensaje al mismo tiempo, lo hará la que tenga una mayor prioridad. Esta prioridad viene indicada por el identificador.

El proceso de trasmisión de datos se desarrolla siguiendo un ciclo de varias fases:

Suministro de datos: Una unidad de mando recibe información de los sensores que tiene asociados.

Su microprocesador pasa la información al controlador donde es gestionada y acondicionada para a su vez ser pasada al trasmisor-receptor donde se transforma en señales eléctricas.

Trasmisión de datos: El controlador de dicha unidad transfiere los datos y su identificador junto con la petición de inicio de trasmisión, asumiendo la responsabilidad de que el mensaje sea correctamente trasmitido a todas las unidades de mando

asociadas. Para trasmitir el mensaje ha tenido que encontrar el bus libre, y en caso de colisión con otra unidad de mando intentando trasmitir simultáneamente, tener una prioridad mayor. A partir del momento en que esto ocurre, el resto de unidades de mando se convierten en receptoras.

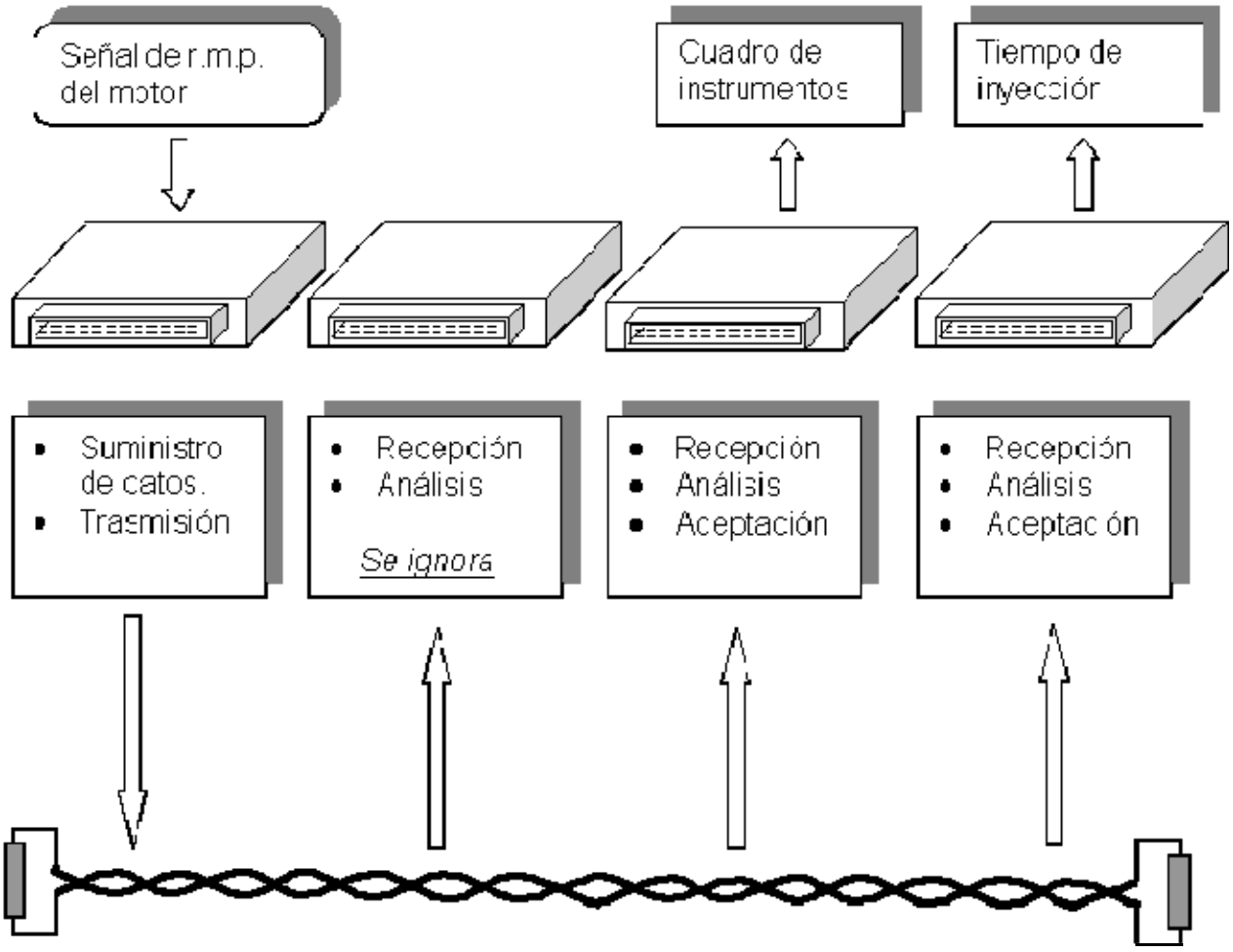
Recepción del mensaje: Cuando la totalidad de las unidades de mando reciben el mensaje, verifican el identificador para determinar si el mensaje va a ser utilizado por ellas. Las unidades de mando que necesiten los datos del mensaje lo procesan, si no lo necesitan, el mensaje es ignorado.

El sistema Can-Bus dispone de mecanismos para detectar errores en la trasmisión de mensajes, de forma que todos los receptores realizan un chequeo del mensaje analizando una parte del mismo, llamado campo CRC. Otros mecanismos de control se aplican en las unidades emisoras que monitorizan el nivel del bus, la presencia de campos de formato fijo en el mensaje (verificación de la trama), análisis estadísticos por parte de las unidades de mando de sus propios fallos etc.

Estas medidas hacen que las probabilidades de error en la emisión y recepción de mensajes sean muy bajas, por lo que es un sistema extraordinariamente seguro.

El planteamiento del Can-Bus, como puede deducirse, permite disminuir notablemente el cableado, puesto que si una unidad de mando dispone de una información, esta puede ser utilizada por el resto de unidades de mando sin que sea necesario que cada una de ellas reciba la información de dicho sensor.

Otra ventaja obvia es que las funciones pueden ser repartidas entre distintas unidades de mando, y que incrementar las funciones de las mismas no presupone un coste adicional excesivo.



