(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 493 068**

(51) Int. CI.:

G05B 9/03 (2006.01)
G05B 19/042 (2006.01)
G05B 23/02 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **17.02.2010 E 10708907 (0)**

(97) Fecha y número de publicación de la concesión europea: **06.08.2014 EP 2399174**

(54) Título: **Procedimiento y dispositivo para crear un programa de aplicación para un mando de seguridad**

(30) Prioridad:

23.02.2009 DE 102009011679

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
11.09.2014

(73) Titular/es:

PILZ GMBH & CO. KG (100.0%)
Felix-Wankel-Strasse 2
73760 Ostfildern, DT

(72) Inventor/es:

MOOSMANN, PETER;
REUSCH, MATTHIAS;
WALTER, HERBERT y
HECKEL, ANDREAS I

(74) Agente/Representante:

UNGRÍA LÓPEZ, Javier

DESCRIPCIÓN

Procedimiento y dispositivo para crear un programa de aplicación para un mando de seguridad

La presente invención se refiere a un procedimiento y un dispositivo para crear un programa de aplicación para un mando de seguridad que está diseñado para controlar una instalación automatizada con una pluralidad de sensores y una pluralidad de actores, en donde el programa de aplicación comprende una primera parte de programa, en la que se procesan a prueba de errores variables de programa relevantes para la seguridad, y comprende por lo menos una segunda parte del programa, en la que se procesan variables de programa no relevantes para la seguridad, en donde para las variables de programa no relevantes para la seguridad dentro de la segunda parte del programa no se requiere un procesamiento a prueba de errores.

Un mando de seguridad en el sentido de la presente invención es un aparato, o un dispositivo, que por ejemplo recibe las señales de entrada suministradas por sensores y a partir de las mismas genera señales de salida mediante vínculos lógicos y eventualmente otras etapas de procesamiento de señales o de datos. Las señales de salida pueden suministrarse entonces a los actores que dependiendo de las señales de entrada causan acciones o reacciones en una instalación controlada.

Un campo de aplicación preferido para tales mandos de seguridad es la vigilancia de pulsadores de parada de emergencia, mandos a dos manos, puertas de seguridad o barreras de luz en el ámbito de la seguridad de máquinas. Tales sensores se emplean, por ejemplo, para asegurar una máquina que durante el funcionamiento implica un peligro para las personas o los bienes materiales. Cuando se abre la puerta de seguridad o cuando se acciona el pulsador de parada de emergencia se genera respectivamente una señal que se suministra al mando de seguridad como señal de entrada. Como reacción a la misma, el mando de seguridad desconecta entonces la parte de la máquina que implica el peligro, por ejemplo, mediante un actor.

Algo característico en un mando de seguridad, en comparación con un mando "normal", es que el mando de seguridad siempre garantiza una condición segura de la instalación o máquina que implica el peligro, incluso si en ella misma o en un aparato relacionado con ella se presenta un error de funcionamiento. Por lo tanto, a los mandos de seguridad se aplican requisitos extremadamente altos en cuanto a su propia seguridad frente a los errores, lo cual tiene como consecuencia un dispendio considerable en su desarrollo y fabricación.

Normalmente, para el uso de los mandos de seguridad se requiere una autorización especial previa otorgada por las autoridades competentes, tales como por ejemplo las asociaciones o sindicatos profesionales o la asociación TÜV en Alemania (equivalente a la Inspección Técnica de Vehículos ITV en España). A este respecto, el mando de seguridad debe cumplir normas de seguridad especificadas, las cuales se detallan por ejemplo en la norma europea EN 954-1 o en una norma comparable, por ejemplo la norma IEC 61508 o la norma EN ISO 13849-1. Por lo tanto, en los siguientes entiende bajo un mando de seguridad un aparato o un dispositivo, respectivamente, que cumpla por lo menos con la categoría de seguridad 3 de la norma europea EN 954-1 previamente mencionada o cuyo Nivel de Integridad de Seguridad (Safety Integrity Level - SIL) por lo menos alcance el nivel 2 de acuerdo con la norma IEC 61508 previamente mencionada.

