

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 725 684**

51 Int. Cl.:

B60C 23/04 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **08.08.2017** **E 17185238 (7)**

97 Fecha y número de publicación de la concesión europea: **13.02.2019** **EP 3284619**

54 Título: **Sensor de supervisión de presión de neumáticos**

30 Prioridad:

19.08.2016 EP 16185015

45 Fecha de publicación y mención en BOPI de la traducción de la patente:

26.09.2019

73 Titular/es:

WEGMANN AUTOMOTIVE GMBH (100.0%)
Rudolf-Diesel-Straße 6
97209 Veitshöchheim, DE

72 Inventor/es:

DR. RIEMANN, ANDREAS y
WICHER, BARTLOMIEJ

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 725 684 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Sensor de supervisión de presión de neumáticos

Campo de la invención

5 La invención se refiere a sensores de supervisión de presión de neumáticos para vehículos y específicamente a sensores de supervisión de presión de neumáticos universales que son configurables para una variedad de vehículos.

Descripción de la técnica relacionada

10 Un sistema de supervisión de presión de neumáticos (TPMS) es un sistema electrónico para supervisar la presión del aire dentro de un neumático en diversos tipos de vehículos. Un TPMS proporciona información sobre la presión de los neumáticos casi en tiempo real al sistema de control del vehículo y al conductor. Los TPMS mencionados en el presente documento son TPMS directos, que usan sensores (unidades de rueda) montados dentro de un neumático para medir la presión de los gases dentro del neumático.

15 El sensor se comunica con el módulo de control del vehículo mediante el uso de señales inalámbricas, que típicamente son señales de radiofrecuencia (RF). Estas señales contienen la información de presión del sensor y posiblemente otros datos como la temperatura, el número de identificación del sensor o la información de ubicación de la rueda. Además, se puede usar una unidad de programación externa para comunicarse con los sensores, típicamente con el uso de señales de radiofrecuencia baja (LF) o contactos físicos. La comunicación entrante puede ser usada por un instalador de sensores para activar el sensor con el fin de recibir información de diagnóstico o para modificar el sensor para que opere de acuerdo con una especificación del sistema TPM particular.

20 Los sensores pueden instalarse por fabricantes de vehículos como productos OEM (del inglés, Original Equipment Manufacturers, Fabricantes de Equipo Original) y pueden instalarse en instalaciones de instalación para fines de reemplazo o acondicionamiento como una solución para el mercado posventa (PV). Los sensores pueden operar de manera diferente en diferentes sistemas TPM, dependiendo del fabricante del vehículo, el modelo, el año de producción, la marca y el fabricante del TPMS. Las diferencias entre los sistemas TPM influyen en la implementación del sensor, por ejemplo: transición entre diferentes modos de operación, condiciones de activación para un flujo de programa interno, algoritmos de aprendizaje, temporización, características de la señal inalámbrica, protocolo de comunicación, contenido de paquete de datos, etc.

25 Para que un único sensor cubra la mayoría de los sistemas TPM de mercado de posventa (AM), estos sistemas deben contar con el respaldo de las respectivas implementaciones de sensores AM-TPMS. Para lograr este objetivo, uno puede implementar una multitud de tipos de sensores, cada uno implementando un único sistema TPM, o un sensor universal, que puede usarse directamente en todos los vehículos existentes relevantes, o que puede ser programado/configurado por un instalador para soportar uno o varios sistemas TPM. El uso de una multitud de tipos de sensores de sistema único no es deseable, ya que requiere que los instaladores de sensores almacenen una multitud de sensores. Esto resulta en una alta inversión inicial para el instalador y la cadena de suministro y hace que la selección del sensor sea más lenta. Un sensor universal parece ser una solución mucho más económica.

35 El documento US 7.518.495 B2 desvela un procedimiento, sistemas y herramientas para programar sensores con un programa de software que soporta un único sistema TPM. El software de programa adecuado para el sensor se selecciona de una base de datos. Esto es muy flexible ya que las nuevas implementaciones de programas pueden agregarse más adelante a la base de datos. Como el software completo debe cargarse en el sensor, los tiempos de programación son comparativamente largos, ya que se usa una interfaz de comunicación de baja velocidad. Esta interfaz se usa normalmente para transmitir datos específicos de sensores y se acciona por los instaladores de sensores. Además, la comunicación intensa reduciría la capacidad de la batería incorporada en el sensor. Alternativamente, se puede usar una interfaz cableada. Tal interfaz cableada requiere hardware adicional como controladores y contactos eléctricos, lo que hace que el sensor sea susceptible a daños por ESD y corrosión en los puntos de contacto eléctrico. El manejo de una interfaz cableada es más complejo ya que un cable debe conectarse al sensor antes de la programación y desconectarse después de la programación.

40 El documento US 8692661 B2 desvela un sensor universal. Una pluralidad de programas seleccionables se almacena en el sensor durante la producción y para adaptar el sensor a un vehículo específico, el programa requerido es seleccionado por el instalador del sensor. Esto permite una programación muy rápida, ya que solo se debe seleccionar el programa correcto. El inconveniente de esta solución es que se debe almacenar una gran cantidad de programas en el sensor, lo que requiere una gran sobrecarga de memoria, lo que aumenta los costes de los sensores. Alternativamente, el microcontrolador de memoria limitada del sensor solo puede preconfigurarse para una selección de modelos o protocolos de vehículos, lo que requeriría mantener una gran cantidad de sensores en stock para proporcionar cobertura de AM. Otra desventaja es que, debido a los programas prealmacenados, no es posible la adaptación a requisitos futuros. En su lugar, se deben desarrollar nuevos sensores.

55 El documento US 2015/0202932 A1 desvela un sensor que almacena una versión básica de un programa en su memoria. Para la configuración, los parámetros de programa de tipo de vehículo seleccionado se almacenan en la memoria. Esto permite una programación comparativamente rápida, ya que solo los parámetros deben transmitirse al

sensor. Una adaptación a los requisitos futuros solo es posible dentro del alcance de los parámetros. No se pueden agregar nuevas funciones básicas.

5 El documento EP 2821260 A1 desvela un procedimiento para configurar un sensor mediante la eliminación de procedimientos de codificación innecesarios. Como inicialmente se debe almacenar una gran cantidad de procedimientos de codificación en la memoria, se requiere una memoria comparativamente grande o la limitación de memoria de los microcontroladores comerciales obliga a almacenar una cantidad de sensores que aumenta aún más los costes del sensor. Finalmente, la adaptación a los nuevos sistemas TPM no siempre es posible, a menos que encajen en un sistema TPM existente. De lo contrario, se debe liberar un nuevo sensor.

10 Cada vez que aparece un nuevo sistema TPM en un mercado o se modifica un sistema existente o si se descubren posibilidades de mejora o errores dentro del software de sensor existente, el software debe actualizarse. Parte de la técnica anterior mencionada anteriormente se basa en una unidad de programación externa usada por el instalador de AM para realizar actualizaciones de campo en un sensor por medio de comunicación cableada o inalámbrica. En el documento US 2015/0202932 A1, el intervalo de actualizaciones de campo está limitado por el sistema de software subyacente. Esto puede requerir que el instalador devuelva los sensores al productor para obtener actualizaciones.
15 En algunos casos, tales actualizaciones pueden no ser posibles debido a limitaciones de hardware y software, lo que obliga a retirar los sensores. Los documentos US 8692661 B2 y EP 2821260 A1 no permiten ningún nivel de actualizaciones de software, por lo que siempre es necesario reemplazar el sensor. Esto es desventajoso y oneroso no solo para los instaladores, sino también para toda la cadena de suministro, así como para los propios productores de sensores. También puede requerir la introducción del nuevo número de versión del sensor para cada actualización de software, lo que afecta a todas las partes implicadas en el sensor después del mercado con costes adicionales, mayor complejidad de manejo, demoras, errores humanos y tasas de retorno.

20 La divulgación del documento US 7.518.495 B2 permite actualizaciones de campo de software completas, pero los tiempos de carga pueden ser largos, lo que los hace más propensos a errores de comunicación, lo que obliga al instalador a repetir el procedimiento. Según se informa, este es un problema importante para los instaladores, ya que induce costes adicionales y retrasos. Además, las intensas transferencias de datos pueden consumir una gran cantidad de potencia de la batería, lo que limita la vida útil del sensor.

25 Un procedimiento para configurar un sensor de presión de neumáticos en el documento WO 2016/012627 A1. Otro sistema de integración, procedimientos y dispositivos para sensores de supervisión de presión de neumáticos se desvela en el documento US 2015/015387 A1 y una herramienta del sistema de supervisión de presión de neumáticos con base de datos de número de partes se desvela en el documento US 2009/267751 A1.

30 La solicitud PCT no publicada PCT/EP2017/062682 se incluye completamente en el presente documento como referencia. Desvela un sensor TPMS que tiene un cargador de arranque para cargar datos de configuración específicos del vehículo.