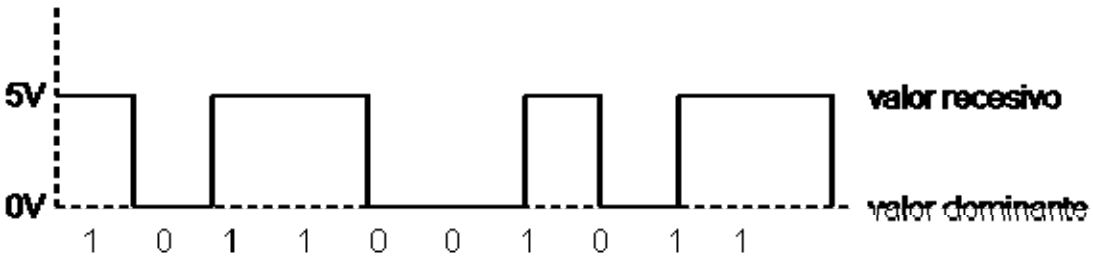
5.1.5. Estructura del mensaje

El mensaje es una sucesión de “0” y “1”, que como se explicaba al principio, están representados por diferentes niveles de tensión en los cables del Can-Bus y que se denominan “bit”.

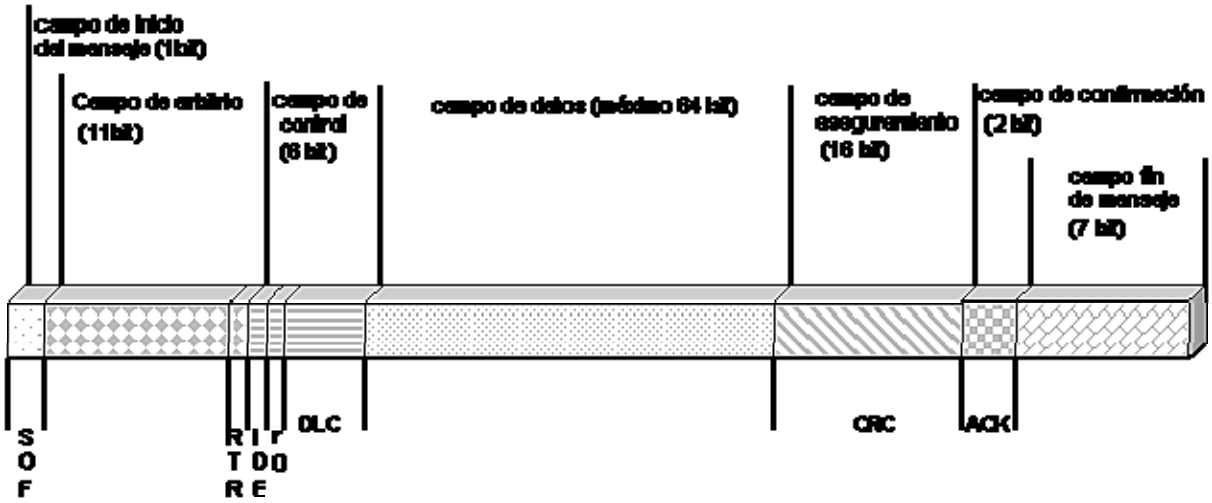
El mensaje tiene una serie de campos de diferente tamaño (número de bits) que permiten llevar a cabo el proceso de comunicación entre las unidades de mando según el protocolo definido por Bosch para el Can-Bus, que facilitan desde identificar a la unidad de mando, como indicar el principio y el final del mensaje, mostrar los datos, permitir distintos controles etc.

Los mensajes son introducidos en la línea con una cadencia que oscila entre los 7 y los 20 milisegundos dependiendo de la velocidad del área y de la unidad de mando que los introduce.

Ejemplo de cómo se escribe un mensaje:



Estructura del mensaje estándar:



Campo de inicio del mensaje: El mensaje se inicia con un bit dominante, cuyo flanco descendente es utilizado por las unidades de mando para sincronizarse entre sí.

Campo de arbitrio: Los 11 bit de este campo se emplean como identificador que permite reconocer a las unidades de mando la prioridad del mensaje. Cuanto más bajo

sea el valor del identificador más alta es la prioridad, y por lo tanto determina el orden en que van a ser introducidos los mensajes en la línea.

El bit RTR indica si el mensaje contiene datos (RTR=0) o si se trata de una trama remota sin datos (RTR=1). Una trama de datos siempre tiene una prioridad más alta que una trama remota.

La trama remota se emplea para solicitar datos a otras unidades de mando o bien porque se necesitan o para realizar un chequeo.

Campo de control: Este campo informa sobre las características del campo de datos. El bit IDE indica cuando es un “0” que se trata de una trama estándar y cuando es un “1” que es una trama extendida. Los cuatro bit que componen el campo DLC indican el número de bytes contenido en el campo de datos.

La diferencia entre una trama estandar y una trama extendida es que la primera tiene 11 bits y la segunda 29 bits. Ambas tramas pueden coexistir eventualmente, y la razón de su presencia es la existencia de dos versiones de CAN.

Campo de datos: En este campo aparece la información del mensaje con los datos que la unidad de mando correspondiente introduce en la linea Can-Bus. Puede contener entre 0 y 8 bytes (de 0 a 64 bit).

Campo de aseguramiento (CRC): Este campo tiene una longitud de 16 bit y es utilizado para la detección de errores por los 15 primeros, mientras el último siempre es un bit recesivo (1) que delimita el campo CRC.

Campo de confirmación (ACK): El campo ACK está compuesto por dos bit que son siempre trasmitidos como recesivos (1). Todas las unidades de mando que reciben el mismo CRC modifican el primer bit del campo ACK por uno dominante (0), de forma que la unidad de mando que está todavía trasmitiendo reconoce que al menos alguna unidad de mando ha recibido un mensaje escrito correctamente. De no ser así, la unidad de mando trasmisora interpreta que su mensaje presenta un error.

Campo de final de mensaje (EOF): Este campo indica el final del mensaje con una cadena de 7 bits recesivos.