Un mando de seguridad programable le ofrece al usuario la posibilidad de adaptar las vinculaciones lógicas y, dado el caso, otras etapas de procesamiento de señales o datos mediante el uso de un software, el así llamado programa de aplicación, individualmente de manera correspondiente a sus necesidades. De esto resulta una gran flexibilidad en comparación con soluciones anteriores, en las que las vinculaciones lógicas se producían a través de un cableado definido entre diferentes módulos de seguridad. Un programa de aplicación se puede crear, por ejemplo, mediante el uso de un ordenador personal comercial (PC) y empleando programas de software correspondientemente instalados.

En instalaciones que corresponden al estado actual de la técnica, normalmente se usan dos mandos programables. Un mando de seguridad para cumplir con las tareas de seguridad y un mando estándar para cumplir con las tareas estándar. En casos aislados también se usa un mando común, a través del cual se cumplen todas las tareas tanto estándar como también de seguridad. En ambas formas de realización, el cumplimiento de las funciones de seguridad se realiza a través de un procesamiento a prueba de errores de variables de programa relevantes para la seguridad. Para ello, a través de sensores de seguridad, es decir, sensores diseñados a prueba de errores, se registran magnitudes relevantes para la seguridad y se suministran por medio de señales de entrada de mando relevantes para la seguridad al mando de seguridad o al mando común. En el mismo se determinan valores para señales de salida de mando relevantes para la seguridad usando variables de programa relevantes para la seguridad. Con estas señales de salida de mando se activan actores de seguridad, es decir, actores configurados a prueba de errores, para realizar acciones relevantes para la seguridad. La realización de tareas estándar se lleva a cabo mediante el procesamiento de variables de programa no relevantes para la seguridad, para los que no se requiere un procesamiento a prueba de errores. Para ello, a través de sensores estándar se registran magnitudes no relevantes para la seguridad, que también se denominan como magnitudes relevantes para el proceso. Mediante señales de entrada de mando no relevantes para la seguridad, dichas magnitudes se suministran al mando estándar o al mando común. Usando variables de programa no relevantes para la seguridad, en el mismo se determinan

señales de salida de mando no relevantes para la seguridad. Con estas señales de salida de mando se activan actores estándar que entonces se realizan acciones no relevantes para la seguridad.

En ambas formas de realización, para llevar a cabo las funciones de seguridad se requieren sensores de alto coste, debido a que los mismos están diseñados a prueba de errores. Para esto ni se pueden emplear sensores estándar de menor coste, ya que los mismos no están configurados a prueba de errores, ni tampoco se pueden usar variables de programa no relevantes para la seguridad.

Tanto el uso de mandos de seguridad programables como también el uso de un mando común para realizar todas las funciones estándar y funciones de seguridad pueden contribuir a incrementar la flexibilidad en la creación de programas de aplicación y, por lo tanto, en la realización de mandos de seguridad. Sin embargo, los mismos todavía no son óptimos en lo referente a los costes – los costes para la elaboración del programa de aplicación y los costes de los componentes requeridos para la realización del mando de seguridad.

El documento US 2007/0255429 A1 desvela un sistema de mando para el control automático de una instalación, en donde el mando procesa tanto variables de programa relevantes para la seguridad como también variables de programa (estándar) no relevantes para la seguridad. De manera correspondiente, el mando incluye partes de programa a prueba de errores y las así llamadas partes de programa estándar. En el mando, las partes de programa a prueba de errores y las partes de programa estándar se ejecutan de manera completamente separada. El sistema de mando incluye unidades de entrada/salida escalonadas, a las cuales están conectados sensores y/o actores. Los actores son controlados conjuntamente por las partes de programa a prueba de errores y por las partes de programa estándar, a la manera de un vínculo UND. Sin embargo, allí no se prevé el uso o la evaluación de variables de programa no relevantes para la seguridad en la parte de programa a prueba de errores.

El documento US 2009/0030534 A1 desvela un procedimiento para programar un mando de seguridad, en donde se usa una superficie de programa gráfica para crear primero un esquema de cableado, mediante el cual se conecta el mando de seguridad con los sensores y actores. A continuación, en base descripciones de elementos que se encuentran almacenados en el mando de seguridad y/o en los sensores y actores, puede llevarse a cabo una creación parcialmente automatizada del programa. Para este propósito, en un aparato programador existe una biblioteca de partes de programa que se seleccionan dependiendo de las descripciones de elementos y se combinan para formar un programa.

El documento US 5.504.473 desvela un procedimiento para analizar señales de un detector de movimientos redundante. En el contexto del análisis, el sistema suministra una versión simplificada de una señal actual.