Sumario de la invención

35 El problema a resolver por la invención es proporcionar un sensor de TPMS universal (en el presente documento denominado también sensor) que puede adaptarse (programarse) a una pluralidad de vehículos y una unidad de programación para adaptar el sensor. Preferentemente, la unidad de programación debe ser un dispositivo de mano compacto que permita la programación tanto en el campo como en un taller de neumáticos. El sensor TPMS no debería ser mucho más complejo y, por lo tanto, caro en comparación con los sensores TPMS estándar. Además, la adaptación del sensor debería ahorrar energía, de tal modo que el sensor no requiera una batería más grande o sufra una vida útil reducida. Finalmente, la adaptación debe requerir un tiempo comparativamente corto, de tal manera que se pueda realizar durante un cambio de neumático y/o un procedimiento de balanceo.

Las soluciones del problema se describen en las reivindicaciones independientes. Las reivindicaciones dependientes se refieren a mejoras adicionales de la invención.

45 Las realizaciones proporcionan un sensor TPMS que puede adaptarse a un vehículo específico, por ejemplo, por los instaladores de sensores en talleres de reparación de automóviles con el uso de una unidad de programación.

50 En una realización preferente, un sensor TPMS comprende al menos un componente de detección ambiental como un sensor de presión, sensor de aceleración o temperatura que indica al menos un parámetro indicativo de al menos una condición de o dentro del neumático, una unidad de procesamiento para ejecutar un programa, que puede ser un microcontrolador, una memoria no volátil para almacenar al menos un código de programa y/o datos y un módulo de comunicación de sensor que puede comprender un transmisor inalámbrico, un receptor inalámbrico o cableado o un transceptor inalámbrico o cableado para enviar información a/desde un vehículo y/o una unidad de programación. La información puede comprender detalles acerca del al menos un parámetro indicativo de al menos una condición de o dentro del neumático, por ejemplo, presión, temperatura, dirección de rotación, el estado de la batería del sensor o alguna otra información con respecto al estado del sensor. El sensor tiene preferentemente una carcasa que puede montarse en un neumático o una llanta de tal manera que se mantiene dentro de un neumático montado en la llanta.

55 El sensor TPMS implementa un cargador de arranque que permite cargar datos de adaptación al sensor. El cargador

de arranque es un código de programa almacenado en la memoria del sensor. Una unidad de programación se comunica preferentemente con el cargador de arranque mediante señales de baja frecuencia (LF) y señales de alta frecuencia (RF) y mediante el uso de un protocolo de carga de arranque para la comunicación. Los datos de adaptación pueden comprender el código de programa y/o los datos de una aplicación (aplicación) y/o una subaplicación (subaplicación). En el presente documento, una aplicación puede incluir tales datos de adaptación. El término subaplicación se usa para una aplicación que depende de una aplicación. El contenido de los datos de adaptación cargados en el sensor depende de las versiones de los datos actuales almacenados en el sensor y del vehículo objetivo. La unidad de programación recupera las versiones actuales de los datos de adaptación que residen en el sensor y decide si y qué tipo de datos nuevos de adaptación deben cargarse en el sensor. La decisión se basa en la selección del vehículo objetivo que puede ser realizada por un usuario de la unidad de programación. El usuario puede seleccionar el vehículo objetivo mediante una pantalla táctil o un teclado. Alternativamente, la selección puede ser realizada por un lector electrónico como un lector de código de barras o un lector de transpondedor. A continuación, los datos de adaptación se cargan en el sensor, por lo que se adaptan al vehículo objetivo.

El cargador de arranque es una parte del software dentro del sensor de TPMS y se usa preferentemente para cargar una aplicación y subaplicación de TPMS en el sensor. El fin de cargar las aplicaciones es adaptar el sensor a un sistema TPM específico. El cargador de arranque acepta preferentemente órdenes de baja frecuencia (LF) y les responde con respuestas de alta frecuencia (RF). Algunas de las órdenes y/o respuestas pueden estar encriptadas. Las órdenes y las respuestas se transmiten de acuerdo con un protocolo de carga de arranque.

Las órdenes LF de la unidad de programación pueden comprender al menos uno de los siguientes:

OBTENER DESAFÍO: Esta orden inicia la carga de arranque e inicia una nueva sesión de carga de arranque. Puede ser enviada en cualquier momento.

Se puede usar una cookie mágica para verificar la integridad, porque los datos de la orden no son dinámicos.

Preferentemente, el sensor puede responder a esta orden con:

- **DESAFÍO:** si la integridad de la orden se ha preservado. Esta respuesta comprende preferentemente un paquete de desafío, que se puede usar más adelante, por ejemplo, para la autenticación.
- **NADA:** en orden incompleta y/o incorrecta (fallo de verificación de integridad). Normalmente, esta orden es una respuesta que falta (no hay respuesta), pero también puede ser un código de respuesta específico.

INIT APP y INIT SUBAPP: Estas órdenes contienen datos de autenticación. Los datos de autenticación pueden ser el desafío recibido en la respuesta a la orden OBTENER DESAFÍO. Preferentemente, el sensor descifra los datos de autenticación y si los datos de autenticación son correctos, el sensor puede borrar una aplicación y/o memoria flash de subaplicación (INIT APP) o solo la memoria flash de subaplicación (INIT SUBAPP), preparando así la memoria flash para recibir órdenes de ESCRIBIR consecutivas.

La integridad y la autenticación de los datos se controlan preferentemente verificando la exactitud de los datos de autenticación.

El sensor puede responder a esta orden con:

- **ESTADO OK:** en el éxito de la orden
- **ESTADO NOK:** si se produjo un error durante la ejecución de la orden. Se puede producir un error, por ejemplo, si la memoria flash no se puede borrar.
- **NADA:** en orden incompleta y/o incorrecta (fallo de verificación de integridad) o datos de autenticación incorrectos.

ESCRIBIR: Esta orden comprende un campo de datos que comprende preferentemente una dirección de 2 bytes seguida de un campo binario que comprende preferentemente un código de aplicación. La orden puede comprender además una cookie de 1 byte y un contador de orden de 2 bytes.

El campo Datos está preferentemente encriptado (ver Encriptación de datos), por lo tanto, su longitud es preferentemente múltiple de un tamaño de bloque de cifrado. Esto implica que la longitud del campo binario es preferentemente igual a $4*n + 3$, donde n es cualquier número natural preferentemente inferior o igual a 61.

Si la línea de datos binarios no está alineada con un límite de 4 bytes, debe rellenarse con ceros.

Preferentemente, la dirección es la dirección física absoluta dentro de la memoria flash.

El contador de órdenes de 2 bytes se puede usar para verificar si las órdenes llegan en la secuencia correcta. El contador se establece preferentemente en 0 para la primera orden ESCRIBIR dentro de la sesión y preferentemente se incrementa por cada orden consecutiva.

Esta orden puede fallar con ESTADO NOK debido a un error de escritura. En este caso, todo el procedimiento de

carga de arranque debe interrumpirse y repetirse desde cero (comenzando con la orden Obtener Desafío). Esto se debe a que algunas plataformas no permiten escribir en flash más de una vez después de borrar.

El sensor puede responder a esta orden con:

- **ESTADO OK:** en el éxito de la orden
- 5 • **ESTADO NOK:** si se produjo un error durante la ejecución de la orden. Se puede producir un error, por ejemplo, si la memoria flash no se puede borrar.
- **NADA:** en orden incompleta y/o incorrecta (error de verificación de integridad) o datos encriptados incorrectamente.

SALIDA: Al recibir una orden de salida válida, el cargador de arranque transmite OK y sale a la aplicación.

El sensor puede responder a esta orden con:

- 10 • **ESTADO OK:** en el éxito de la orden
- **ESTADO NOK:** si se produjo un error durante la ejecución de la orden. Se puede producir un error, por ejemplo, si la memoria flash no se puede borrar.
- **NADA:** en orden incompleta y/o incorrecta (fallo de verificación de integridad).

15 Se envía una respuesta de RF desde el sensor TPMS a la unidad de programación como una reacción a las órdenes LF. Preferentemente, cada respuesta consiste en una única PDU de RF (Unidad de Datos de Protocolo).

La respuesta normal (excepto el paquete de desafío) es un paquete de estado que preferentemente contiene solo un campo de datos de 1 byte:

- **ESTADO OK** - el campo de datos es igual al campo Func de la orden LF. Este estado confirma que una orden ha sido ejecutada exitosamente.
- 20 • **ESTADO NOK** - 0. Este estado indica un error crítico o de seguridad y, preferentemente, requiere que se repita toda la secuencia de carga de arranque (comenzando desde OBTENER DESAFÍO).

25 El DESAFÍO (paquete de desafío) es utilizado preferentemente por la unidad de programación para derivar una clave de dispositivo y un vector de inicialización (IV). La clave de dispositivo y el vector de inicialización pueden usarse por la unidad de programación para encriptar el desafío y enviarlo de nuevo en el orden INIT. Preferentemente, la clave del dispositivo es única para cada sensor TPMS.

NADA es una respuesta que falta (no hay respuesta), pero también puede ser un código de respuesta específico.