Puede ocurrir que en determinados mensajes se produzcan largas cadenas de ceros o unos, y que esto provoque una pérdida de sincronización entre unidades de mando. El protocolo CAN resuelve esta situación insertando un bit de diferente polaridad cada cinco bits iguales: cada cinco “0” se inserta un “1” y viceversa. La unidad de mando que utiliza el mensaje, descarta un bit posterior a cinco bits iguales. Estos bits reciben el nombre de bit stuffing.

5.1.6. Diagnóstico del Can-Bus

Los sistemas de seguridad que incorpora el Can-Bus permiten que las probabilidades de fallo en el proceso de comunicación sean muy bajas, pero sigue siendo posible que cables, contactos y las propias unidades de mando presenten alguna disfunción.

Para el análisis de una avería, se debe tener presente que una unidad de mando averiada abonada al Can-Bus en ningún caso impide que el sistema trabaje con normalidad. Lógicamente no será posible llevar a cabo las funciones que implican el uso de información que proporciona la unidad averiada, pero sí todas las demás.

Una alternativa es emplear el programa informático CANAlyzer (Vector Informatik GmbH) con el ordenador con la conexión adecuada. Este programa permite visualizar el tráfico de datos en el Can-Bus, indica el contenido de los mensajes y realiza la estadística de mensajes, rendimiento y fallos.

Probablemente, la herramienta más adecuada y asequible sea el osciloscopio digital con dos canales, memoria y un ancho de banda de 20 MHz. (FLUKE, MIAC etc.) con el que se pueden visualizar perfectamente los mensajes utilizando una base de tiempos de 100 microsegundos y una base de tensión de 5V. En este caso, se debe tener en cuenta que los bits stuff (el que se añade después de cinco bits iguales) deben ser eliminados.

5.2. CAN BUS en IFB SIMCAM

En la aplicación desarrollada se ha utilizado como soporte hardware de CAN BUS una tarjeta NI **CAN Bus Controller PXI/PCI 3052** que dispone de dos puertos CANBUS versión 2.0 que se han programado a 250 Kbaudios y que permite un identificador extendido de 32 bits en lugar de los 11 de la primera versión. Los equipos de control de Thales S.A. incorporan hardware de diseño propio para las conexiones CANBUS.

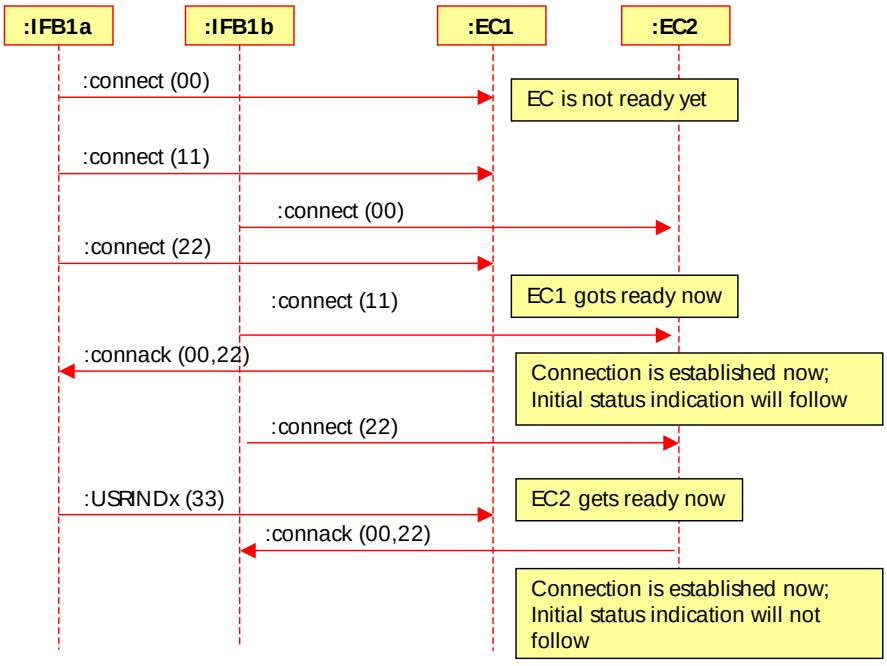
El nivel de aplicación propuesto por Thales sobre el protocolo estándar CANBUS resulta muy sofisticado. Para permitir mensajes de longitud variable y superior a los 8 bytes del mensaje CANBUS se usa un byte de datos para implementar una secuencia de mensajes por canal y operación.

La conexión y desconexión resultan algo complejas para garantizar la disponibilidad de los elementos y su respuesta en tiempo a las ordenes de control. Se muestran a continuación algunos escenarios y el intercambio de mensajes.