Las órdenes LF INIT APP e INIT SUBAPP y WRITE contienen preferentemente datos encriptados. Básicamente, cualquier cifrado de bloque puede usarse para la encriptación.

30 Para la encriptación, se puede utilizar cualquier modo de encadenamiento de propagación de errores, como OMAC, ABC, DCM, PCBC, IACBC, ECBC, XECB, OCM, CWC, CTR/CBC-MAC, CCM, etc. Tal modo garantiza un buen secreto del mensaje, ya que es un modo de encadenamiento. Sin embargo, el modo debe estar diseñado para propagar incluso un error muy pequeño (único bit) en todo el mensaje. Esta característica lo hace adecuado para probar la integridad y autenticidad del mensaje.

35 En consecuencia, al seleccionar el modo de propagación de errores, se elimina la necesidad de un código de autenticación de mensaje (MAC) y una suma de comprobación (como CRC16), lo que acorta la longitud del paquete y minimiza el tamaño de código requerido.

40 La ventaja adicional de usar los modos de propagación de errores de encadenamiento para la encriptación es que es imposible detectar patrones de bytes que aparecen comúnmente en los datos encriptados. Una función característica del código de bytes es que algunos patrones de bytes (cadenas) aparecen muy a menudo. Ya sea en forma de instrucciones únicas o una secuencia de instrucciones. Por ejemplo, una instrucción asm para agregar los 2 últimos valores de la pila puede tomar 4 bytes de datos. Tal instrucción puede aparecer decenas o incluso cientos de veces en el código fuente. Si la instrucción ocurre en un único bloque de encriptación de cifrado que no usa el modo encadenado, su valor encriptado será el mismo para todas las ocurrencias. Esta característica se puede usar por el atacante para ataques de observación. En el caso del modo de operación encadenado (como PCBC), el valor encriptado es siempre diferente, ya que depende de los datos anteriores en el texto sin encriptar.

45 El modo PCBC requiere un vector de inicialización (IV), que puede calcularse mediante el uso del número aleatorio del DESAFÍO. Esto asegura que el texto cifrado es (casi) siempre diferente para el mismo texto plano. Por lo tanto, el valor del texto cifrado es específico de la sesión, lo que además complica los posibles ataques de protocolo.

El papel de la unidad de programación es adaptar el sensor TPMS a un vehículo objetivo seleccionado por un usuario.

En una primera etapa, la unidad de programación se comunica automáticamente con el sensor para obtener sus versiones de software actuales. Tal comunicación puede ser activada por un usuario o simplemente acercándose a la unidad de programación cerca del sensor. Basándose en esa información y en una selección de vehículo objetivo, la unidad de programación identifica si y qué tipo de actualización se requiere para adaptar el sensor al vehículo objetivo. A continuación, ejecuta la adaptación.

La unidad de programación debe soportar y/o proporcionar varias características. Se debe proveer un receptor de RF compatible con el sensor y su cargador de arranque, así como un transmisor LF que cumpla con el sensor y su cargador de arranque para la comunicación con el sensor y su software de cargador de arranque. Para la comunicación del usuario, se deben proporcionar dispositivos de interfaz humana (entrada y salida). Estos pueden ser al menos uno de un panel táctil, un teclado, una pantalla, un altavoz, un timbre, un escáner de código de barras o una cámara. Además, se puede proporcionar una interfaz de datos externa para actualizaciones regulares de la base de datos, como una tarjeta de memoria, una conexión a Internet, etc.

Los dispositivos de entrada/salida de la unidad de programación pueden permitir que el instalador seleccione uno o más sistemas TPM objetivo, según el cual el sensor debería operar, por ejemplo, ingresando manualmente el número de identificación del vehículo (VIN), leyendo el VIN usando un escáner de código de barras o una cámara o seleccionando manualmente el productor del vehículo, marca, modelo, fecha de producción, etc. Los dispositivos de entrada/salida de la unidad de programación también pueden permitir la inserción de números de versión de las aplicaciones o la inserción de otra información que indique tales números de versión, como el número de serie del sensor, que puede ser usado por la unidad de programación para encontrar los números de versión requeridos en su memoria. La unidad de programación también puede adquirir los números de versión directamente desde el sensor, usando el módulo de comunicación del sensor. La unidad de programación puede comprender un algoritmo que usa la selección del sistema TPM objetivo y los números de versión para detectar automáticamente si para configurar el sensor para que opere de acuerdo con el sistema TPM objetivo, el sensor requiere alguna actualización de software de campo completo, carga de datos o ninguna acción adicional.

Preferentemente, la unidad de programación comprende al menos uno y, más preferentemente, todos los siguientes componentes de software:

Base de datos de grupo de vehículos (VGDB): El VGDB divide los vehículos en grupos. Cada grupo implementa un sistema TPM. La base de datos de grupos de vehículos contiene varias entradas que representan grupos de vehículos. Cada entrada contiene una serie de modelos de vehículos específicos determinados por el fabricante, modelo, marca, fecha y posiblemente otros factores. Las entradas son preferentemente números, pero también pueden ser identificadores alfanuméricos. El selector de archivos binarios usa la base de datos del grupo de vehículos para determinar qué tipo de sistema TPM (grupo de vehículos) es compatible con el vehículo objetivo.

Base de datos de par de versiones (VPDB): El VPDB almacena una lista de versiones de aplicación y subaplicación compatibles con los grupos de vehículos. Cada entrada en la base de datos de par de versiones representa un grupo de vehículos. La entrada contiene varios pares de versiones (VP), cada VP comprendiendo al menos exactamente una versión de la aplicación (AV) y exactamente una versión de la subaplicación (SAV). Las entradas son preferentemente números, pero también pueden ser identificadores alfanuméricos. El selector de archivos binarios usa los pares de versiones para decidir si un sensor requiere una actualización completa del software, solo una adaptación o ninguna acción.

Base de datos de aplicaciones (ADB) y base de datos de subaplicaciones (SADB): La base de datos de aplicaciones (ADB) y la base de datos de subaplicaciones (SADB) almacenan archivos de registro S (SREC) que contienen archivos binarios de aplicaciones disponibles y subaplicaciones. Cada SREC consiste en un número de registros colocados en líneas separadas. Un registro puede tener la siguiente estructura:

| Campo | Contenido |
|-------|---|
| Tipo | cadena "S1" |
| Len | 2 dígitos hexadecimales (1 byte): indica el número de bytes que siguen en el resto del registro (Addr + Bin + Chksm) |
| Addr | 4 dígitos hexadecimales (2 bytes): dirección objetivo de los datos en la memoria del sensor (valor del campo Addr de la orden Escribir) |
| Bin | Datos binarios para almacenar en Addr del sensor (valor del campo Bin de la orden Escribir) |
| Chksm | 2 dígitos hexadecimales (1 byte): Registro de suma de comprobación (puede ser ignorado) |

Los registros de los archivos binarios pueden transmitirse al sensor con el uso de un protocolo de cargador de arranque (sec_btldr_prot). Cada registro puede ser transmitido dentro de una única orden Escribir.

Puede haber una pluralidad de bases de datos como se describió anteriormente. Alternativamente, puede haber cualquier cantidad de bases de datos que almacenen tablas con la información de la base de datos. Preferentemente, solo hay una base de datos que almacena cuatro tablas que incluyen grupos de vehículos, pares de versiones,

aplicaciones y subaplicaciones. La estructura de datos es importante. No es relevante si los datos se almacenan en una o múltiples bases de datos o tablas.

5 Actualizador de base de datos: El VGDB, VPDB, ADB y SADB deben actualizarse regularmente para adaptarse a las últimas actualizaciones de software, nuevos modelos de vehículos y sistemas TPM. El contenido de las bases de datos será proporcionado por un proveedor a los productores de la unidad de programación cuando sea necesario. Un papel del actualizador de base de datos es descargar la última versión de las bases de datos (probablemente desde una ubicación de Internet) y actualizar las bases de datos localmente.

Selector de archivos binarios: Según el vehículo objetivo y la aplicación actual y la subaplicación que reside en el sensor, una de las siguientes acciones puede ser necesaria para adaptar el sensor al vehículo objetivo:

- 10
- Transmitir solo una subaplicación (típica)
 - Transmitir una aplicación y una subaplicación predeterminada al sensor (raro)
 - Nada: el vehículo no es compatible o el sensor ya es compatible con el vehículo objetivo (raro)

La selección del (de los) archivo(s) binario(s) para transmitir al sensor se realiza mediante el selector de archivos binarios. Opera preferentemente de acuerdo con el algoritmo desvelado en el presente documento.

15 Cargador de aplicaciones: El cargador de aplicaciones implementa el lado del transmisor del protocolo del cargador de arranque (sec_btldr_prot). Su papel es transmitir los archivos binarios (SREC) al sensor, tal como lo solicita el Selector de archivos binarios.

En caso de que una aplicación binaria y una subaplicación predeterminada se transmitan al sensor, la transmisión se realiza preferentemente dentro de una única sesión de carga de arranque.