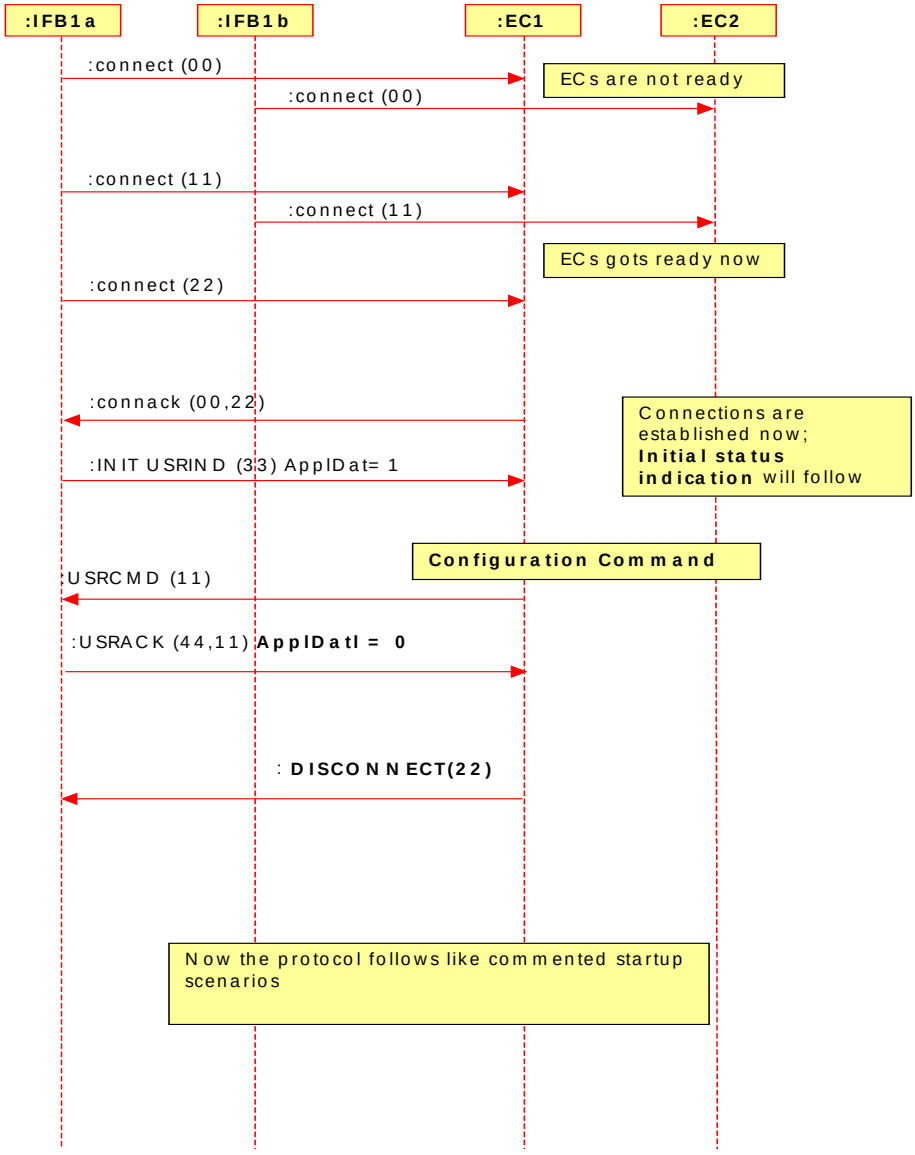
Junto a las especificaciones de las interfaces del apartado 2.6 también se facilitó un documento sobre el protocolo de comunicación específico para nuestra aplicación en el sistema.

A modo de ejemplo, se muestran a continuación algunos escenarios y el intercambio de mensajes.

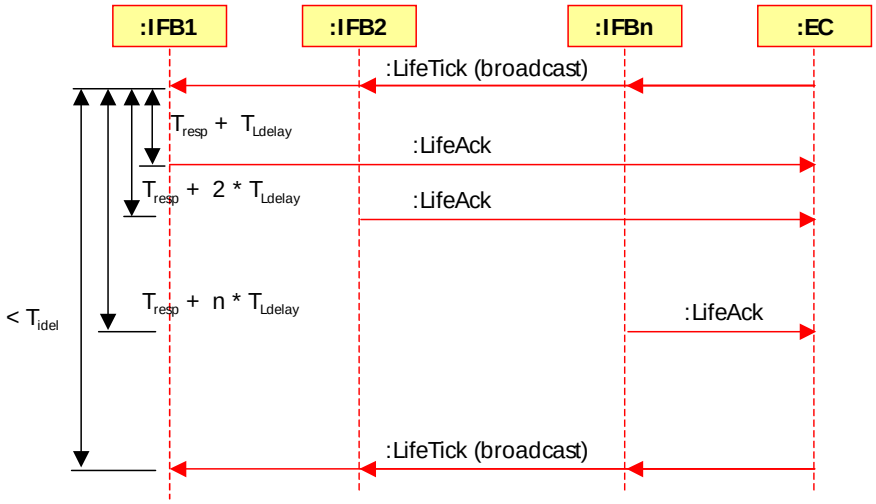
Escenario 1. Conexión de una tarjeta IFB por ambos canales