20 Otra realización se refiere a un procedimiento de programación de un sensor TPMS. Después de iniciar el procedimiento, la unidad de programación obtiene al menos una de tipo de vehículo/marca/modelo de un dispositivo de entrada de usuario, como un teclado, pantalla táctil, etc. En una siguiente etapa de extracción, el grupo de vehículos del vehículo objetivo se extrae de la base de datos del grupo de vehículos (VGDB). Si no se encuentra ningún grupo de vehículos, se muestra un error que indica que el tipo/marca/modelo seleccionado no es compatible. Entonces el
 25 procedimiento puede ser terminado o reiniciado. Si se encuentra un grupo de vehículos, al menos un par de versiones se extrae del grupo de vehículos de la base de datos de pares de versiones. Luego se transmite una orden Obtener Desafío al sensor TPMS y se recibe una respuesta. En caso de una respuesta positiva, el algoritmo continúa, de lo contrario se cancela y puede aparecer un mensaje de error. En una etapa siguiente, la versión actual de la aplicación del sensor (AV) y la versión de la subaplicación (SAV) se extraen de la respuesta. En la siguiente etapa se comprueba
 30 si una de estas versiones de aplicación (AV) existe en uno de los pares de versiones (VP) extraídos de la base de datos de versión (VPDB).

Si esto no es cierto, se selecciona el par de versiones (VP) con la mejor versión de la aplicación (AV). Luego, se obtiene el binario de la aplicación de la base de datos de la aplicación (ADB) y la base de datos de la subaplicación (SADB) en función de la versión de la aplicación (AV) y la versión de la subaplicación (SAV) del par de versiones
 35 seleccionadas (VP). El binario obtenido de la aplicación y la subaplicación se carga en el sensor. Después de la descarga, preferentemente se muestra un mensaje de éxito. Finalmente, se termina el algoritmo.

Si en la etapa anterior de verificar si una de las versiones de la aplicación (AV) existe en uno de los pares de versiones (VP), se determina que la versión actual de la aplicación del sensor (AV) existe en uno de los pares de versiones (VP) extraídos de la base de datos, luego se verifica, si la versión de la subaplicación actual (SAV) del sensor es igual a
 40 una versión de subaplicación (SAV) de uno de los pares de versiones (VP) extraídos de la base de datos de pares de versiones (VPDB). Si esto es cierto, entonces no es necesaria ninguna modificación de la memoria del sensor y se procede con una etapa que muestra un mensaje de éxito. Si no es cierto, se obtiene el binario de la subaplicación de la base de datos de subaplicación (SADB) basada en la versión de subaplicación (SAV) de la de los pares de versiones extraídos (VP). Esta subaplicación se carga en el sensor. Después de una descarga exitosa, se puede mostrar un
 45 mensaje de éxito.

Descripción de los dibujos

A continuación, la invención se describirá a modo de ejemplo, sin limitación del concepto inventivo general, en ejemplos de realización con referencia a los dibujos.

- La figura 1 muestra una estructura básica y el uso de un sistema de configuración de sensores.
- 50 La figura 2 muestra detalles de un sistema de configuración de sensores.
- La figura 3 muestra la estructura lógica de un sensor.
- La figura 4 muestra la estructura lógica de una unidad de programación.

La figura 5 muestra los detalles de los datos transmitidos entre el sensor y la unidad de programación.

La figura 6 muestra el sistema de configuración del sensor desde la perspectiva cronológica y de despliegue.

La figura 7 muestra los componentes de software en la unidad de programación.

La figura 8 muestra un algoritmo de adaptación del sensor.

5 En la figura 1 se muestra una estructura básica y el uso de un sistema 840 de configuración de sensores. Una unidad
100 de programación puede tener un dispositivo 110 de salida, preferentemente una pantalla o una impresora, y un
dispositivo de entrada 170, preferentemente una pantalla táctil, un teclado o un escáner de código de barras. El
dispositivo de salida puede proporcionar a un instalador 830 (persona que instala el sensor) una pluralidad de
opciones, por ejemplo, para seleccionar un vehículo objetivo, dentro del cual el sensor debe operar. El instalador puede
comunicar su selección de opciones con el dispositivo de entrada. La unidad de programación configurará el sensor,
por ejemplo, a través de una interfaz 120 de comunicación inalámbrica, para que el sensor opere de acuerdo con un
sistema de TPM específico usado por el vehículo seleccionado por el instalador.

En la figura 2 se muestran más detalles de un sistema de configuración de sensores. La unidad 100 de programación
tiene preferentemente un microcontrolador o microprocesador 130 que está conectado además a una memoria 140 o
almacenamiento. La memoria puede estar organizada como una base de datos y puede contener información sobre
vehículos y sensores. Puede ser cualquier tipo de RAM/ROM, una unidad de disco o un almacenamiento conectado a
la red. Además, se prefieren, si la información sobre los vehículos comprende al menos uno de los fabricantes de
vehículos, modelos de vehículos, años de fabricación, número de identificación del vehículo y requisitos específicos
de TPMS. La información acerca de los sensores puede comprender al menos una de una primera aplicación, una
primera subaplicación, una segunda aplicación, una segunda subaplicación y datos auxiliares. Además, puede haber
enlaces cruzados entre fabricantes de vehículos específicos, modelos, años o vehículos individuales y aplicaciones
específicas. También puede haber información sobre el hardware y software del sensor, como el fabricante, el número
de modelo, el año de fabricación, los números de versión del software y las capacidades de hardware específicas
como la capacidad de procesamiento y el espacio de memoria. La unidad de programación 100 puede contener un
cargador 180 de aplicación para vincular la información acerca de los vehículos y sensores y otra información
almacenada en su memoria para seleccionar el contenido apropiado de los datos 500 de comunicación que se
transferirán al sensor 200, por ejemplo, a través de la comunicación 120 inalámbrica. El contenido de los datos de
comunicación puede ser recibido y usado por el sensor para configurarse para la operación con un sistema TPM
específico.

Para la comunicación con y la carga de software a un sensor, se puede proporcionar un módulo 160 de comunicación
de la unidad de programación. Este módulo de comunicación es preferentemente un módulo de comunicación
inalámbrico, pero también puede ser un módulo de comunicación cableado. Aunque un módulo de comunicación
direccional único es suficiente para cargar datos en el sensor, se prefiere tener un módulo de comunicación
bidireccional para transmitir datos a un sensor y recibir información de un sensor. Los datos recibidos pueden ser
información acerca del sensor como fabricante, tipo, año de fabricación, versión de software o número de serie.
También se puede recibir una suma de comprobación o confirmación de los datos cargados. Además, se prefiere, si
la unidad de programación tiene un dispositivo de salida 110 y un dispositivo de entrada 170, que pueda ser controlado
por el microcontrolador o microprocesador 130. Además, se prefiere que el microcontrolador/microprocesador tenga
acceso a una interfaz 150 de red que puede proporcionar una conexión a un ordenador externo, a una red externa, a
un almacenamiento de datos externo o a Internet. Esto se puede usar para actualizar la base 420 de datos, descargar
y/o actualizar información sobre vehículos, sensores, aplicaciones o cualquier software requerido para manejar y/o
compilar tal información.

Un sensor 200 comprende preferentemente una unidad 230 de procesamiento para controlar el sensor junto con una
memoria 240 no volátil. La memoria puede ser preferentemente una memoria flash, EEPROM, ROM o una memoria
RAM alimentada por batería. Puede proporcionarse un sensor 280 de presión ambiental de supervisión de la presión
del gas dentro del neumático y proporcionar tal información de presión a la unidad de procesamiento. También puede
haber al menos un sensor 290 auxiliar para medir las condiciones ambientales del sensor, como la aceleración, el
choque o la temperatura. Para la comunicación con la unidad 100 de programación, puede proporcionarse al menos
un módulo 260 de comunicación de sensor que comprende al menos un transmisor 261 inalámbrico y un receptor 262
inalámbrico o cableado. Este módulo de comunicación de sensor comprende un cargador 530 de arranque para recibir
los datos 500 de comunicación de la unidad 100 de programación y almacenarlos en la memoria 240. Preferentemente,
los datos de comunicación pueden comprender al menos una de una primera aplicación 510 o una segunda aplicación
520. Para alimentar el sensor 200, se puede proporcionar una batería 270.

Es preferente que el mismo módulo de comunicación por sensor permita la comunicación con un vehículo. Aunque los
requisitos de comunicación para cargar el software desde la unidad 100 de programación y durante la operación normal
y la supervisión de la presión de los neumáticos son significativamente diferentes, pueden ser manejados por el mismo
módulo de comunicación del sensor. El requisito mínimo para la carga de datos es que el receptor 262 en el módulo
de comunicación del sensor 260 se comunique con el cargador 530 de arranque. El requisito mínimo para la
supervisión de la presión de los neumáticos es que el transmisor 261 transmita información de estado a un vehículo.

Puede haber un transceptor de baja potencia para estas dos vías de comunicación. También puede haber un transmisor de alta potencia de alta velocidad. Por supuesto, se puede proporcionar cualquier combinación adicional, como un transceptor de alta velocidad completo para la carga de software y un transceptor de baja velocidad y baja potencia para la supervisión de la presión.