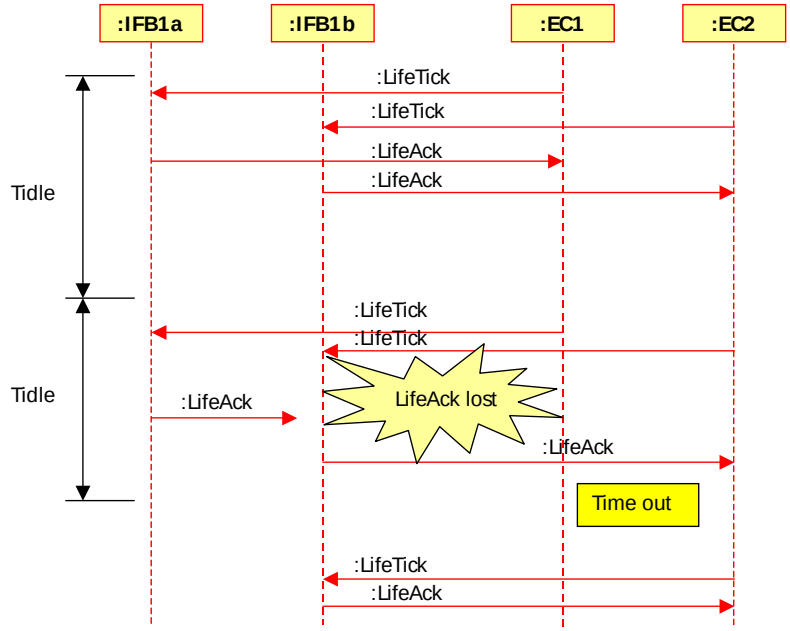


Escenario 2. Conexión fallida de una SEÑAL por ambos canales



Escenario 3. Gestión de los mensajes de vida LIFETICS-LIFEACK





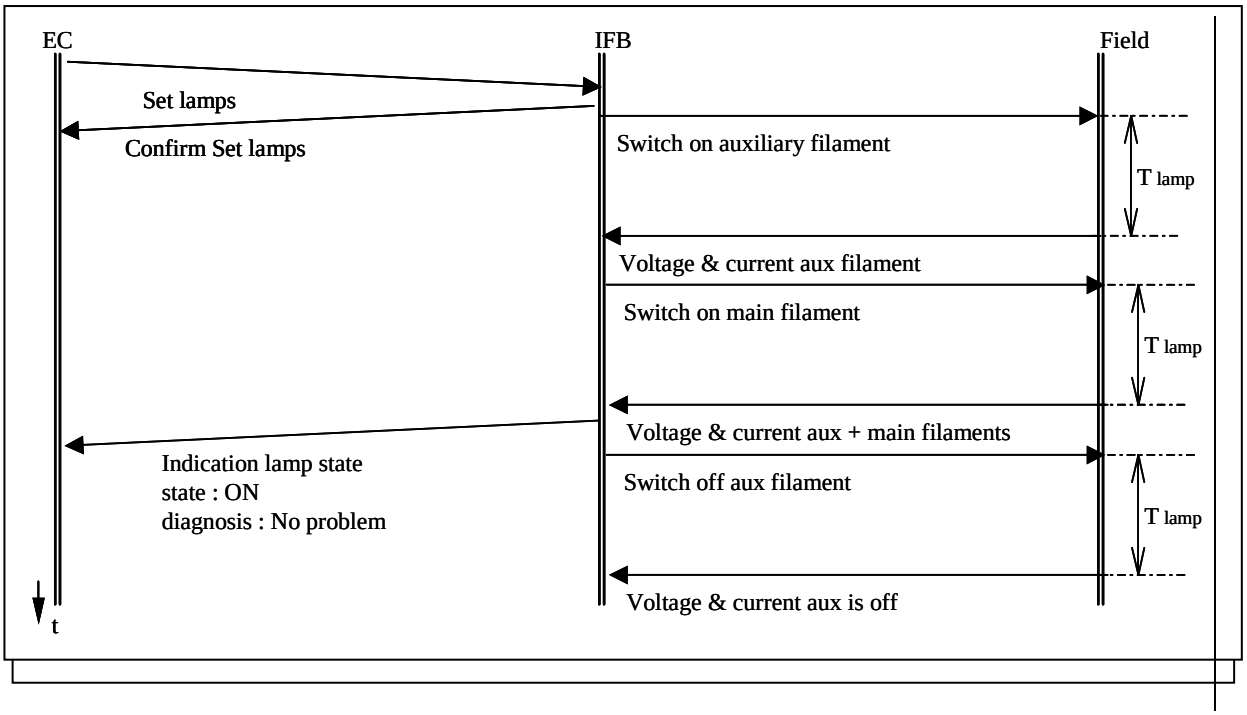
Este escenario pone de manifiesto algunas de las características de tiempo real. El simulador IFBSIMCAM diseñado reproducir los mensajes de las tarjetas IFB, generados en los intervalos de tiempo descritos o bajo petición del usuario con un retraso adicional para verificar que la respuesta del sistema de control (EC) es la descrita en los diagramas de secuencia.

Algunos escenarios como los que se muestran a continuación en referencia al encendido de lámparas no sólo condicionan la gestión ordenada de mensajes entre IFBs y ECs, sino que imponen restricciones en los tiempos de respuesta del campo. Esto supone

desde el punto de vista del simulador una gestión compleja de tiempos para verificar que el programa SIMENC responde o no en el tiempo previsto.

Se recuerda que el usuario debe poder alterar estos tiempos para que se verifique que el EC detecta estas situaciones.

Escenario 4. Mensajes y tiempos de reacción en el encendido y apagado de una señal



Una forma habitual de recibir los requisitos corresponde a las tablas que definen la reacción del elemento en función de su estado anterior y el mensaje recibido.

Escenario 5. Reacción de una lámpara

On field filament state	**Message from IFB to EC**		**IFB reaction**
	LAMP state	**Diagnosis**	
3 times current mismatch	Undefined	Current mismatch	Blocking halt
3 times voltage mismatch	Undefined	Voltage mismatch	Blocking halt
OFF	FUSED	Main filament fused	
ON	ON	No problem	
BLINK	N/A		
Short circuit	ON	Short circuit	Non blocking halt
ON + Out of window	-	-	-
BLINK + Out of window	N/A		
Overvoltage	ON	Over voltage	Non blocking halt
Degraded blinking	N/A		

El proceso para comprender los requisitos ha sido muy largo, pero el diseño orientado a objetos ha facilitado mucho la programación final.

6. Verificación

6.1. La verificación y validación de un programa

Para asegurarse de que el programa a implementar cumple los requisitos y tiene la funcionalidad esperada se debe seguir un proceso de validación y verificación. Este proceso tiene que ser llevado a cabo durante y después del proceso de implementación. Se basa en el análisis y pruebas del sistema, comienza con revisiones de los requerimientos y continúa con revisiones del diseño e inspecciones de código hasta la prueba del producto final.

Para definir la verificación y la validación de manera clara y sencilla se puede citar a Boehm [Boehm, 1979]:

- << Validación: ¿Estamos construyendo el producto correcto?>>
- <<Verificación: ¿Estamos construyendo el producto correctamente?>>

Gracias a estas preguntas podemos diferencia claramente la función de cada uno de estos dos procesos. La verificación se encargará de comprobar que el programa satisface los requerimientos funcionales y no funcionales. Mientras que la validación asegura que el sistema satisface su especificación para demostrar que hace lo que se espera que haga.

El fin de estos dos procesos es dotar al sistema de la seguridad necesaria y demostrar que el sistema es suficientemente bueno para cumplir su propósito. El nivel de confianza del sistema depende de:

1. Función del software: Depende del nivel de uso del software y de su importancia para el cliente. Es decir, un sistema de seguridad crítico tendrá un nivel de confianza mucho más alto que un prototipo software de un sistema.