5 La figura 3 muestra la estructura lógica de un sensor. Puede haber un administrador 310 del sistema para tareas generales de administración del sistema, como administración de potencia, programación de tareas o administración de hardware para los recursos del sensor como componentes de detección ambiental, memoria no volátil, módulo de comunicación, unidad de procesamiento, etc. El administrador del sistema también puede comprender funciones de utilidad, como temporizadores, operadores matemáticos, manipulación de memoria, manipulación de cadenas, etc. El administrador del sistema puede proporcionar acceso a la funcionalidad que implementa en al menos una aplicación o subaplicación. El sensor también puede comprender al menos una de una primera aplicación 510, una primera subaplicación 511, una segunda aplicación 520 y una segunda subaplicación 521 almacenadas dentro de la memoria no volátil 240.

15 La figura 4 muestra la estructura lógica de la unidad de programación. Tiene un cargador 180 de aplicaciones para cargar datos en un sensor. Puede haber un administrador 410 de carga para recopilar los datos descargables. Puede ser parte del cargador 180 de aplicaciones. Además, se puede proporcionar una base 420 de datos para almacenar información sobre vehículos y/o sensores y/o para almacenar aplicaciones relacionadas. Puede proporcionarse un administrador 430 de interfaz de usuario para comunicarse con un instalador 830. Además, puede haber un administrador 440 de red y un administrador 450 de receptor TPMS.

20 La figura 5 muestra los datos transferidos entre la unidad 100 de programación y el sensor 200. Los datos 500 pueden comprender al menos una de una primera aplicación 510, una primera subaplicación 511, una segunda aplicación 520, una segunda subaplicación 521 y datos 550 auxiliares. Los datos 550 auxiliares pueden contener un número de identificación de sensor, actualizaciones de firmware, confirmación de recepción de datos, establecimientos de comunicación de protocolo de enlace, números de versión del software que residen en el sensor, sumas de comprobación o código usado para invocar la funcionalidad propietaria del productor del sensor. Lo que se transmite exactamente dentro de los datos 500 de comunicación lo decide el cargador 180 de aplicaciones y depende del contenido de la memoria 240 del sensor, el punto actual en la vida útil del sensor y posiblemente otros factores.

La figura 6 muestra el sistema de configuración del sensor desde una perspectiva cronológica y de despliegue.

30 La figura 6a muestra un primer escenario, donde un sensor 200 no requiere una actualización de software de campo completo que comprende actualizar al menos uno de una primera aplicación 510, una primera subaplicación 511, una segunda aplicación 520, una segunda subaplicación 521 y datos auxiliares 550 que residen en la memoria 240 no volátil. Dentro de la planta 800 de producción, al menos un cargador 530 de arranque y las aplicaciones opcionales se cargan (710) en la memoria 240 no volátil del sensor con el uso de una unidad 600 de programación de producción. La unidad de programación de producción es preferentemente capaz de manejar múltiples sensores a la vez y puede usar un módulo 260 de comunicación de sensores o su módulo 220 de acceso directo para escribir en la memoria no volátil. En otra realización, solo se puede cargar una primera aplicación a la unidad de procesamiento usando su módulo 220 de acceso directo y la segunda aplicación se carga con el uso del módulo 260 de comunicación del sensor. Cualquier otra combinación también es posible. A continuación, el sensor se envía a la planta 820 de instalación a través de su cadena 810 de suministro. En el destino, la unidad 100 de programación se usa para especificar a qué sistema TPM objetivo se debe configurar el sensor. La unidad de programación se comunica con el sensor TPMS y usa la información recibida del sensor, como los números de versión del software almacenado en el sensor, así como su base 420 de datos interna para determinar que el sensor no requiere una actualización completa del software de campo. La unidad de programación también determina que el sensor requiere la instalación de nuevas aplicaciones. En consecuencia, solo el código de la aplicación se transfiere al sensor (720), lo que lo hace compatible con el sistema TPM requerido. Todo el procedimiento de comunicación es relativamente rápido, dura de manera registrada alrededor de 3s.

45 La figura 6b muestra un segundo escenario, donde el sensor requiere la actualización completa del software de campo. De manera similar al procedimiento que se muestra en la figura 6a, el sensor se inicializa en la planta 800 de producción y luego se envía a la planta 820 de instalación a través de la cadena 810 de suministro. De manera similar, el instalador selecciona el sistema TPM objetivo en la unidad 100 de programación y la herramienta determina si es necesaria la actualización completa del software de campo. De acuerdo con la base de datos de herramientas, la versión de la aplicación que reside en la memoria del sensor no es compatible con el sistema TPM objetivo. Una versión más nueva de la aplicación se carga en el sensor dentro de la actualización 710 de software de campo completo, la versión más nueva compatible con el sistema TPM objetivo. Una vez que se completa la carga, la unidad de programación carga automáticamente en el sensor el código (720) de aplicación requerido para que la aplicación recién cargada configure el sensor para que opera de acuerdo con el sistema TPM objetivo. Todo el procedimiento de comunicación es significativamente más largo que el proceso presentado en la figura 6a, dura de manera registrada alrededor de 30s. El procedimiento presentado en la figura 6a es mucho más común que el procedimiento presentado en la figura 6b, ya que la actualización de la aplicación solo es necesaria cuando se necesitan actualizaciones completas de software de campo.

En la figura 7 se muestran los componentes de software en la unidad de programación. La unidad 100 de programación obtiene las últimas bases de datos y/o las últimas 851 actualizaciones de la base 850 de datos de una base de datos en línea y genera a partir de ellas los datos que se transmiten por comunicación 120 inalámbrica a un sensor 200 TPMS. Las actualizaciones de la base de datos son administradas por un actualizador 910 de la base de datos que obtiene las últimas bases de datos de la base de datos en línea y distribuye los datos obtenidos en diferentes unidades de programación de bases de datos. Aquí, la expresión "base de datos" se usa para la simplicidad y claridad. Por supuesto, en lugar de usar múltiples bases de datos, se puede usar una única base de datos. En lugar de usar múltiples bases de datos, alternativamente se pueden usar múltiples tablas en la misma base de datos. El actualizador 910 de la base de datos actualiza 911 la base 920 de datos del grupo de vehículos (VGDB). La base de datos de grupos de vehículos contiene preferentemente una pluralidad de grupos 921, 922 de vehículos. Cada grupo de vehículos comprende preferentemente una pluralidad de entradas 923, 924 de vehículos. Preferentemente, los vehículos dentro de un grupo de vehículos comunes usan el mismo sistema TPM. Los vehículos individuales pueden diferir en parámetros específicos.

El actualizador 910 de la base de datos actualiza aún más 912 la base 930 de datos de par de versiones (VPDB). La base de datos de par de versiones contiene preferentemente una pluralidad de grupos 931, 932. Cada grupo comprende preferentemente un par 933, 934 de versiones. Cada par de versiones comprende exactamente una versión 935 de aplicación (AV) y exactamente una versión 936 de subaplicación (SAV).

El actualizador 910 de base de datos actualiza 913 más una base 940 de datos de aplicación (ADB).

Un selector de archivos final obtiene el grupo 961 de vehículos de la base 920 de datos de grupos de vehículos, obtiene los pares 962 de vehículos de una base 930 de datos de par de versiones, obtiene un binario 963 de la base de datos de la aplicación 940 y obtiene un binario 964 de la base 950 de datos de subaplicación. Al menos los binarios se envían 965 al cargador 180 de aplicaciones, que además transmite estos binarios al sensor 200 TPMS.

En la figura 8, se muestra un algoritmo de adaptación del sensor. Comienza con el inicio 970. Después del inicio 970, la unidad de programación obtiene el tipo de vehículo/marca/modelo de un dispositivo de entrada de usuario, como un teclado, pantalla táctil, etc. En una etapa 972 de extracción, el grupo de vehículos del vehículo objetivo se extrae de la base 920 de datos del grupo de vehículos (VGDB). Si no se encuentra 973 ningún grupo, se muestra un error 974 que indica que el tipo/marca/modelo seleccionado no es compatible. Después de esta visualización, se puede indicar un error 975 adicional y/o se termina el algoritmo. Si se encuentra un grupo, al menos un par de versiones se extrae 976 del grupo de vehículos de la base de datos de pares de versiones. Luego se transmite la orden Obtener Desafío al sensor TPMS y se recibe una respuesta. En caso de una respuesta positiva, el algoritmo continúa, de lo contrario se cancela. En una etapa siguiente, la versión actual de la aplicación del sensor (AV) y la versión de la subaplicación (SAV) se extraen 978 de la respuesta. En la siguiente etapa se comprueba 979 si una de estas versiones de aplicación (AV) existe en uno de los pares de versiones (VP) extraídos de la base de datos de versión (VPDB).

Si esto no es cierto, se selecciona 980 el par de versiones (VP) con la mejor versión de la aplicación (AV). Luego, se obtiene 981 el binario de la aplicación de la base de datos de la aplicación (ADB) y la base de datos de la subaplicación (SADB) en función de la versión de la aplicación (AV) y la versión de la subaplicación (SAV) del par de versiones seleccionadas (VP). El binario obtenido de la aplicación y la subaplicación se cargan 982 en el sensor. Después de la descarga, se muestra 986 un mensaje de éxito. Finalmente, se termina 987 el algoritmo.

Si en la etapa 979 se determina que la versión actual de la aplicación del sensor (AV) existe en uno de los pares de versiones (VP) extraídos de la base de datos, luego se verifica 983, si la versión de la subaplicación actual (SAV) del sensor es igual a una versión de subaplicación (SAV) de uno de los pares de versiones (VP) extraídos de la base de datos de par de versiones (VPDB). Si esto es cierto, entonces no es necesaria ninguna modificación de la memoria del sensor y se procede con la etapa 986 que muestra un mensaje de éxito. Si no es cierto, se obtiene 984 el binario de la subaplicación de la base de datos de subaplicación (SADB) basada en la versión de subaplicación (SAV) de la de los pares de versiones extraídos (VP). Esta subaplicación se carga 985 en el sensor. Después de una descarga exitosa, se muestra un mensaje de éxito 986.

Lista de números de referencia

| | |
|-----|---|
| 100 | unidad de programación |
| 110 | dispositivo de salida |
| 111 | pantalla de opciones de programa |
| 120 | comunicación inalámbrica |
| 130 | microcontrolador/microprocesador |
| 140 | memoria |
| 150 | interfaz de red |
| 160 | módulo de comunicación de la unidad de programación |
| 170 | dispositivo de entrada |
| 180 | cargador de aplicaciones |
| 200 | sensor |

| | |
|-----|---|
| 220 | módulo de acceso directo |
| 230 | unidad de procesamiento |
| 240 | memoria no volátil |
| 260 | módulo de comunicación de sensor |
| 261 | transmisor inalámbrico |
| 262 | receptor inalámbrico o cableado |
| 270 | batería |
| 280 | sensor de presión ambiental |
| 290 | sensor auxiliar |
| 310 | administrador de sistemas |
| 410 | administrador de carga |
| 420 | base de datos |
| 430 | administrador de interfaz de usuario |
| 440 | administrador de red |
| 450 | administrador de receptor |
| 500 | datos de comunicación |
| 510 | primera aplicación |
| 511 | primera subaplicación |
| 520 | segunda aplicación |
| 521 | segunda subaplicación |
| 530 | cargador de arranque |
| 550 | datos auxiliares |
| 600 | unidad de programación de producción |
| 710 | actualización de software de campo completo |
| 720 | carga del código de configuración |
| 800 | planta de producción |
| 810 | cadena de suministro |
| 820 | planta de instalación |
| 830 | instalador |
| 840 | Sistema de configuración de sensor |
| 850 | base de datos en línea |
| 851 | obtener las últimas bases de datos |
| 910 | actualizador de base de datos |
| 911 | actualización de VGDB |
| 912 | actualización de VPDB |
| 913 | actualización de ADB |
| 914 | actualizar |
| 920 | base de datos de grupo de vehículos (VGDB) |
| 921 | primer grupo de vehículos |
| 922 | n grupos de vehículos |
| 923 | vehículo 1 |
| 924 | n vehículos |
| 930 | base de datos de par de versiones (VPDB) |
| 931 | primer grupo |
| 932 | n grupo |
| 933 | primer par de versiones |
| 934 | par de versiones n |
| 935 | versión de aplicación |
| 936 | versión de subaplicación |
| 940 | base de datos de aplicaciones (ADB) |
| 941 | versión de aplicación 1 |
| 942 | versión de n aplicación |
| 943 | binario |
| 950 | base de datos de subaplicación (SADB) |
| 951 | versión de subaplicación 1 |
| 952 | versión de n subaplicación |
| 953 | binario |
| 960 | selector de archivos binarios |

ES 2 725 684 T3

- 961 obtener grupo de vehículos
- 962 obtener pares de vehículos
- 963 obtener binario
- 964 obtener binario
- 965 solicitar transmisión binaria
- 970 inicio
- 971 obtener el tipo/marca/modelo de objetivo del vehículo a partir de la entrada de usuario
- 972 extraer el grupo de vehículos del vehículo objetivo de VGDB
- 973 ¿grupo encontrado?
- 974 error de visualización: no compatible
- 975 error
- 976 extraer VP del grupo de VPDB
- 977 transmisión de Obtener Desafío y obtener la respuesta
- 978 extraer el sensor actual AV y SAV de la respuesta
- 979 ¿existe AV actual de sensor en uno de los VP extraídos de DB?
- 980 seleccionar el VP con la mejor AV
- 981 obtener el binario de la aplicación de ADB y de la subaplicación de SADB basándose en AV y SAV del VP seleccionado
- 982 cargar el binario de la aplicación y la subaplicación (como subaplicación predeterminada) al sensor con el uso del protocolo de carga de arranque.
- 983 ¿SAV actual de sensor igual a SAV de uno de los VP extraídos de VPDB?
- 984 obtener el binario de la subaplicación de SADB basándose en SAV de uno de los VP extraídos
- 985 carga de la subaplicación al sensor con el uso del protocolo de carga de arranque
- 986 éxito de visualización
- 987 éxito/terminar

REIVINDICACIONES

1. Unidad (100) de programación para sensores (200) TPMS que comprende al menos un microcontrolador y/o microprocesador (130) y memoria (140) que almacena al menos una base de datos, comprendiendo la memoria:
- 5 una base (920) de datos de grupos de vehículos que comprende múltiples grupos (921, 922) de vehículos, comprendiendo cada grupo de vehículos al menos un vehículo (923, 924);
una base (930) de datos de par de versiones que comprende múltiples grupos de par (931, 932) de versiones, representando cada grupo de par de versiones un grupo de vehículos y comprendiendo al menos un par (933, 934) de versiones
caracterizada porque
10 cada par de versiones comprende una versión (935) de aplicación asociada y una versión (936) de subaplicación asociada;
una base (940) de datos de la aplicación que comprende múltiples versiones (941, 942) de la aplicación, cada versión de la aplicación comprende un archivo (943) binario asociado;
15 una base (950) de datos de subaplicación que comprende múltiples versiones (951, 952) de subaplicación, cada versión de subaplicación comprende un archivo (953) binario asociado.
2. Unidad (100) de programación de acuerdo con la reivindicación 1,
caracterizada porque
la unidad de programación comprende además un selector (960) de archivo binario para seleccionar al menos un archivo (943, 953) binario basándose en un vehículo seleccionado.
- 20 3. Unidad (100) de programación de acuerdo con la reivindicación 2,
caracterizada porque
el selector (960) de archivo binario selecciona además un archivo (943) binario para una aplicación y un archivo (953) binario para una subaplicación si un sensor TPMS objetivo no almacena este archivo (943) binario para una aplicación en la memoria.
- 25 4. Unidad (100) de programación de acuerdo con la reivindicación 2,
caracterizada porque
el selector (960) de archivo binario selecciona además un archivo (953) binario para una subaplicación si un sensor TPMS objetivo no almacena este archivo (953) binario para una subaplicación en la memoria.
- 30 5. Unidad de programación (100) de acuerdo con una cualquiera de las reivindicaciones anteriores,
caracterizada porque
la unidad de programación comprende además un cargador (180) de aplicaciones que primero solicita al menos una versión (941, 942) de aplicación y/o una versión (951, 952) de subaplicación de archivos binarios almacenados en la memoria no volátil (240) de un sensor TPMS.
- 35 6. Sensor (200) de supervisión de presión de neumáticos que comprende al menos una unidad (230) de procesamiento y una memoria (240) no volátil, comprendiendo la memoria no volátil, además, código ejecutable,
caracterizada porque
el código ejecutable comprende al menos un cargador (530) de arranque para cargar al menos un binario (943) de aplicación y/o un binario (953) de subaplicación en la memoria no volátil.
- 40 7. Sensor (200) de supervisión de presión de neumáticos de acuerdo con la reivindicación 6,
caracterizada porque
el cargador (530) de arranque está configurado para reenviar al menos una versión (941, 942) de aplicación y/o una versión (951, 952) de subaplicación de archivos binarios almacenados en la memoria (240) no volátil.
- 45 8. Sistema de configuración de sensores TPMS que comprende una unidad (100) de programación de acuerdo con una cualquiera de las reivindicaciones 1 a 5 y un sensor (200) de supervisión de presión de neumáticos de acuerdo con la reivindicación 6 o 7.
9. Procedimiento de configuración de un sensor (200) de supervisión de presión de neumáticos por una unidad (100) de programación que comprende las siguientes etapas en la unidad de programación:
- 50 a) obtener un tipo/marca/modelo de vehículo objetivo de un dispositivo (971) de entrada,
b) extraer un grupo de vehículos del vehículo objetivo de una base (972) de datos de grupos de vehículos,
c) si no se encuentra ningún grupo de vehículos, entonces, opcionalmente, se muestra un mensaje (974) de error y se termina el procedimiento,
d) extraer los pares de versiones del grupo de vehículos de una base de datos de par (976) de versiones,
e) transmitir una orden (977) OBTENER DESAFÍO al sensor de supervisión de presión de neumáticos y obtener una respuesta del sensor de supervisión de presión de neumáticos,
55 f) extraer de la respuesta la versión actual de la aplicación y la versión de la subaplicación en el sensor (978),
g) si la versión de la aplicación actual existe en el par de versiones de la etapa d) (979), proceder con la etapa k,
h) seleccionar el par de versiones con la mejor versión de aplicación de los pares de versiones extraídos y obtener

la versión de aplicación asociada y/o la versión (980) de subaplicación,

i) obtener un binario de una aplicación de una base de datos de la aplicación y/o un binario de una subaplicación de una base de datos de subaplicación basándose en la versión de aplicación y/o la versión (981) de subaplicación,