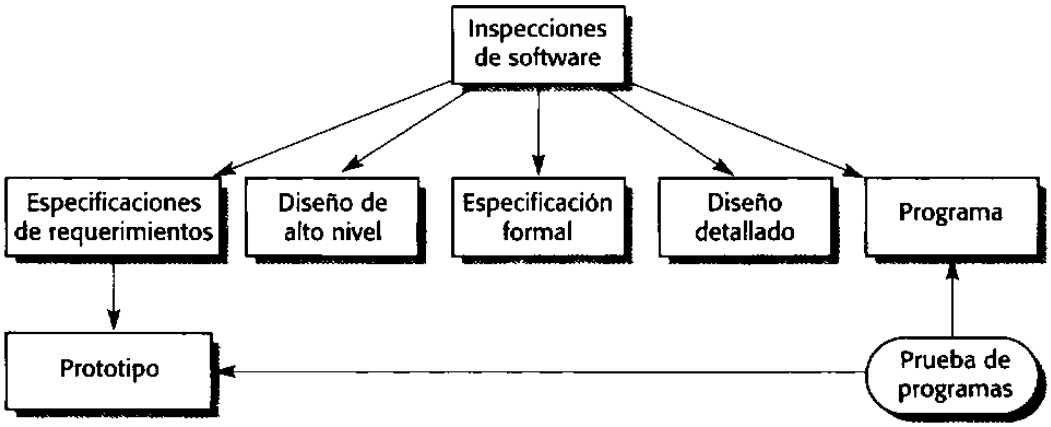
2. Expectativas del usuario: Hasta hace poco tiempo las expectativas del usuario eran más bien bajas. Se aceptaba que el sistema cometiera fallos cuando los beneficios de su uso eran más que sus desventajas. En la actualidad, se da mucha más importancia a estas expectativas y por tanto, a la verificación y la validación del software.

3. Entorno del mercado: Enmarca diferentes factores. La competencia, el precio, la agenda requerida… No es lo mismo fabricar un software que no tiene competencia en el mercado, que uno con gran competencia. Al igual que si un usuario no está dispuesto a pagar mucho dinero por el software tolerará los fallos mejor que por un producto caro.

Dentro de la verificación y la validación se distinguen dos aproximaciones para el análisis y la comprobación del software:

1. Las inspecciones de software. Su función es analizar y comprobar las representaciones del sistema tales como el documento de requerimientos, diagramas de diseño y el código fuente. Además, se puede realizar un análisis automático. Estas dos técnicas se denominan de verificación y validación estáticas, puesto que no hay que ejecutar el software.

2. Las pruebas de software. Se ejecuta una implementación del software con unos datos de prueba y se examinan las salidas de éste, así como su entorno operacional. Las pruebas son una técnica dinámica de verificación y validación.



La figura obtenida del libro de Ingeniería del Software de Sommerville [Sommerville, 2002] muestra que las inspecciones software y las pruebas son actividades complementarias. Las flechas indican las etapas en el proceso en las que pueden utilizarse estas técnicas. Es decir, las inspecciones del software durante todo el proceso.

Sólo una vez que se tiene un prototipo se pueden realizar pruebas del sistema. Una ventaja del desarrollo del software de manera incremental es que se obtiene un prototipo muy pronto y las funcionalidades que se van añadiendo se pueden ir probando en paralelo.

Las técnicas de inspección comprenden las inspecciones de programa, el análisis automático y la verificación formal. Las técnicas estáticas no pueden demostrar que el software es operacionalmente útil, así como comprobar su rendimiento o su eficiencia.

Existen dos tipos diferentes de pruebas que pueden realizarse en diferentes etapas del proceso del software:

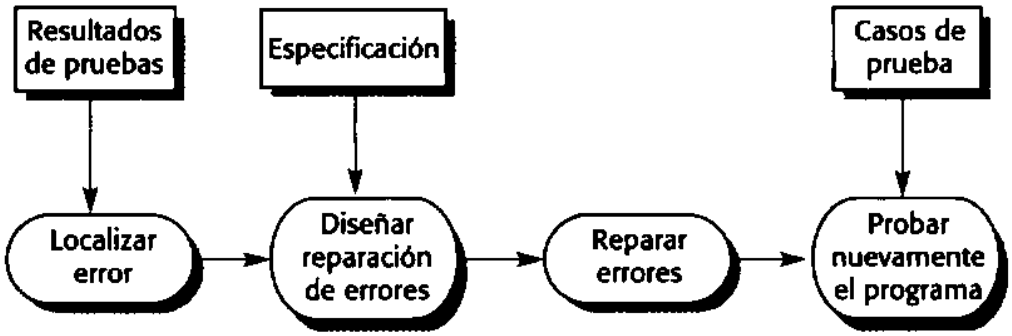
1. Las pruebas de validación: sirven para demostrar que el software cumple los requerimientos. Además, se pueden realzar pruebas estadísticas para probar el rendimiento y la fiabilidad.

2. Las pruebas de defectos intentan mostrar defectos en el sistema en lugar de simular su uso operacional. Halla inconsistencias entre un programa y sus especificaciones.

Normalmente, los procesos de validación y verificación y su depuración se intercalan. Cuando se descubren defectos en el programa que se está probando, tiene que cambiarse para corregir tales defectos. Sin embargo, las pruebas y la depuración tienen diferentes objetivos:

1. Los procesos de verificación y validación buscan la existencia de defectos en el software.

2. La depuración es un proceso mediante el cual se localiza y se corrigen estos defectos. La figura a continuación muestra este proceso [Sommerville, 2002]



No existe un método sencillo para depuración de programas. Normalmente, se buscan patrones clásicos de errores y errores característicos de los lenguajes de programación, como la liberación de memoria en C.

Localizar defectos en un programa es complicado, principalmente porque el defecto no tiene porqué encontrarse donde el programa falló. Por lo que son necesarias pruebas complementarias y trazas que cerquen el área en el que se encuentra el defecto. Las herramientas de depuración que recopilan la información sobre la ejecución del programa también pueden ayudar a localizar la fuente del problema.

Las herramientas de depuración interactivas proporcionan un entorno de ejecución especializado para el programa que permite acceder a la tabla de símbolos del compilador y, desde ahí, a los valores de las variables. Con la ejecución paso a paso se pueden observar dichos valores en cada sentencia y por tanto averiguar la localización del defecto.

Una vez descubierto el defecto hay que corregirlo y volver a validar el sistema. Ya sea inspeccionando el programa de nuevo o haciendo pruebas de regresión en las que se

ejecuten de nuevo los test existentes. Estas pruebas de regresión ayudan a comprobar que no se han introducido nuevos defectos en el programa.

En principio, deberían repetirse todos los tests después de la reparación de cada defecto. En la práctica, el coste es demasiado alto por lo que parte del plan de pruebas consiste en identificar las dependencias entre las diferentes pruebas. Esto es, debería poderse establecer una traza entre los casos de prueba y los componentes que son probados. Si esta trazabilidad se documenta, entonces se puede ejecutar un subconjunto de los casos de prueba del sistema para comprobar el componente modificado y sus dependientes.

6.2. La verificación en Thales

La verificación el Software en THALES, en concreto el área dedicada a señalización y tráfico ferroviario es especialmente fuerte exhaustiva al riesgo para las personas. Se distinguen varias fases:

- Verificación interna del software de enclavamiento
  En esta etapa se pretende obtener la certificación por parte del departamento de verificación de Thales para una versión del software. Se trata de una versión desarrollada sobre C para procesadores propios funcionando en un entorno redundante 2 de 3. La aplicación es un conjunto de módulos para el control de diferentes elementos de campo: agujas, pasos a nivel, señales etc. La verificación se hace para cada módulo y se prueba sobre elementos reales.

- Verificación interna de la configuración de un enclavamiento para una estación concreta
  En esta etapa se seleccionan los módulos necesarios para una estación concreta y se verifica el funcionamiento global del sistema y el correcto funcionamiento de cada elemento. En esta etapa se utiliza el simulador diseñado en este proyecto. Las pruebas de stress y de rotura de elementos son fundamentales. La garantía de haber superado estas pruebas y su registro será utilizada en caso de accidentes frente a juicios por demanda de clientes o usuarios.

- Validación por parte del cliente del enclavamiento específico

  En esta etapa el cliente (Adif, Renfe, etc.) realiza una comprobación del funcionamiento del enclavamiento verificando las maniobras de tráfico de trenes

sobre escenarios válidos y prohibidos para garantizar el correcto funcionamiento. Hasta no superar estas pruebas no se instala en campo.

- Validación en la estación

  En esta etapa el cliente y THALES verifican el correcto funcionamiento sobre el terreno haciendo circular trenes (normalmente máquinas) . Superadas estas pruebas se abre al tráfico normal.

6.3. Tests de validación

A continuación se muestra una tabla de ejemplos de test de verificación del sistema y una clasificación de estos.

TC Id	Descripción de la acción	Resultado Esperado	Req. Cubierto	Resultado del Test	DDTS relacionados
3.2.1	Carga la BDD con el simulador SIMCAM y con IFB's reales	Carga y dibuja los elementos y está conectado con IFB SIMCAM		**No ejecutado**	
3.2.2	Cambia el puerto de comunicación con SIMCAM	Cierra la conexión socket y abre la comunicación con otro puerto.		OK	
3.2.3	Carga la BDD con el simulador SIMCAM y con IFB's reales	La aplicación IFB SIMCAM mandará a SIMENC una lista de los puertos libres y ocupados.		**No ejecutado**	TASdt_21871
3.2.4	Modificación de las indicaciones de elementos TI	Queda reflejado en los objetivos reales IFB SIMCAM		OK	
3.2.5	Seguimiento de las indicaciones TI en el diálogo espía	Se recoge en el diálogo del espía el envío de las indicaciones		OK	
3.2.6	Modificación de las indicaciones de elementos PT	Queda reflejado en los objetivos reales IFB SIMCAM		OK	
3.2.7	Seguimiento de las indicaciones PT en el diálogo espía	Se recoge en el diálogo del espía el envío de las indicaciones		OK	
3.2.8	Modificación de las indicaciones de elementos CI	Queda reflejado en los objetivos reales IFB SIMCAM		OK	
3.2.9	Cambiar estados del SG: normal, fundido, cortocircuito	Queda reflejado en los objetivos reales IFB SIMCAM		OK	

3.2.10	Seguimiento de las indicaciones CI en el diálogo espía	Se recoge en el diálogo del espía el envío de las indicaciones		OK	
3.2.11	Se verifica que los DIO-TEST no son detectados como comandos	Se verifica que el tiempo de filtro de SIMCAM es correcto		OK	
3.2.12	Se verifica la recepción de comandos PT	The PT elements follow and comply the predefined logic		OK	
3.2.13	Se verifica la recepción de comandos PT	Las indicaciones del PT se fuerzan para verificar la respuesta del FEC		OK	
3.2.14	Se verifica la recepción de comandos SG	Los elementos SG siguen y cumplen la lógica predefinida		OK	
3.2.15	Se verifica la recepción de comandos CI	Los elementos CI siguen y cumplen la lógica predefinida		OK	
3.2.16	Se verifica la recepción de comandos CI	Las indicaciones del CI se fuerzan para verificar la respuesta del FEC		OK	
3.2.17	Cargar elementos SI ()FB SG) de un proyecto	Comandos e indicaciones son mandadas desde/a SIMENC-SIMCAM	Funcionalidad general de SI usando tarjetas IFB reales	**No ejecutado**	Tipo de tarjeta 4 con tarjeta IFB SIC real no funciona correctamente TASdt22184

TABLA: TESTS DE VERIFICACIÓN (SIMENC-IFB SIMCAM)

Errores pendientes

De acuerdo con la clasificación de severidad DDTS descrita (3FG-96022-AAAA-ASZZA), DDTS con severidad 5 o severidad 6 no se consideran defectos para esta medición.