5 j) cargar el binario de una aplicación de una base de datos de la aplicación y/o el binario de una subaplicación en el sensor de supervisión de presión de neumáticos y proceder con la etapa n (982),

k) si la versión de la subaplicación actual existe en el par de versiones de la etapa d) (983), proceder con la etapa n,

l) obtener un binario de una subaplicación de la base de datos de subaplicación basándose en la versión (984) de subaplicación de uno de los pares de versiones,

10 m) cargar el binario de una subaplicación en el sensor de supervisión de presión de neumáticos (985),

n) opcionalmente mostrar un mensaje de éxito.

Fig. 1

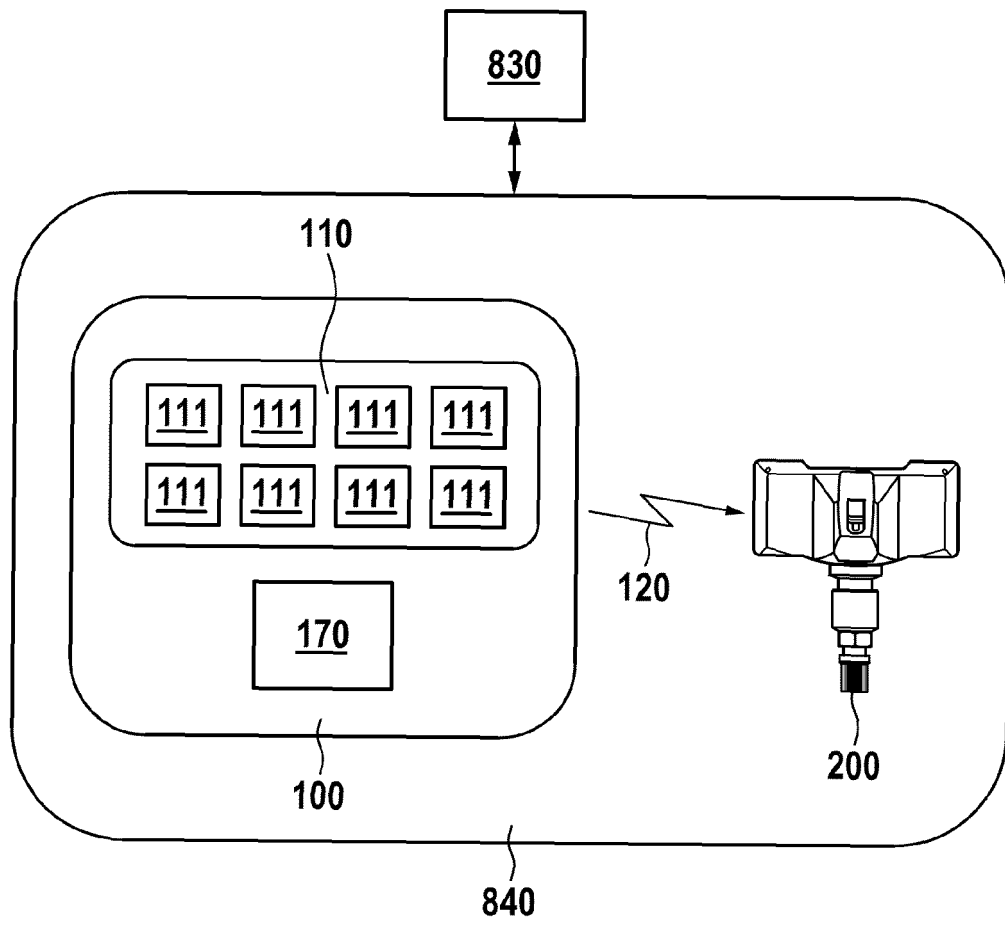


Fig. 2

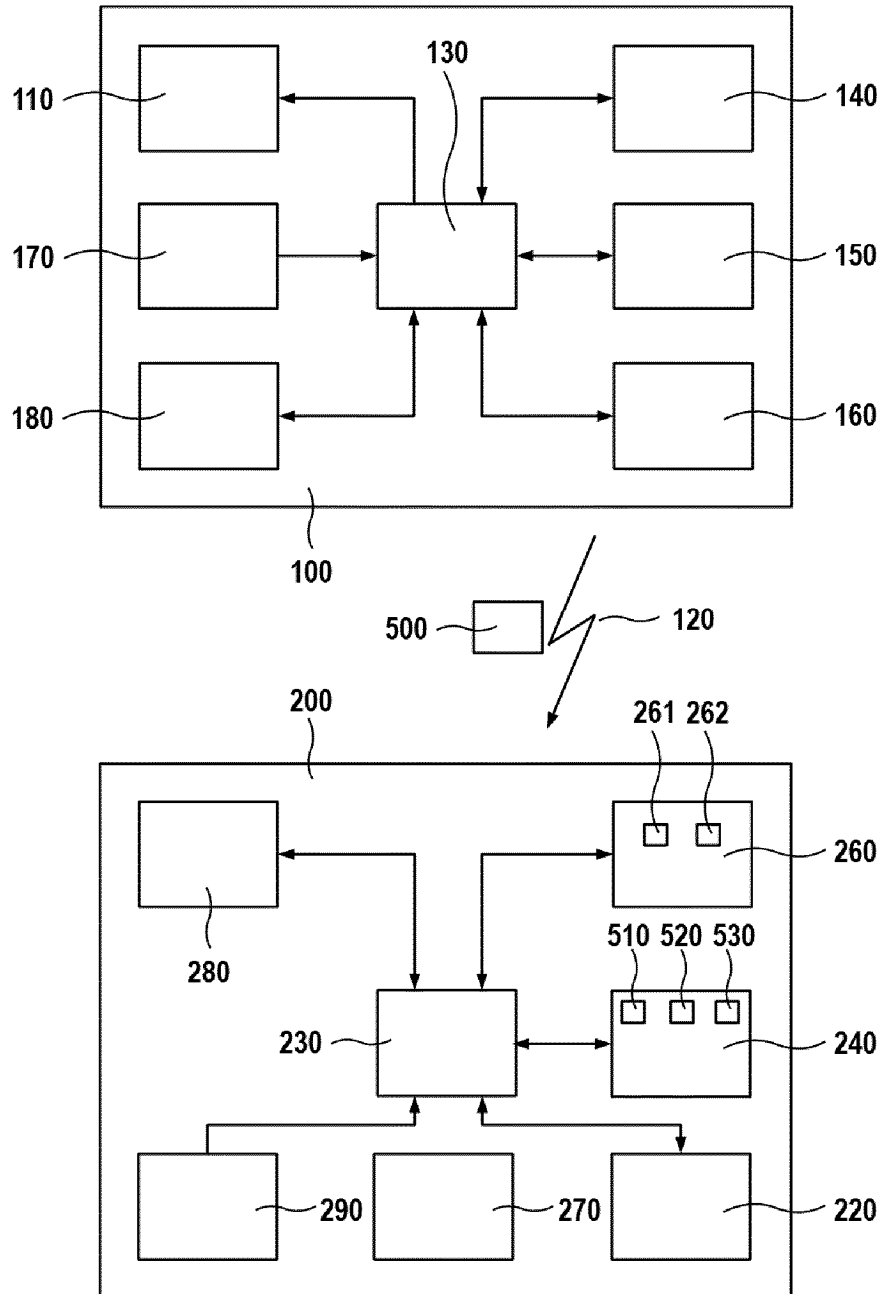


Fig. 3

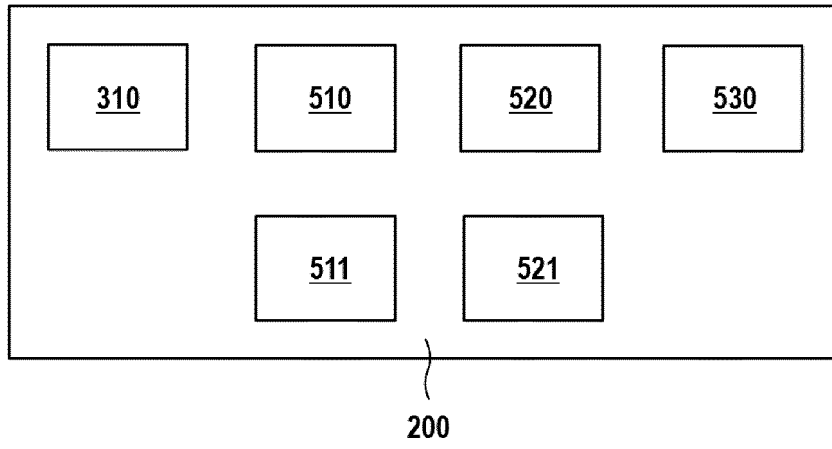


Fig. 4

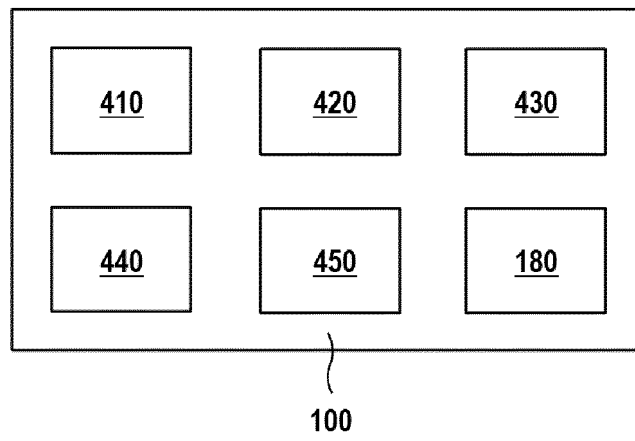


Fig. 5

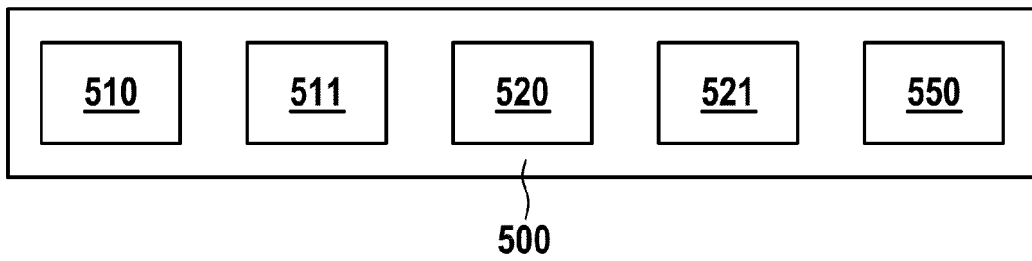


Fig. 6a

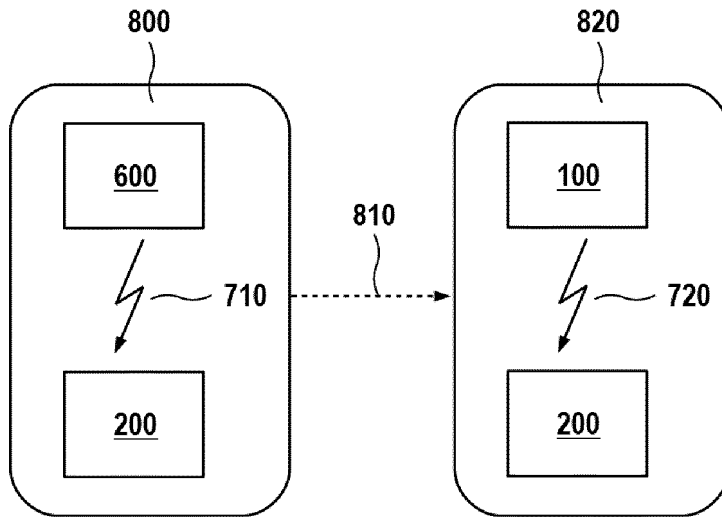


Fig. 6b

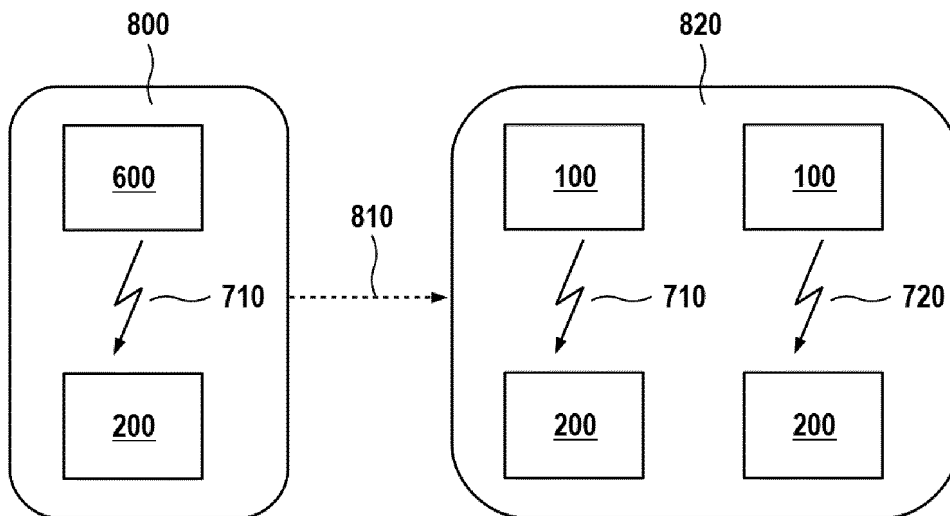


Fig. 7

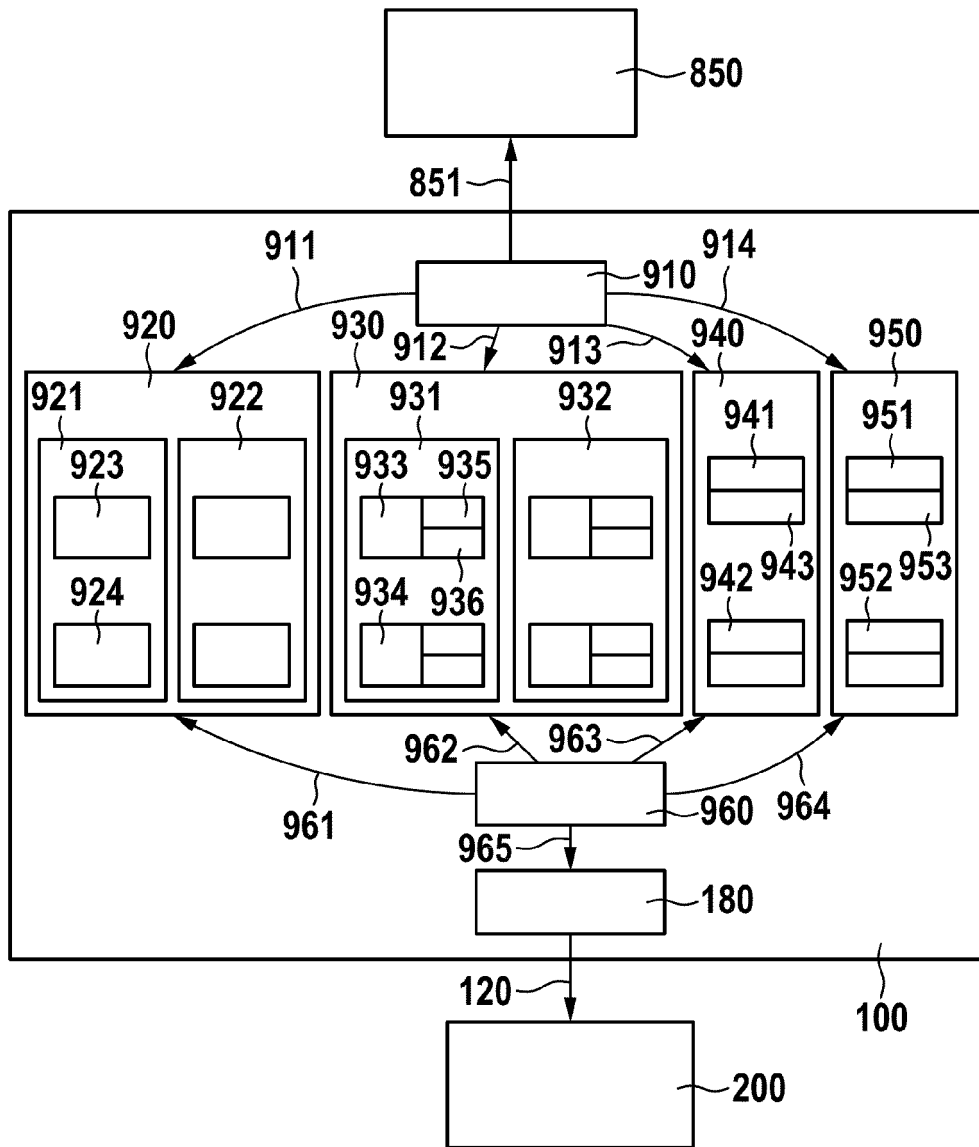


Fig. 8