Sumario de defectos					
SEVERIDAD	**En Análisis** (A)	**Solucionándose** (O)	**Resueltos** ®	**En Cancelación** ©	**TOTAL**
Sev 1 VITAL					
Sev 2 CRITICA					
Sev 3 ELEVADA					
Sev 4 LEVE			1		**1**
					1

TABLA: SUMARIO DE DEFECTOS

7. Conclusiones

Se ha desarrollado una aplicación informática para la empresa THALES S.A., con el objetivo de validar el software desarrollado por ellos para el control de tráfico ferroviario. Las principales características del programa son:

- Simula el comportamiento de tarjetas electrónicas (IFB de THALES S.A.) que controlan elementos de campo como agujas, pasos a nivel, señales, etc.
- Se comunica vía CANBUS de forma redundante con el sistema de control del enclavamiento electrónico
- El objetivo es verificar que el software del enclavamiento es correcto y responde adecuadamente frente a cualquier comportamiento anómalo del "campo"
- Se comunica vía TCP/IP con la aplicación SIMENC desde la que el usuario genera el comportamiento del "campo"
- Permite la monitorización y registro de los mensajes CANBUS enviados y recibidos
- Permite la simulación de errores en el envío y recepción de mensajes CANBUS
- Desarrollado en VISUAL C++ 6.0 de Microsoft, usa la librería y driver de National Instrument para la comunicación CANBUS

La participación en un proyecto real para la industria ha permitido un aprendizaje directo de herramientas de diseño software, de los procedimientos de trabajo y del sector de control de tráfico ferroviario. Se señalan a continuación algunos de estos elementos.

Herramientas:

- Sistema Operativo Windows/XP
- Entorno de desarrollo VISUAL C++ 6.0 de Microsoft
- Bases de datos relacionales Microsoft ACCESS
- Herramienta CASE Rational Rose
- Microsoft Word para el desarrollo de la documentación
- Aplicaciones propias de THALES S.A. para la verificación

Ingeniería software:

- Diseño evolutivo, debido sobre todo a la constante demanda de nuevas funcionalidades
- Diseño de un programa con restricciones de Tiempo Real

- Uso de protocolos de comunicación basados en sockets TCP/IP y CANBUS
- Proceso de gestión de versiones basado en “CLEAR CASE” de la firma Rational Rose
- Proceso de documentación basado en modelos estrictos definidos por la empresa
- Trabajo en equipo coordinado por el jefe de proyecto y en constante contacto con el personal de THALES S.A.
- Proceso de entrega de versiones con compromiso de fecha y formación a los empleados de THALES S.A.
- Procedimientos de “traza” y registro de mensajes para la depuración de código y la integración del programa con el resto de la aplicación
- La importancia de las pruebas de regresión aplicadas a los programas después de una modificación y las ventajas de una automatización de este proceso (Se ha solicitado una modificación del programa para permitir esta característica)

Control de tráfico ferroviario:

- Sistemas de Control (enclavamientos electrónicos) con redundancia
- Control jerárquico de toma de decisiones con interfaces estándar que permiten integrar sistemas de diferentes fabricantes
- Elementos sensores y actuadores (mecánicos y electrónicos) y sus armarios de conexión y protección electrónica
- Normativa y gestión del tráfico ferroviario

Por todo ello el proyecto ha resultado muy útil en nuestra formación. Aunque nos consta que la realidad es así, algunos detalles han ocasionado un trabajo extra

- El desarrollo en paralelo de nuestro programa y del software de THALES con el que debía funcionar ha ocasionado problemas de calendario, de especificaciones y enormes dificultades para depurar nuestro código
- Cambios constantes de requisitos debido a que a medida que disponían del programa descubrían nuevas posibles pruebas para su software. A modo de ejemplo, en un momento dado, solicitaron poder alterar cualquier byte del mensaje solicitado. Luego modificar sólo algunos tipos de mensajes y finalmente un editor del filtro
- La entrega de una nueva versión a THALES S.A. no siempre pudo ir acompañada de una verificación total de nuestro código ya que los escenarios de prueba requerían no sólo 3 o 4 programas de THALES sino también un hardware escaso y de personal suyo casi nunca disponible
- Los horarios de prueba, a veces, eran justo cuando la mayoría del personal de THALES estaba comiendo y se disponía de hora y media escasa

- La formación desigual y la falta de información del proyecto entre el personal de THALES ha ocasionado a veces solicitud de correcciones de errores que no existían. Principalmente la mala configuración de los datos (en teoría realizada por una aplicación de THALES) de la estación, el desconocimiento de algunas funcionalidades y las restricciones de Tiempo Real eran las causas de sus peticiones, otras veces eran fallos de sus programas

Por último destacar que el programa desarrollado está siendo utilizado en 12 puestos de verificación en la sede de Madrid. Que por tratarse de una multinacional en ocasiones algunos clientes extranjeros compran el simulador para validar sus enclavamientos, tal es el caso de Turquía y Finlandia. THALES S.A. está estudiando la solicitud de la patente de todo el simulador del que nuestro programa forma parte.

8. Bibliografía

Documentos aportados por THALES S.A.:

- 3FG_00100_AFAA_PCZZA: SIMENC: User Manual
- 3FG_00100_AFAA_RJZZA: SIMENC: Installation Manual
- 3FG_00100_AFAA_UNZZA: Integration and Verification Test Report
- 3FG_00100_AFAA_UVZZA: Application Data Specification

Libros de consulta:

- Boehm, B. W., *Software engineering; R & D trends and defense needs. In research. Directions in Software Technology*. Cambridge, MA: MIT Press, 1979.
- Booch, G; Rumbaugh, J; Jacobson, I; *UML. El lenguaje de modelado unificado. Guía del usuario. 2ª edición;* Ed. Addison Wesley Iberoamericana, 2006.;
- Gamma, E; Helm, R; Johnson, R; y Vlissides, J; *Patrones de diseño;* Addison Wesley, 2003;
- Horowitz, E; Sahni, S.; Rajasekaran, S. *ComputerAlgorithms*. Computer Science Press, 1998.
- Pressman, R.; *Ingeniería del Software. Un enfoque práctico;* 6ª edición, Ed. McGraw-Hill, 2005;
- Sommerville, Ian. *Ingeniría de software*. Addison Wesley, 6ª edición, 2002.

Páginas WEB consultadas:

- http://canbus.galeon.com/electronica/canbus.htm
- http://www.wikipedia.com