El documento US 2004/0064205 A1 describe otro procedimiento y un dispositivo correspondiente para programar un mando de seguridad. También en este caso, la programación se realiza mediante el uso de una superficie de programación gráfica, en la que un programador puede crear dependencias lógicas con ayuda de símbolos de programa gráficos.

Por consiguiente, un objetivo de la presente invención consiste en perfeccionar un procedimiento y un dispositivo del tipo previamente mencionado para reducir adicionalmente los costes de la creación de un programa de aplicación con funciones relevantes para la seguridad y por lo tanto también de la realización de un mando de seguridad, aumentando al mismo tiempo la flexibilidad, haciendo posible así una programación más fácil, rápida y clara de un mando de seguridad.

Este objetivo se logra a través de un procedimiento del tipo inicialmente mencionado, en el que se realizan las siguientes etapas:

- Definir un número de variables de programa relevantes para la seguridad;
- Definir un número de variables de programa no relevantes para la seguridad;
- Seleccionar una variable de programa relevante para la seguridad entre un número de variables de programa relevantes para la seguridad;
- Seleccionar una primera variable de programa no relevante para la seguridad entre un número de variables de programa no relevantes para la seguridad, en donde a la primera variable de programa no relevante para la seguridad se le asigna repetidamente un valor momentáneo durante la ejecución del programa de aplicación;
- Definir por lo menos una condición de asignación que se procesa durante la ejecución del programa de aplicación;
- Definir una asignación que asigna la primera variable de programa no relevante para la seguridad seleccionada a la variable de programa relevante para la seguridad seleccionada, en donde el valor momentáneo de la primera variable de programa no relevante para la seguridad seleccionada durante la ejecución del programa de

aplicación se asigna a la variable de programa relevante para la seguridad seleccionada dependiendo de la condición de asignación.

El objetivo de la invención se logra adicionalmente a través de un dispositivo del tipo inicialmente mencionado que presenta las siguientes unidades: Unidades para definir un número de variables de programa relevantes para la seguridad y para seleccionar una variable de programa relevante para la seguridad entre un número de variables de programa relevantes para la seguridad, para definir un número de variables de programa no relevantes para la seguridad y seleccionar una primera variable de programa no relevante para la seguridad entre el número de variables de programa no relevantes para la seguridad, en donde a la primera variable de programa no relevante para la seguridad durante la ejecución del programa de aplicación se le asigna repetidamente un valor momentáneo, para definir por lo menos una condición de asignación, la cual se procesa durante la ejecución del programa de aplicación, y para definir una asignación que asigne la primera variable de programa no relevante para la seguridad seleccionada a la variable de programa relevante para la seguridad seleccionada, en donde el valor momentáneo de la variable de programa no relevante para la seguridad seleccionada durante la ejecución del programa de aplicación se le asigna a la variable de programa relevante para la seguridad seleccionada dependiendo de la condición de asignación.

La idea subyacente al nuevo procedimiento y al nuevo dispositivo consiste en que durante la ejecución del programa de aplicación el valor momentáneo de una primera variable de programa no relevante para la seguridad seleccionada se le asigna una variable de programa relevante para la seguridad seleccionada dependiendo de una condición de asignación y por consiguiente, expresado en términos generales, transformar una variable de programa no relevante para la seguridad en una variable de programa relevante para la seguridad. De esta manera se incrementa la flexibilidad en la creación de un programa de aplicación. Para cumplir las funciones de seguridad, adicionalmente a las variables de programa relevantes para la seguridad y las magnitudes relevantes para la seguridad representadas por las mismas, ahora también se dispone de variables de programa no relevantes para la seguridad y magnitudes no relevantes para la seguridad representadas por éstas. De esta manera se incrementa el número de soluciones realizables desde el punto de vista técnico de la programación para las distintas funciones de seguridad.

Al incremento de la flexibilidad también contribuye que la condición de asignación requerida puede ser definida libremente por el creador del programa de aplicación. De esta manera, para la transformación se pueden usar diferentes variables de programa no relevantes para la seguridad y las magnitudes físicas representadas por las mismas.

La asignación de un valor momentáneo de una primera variable de programa no relevante para la seguridad seleccionada a una variable de programa relevante para la seguridad seleccionada tiene como resultado que para el suministro de los valores momentáneos de la variable de programa relevante para la seguridad no se requiere un sensor diseñado a prueba de errores. La magnitud física que representa a la variable de programa relevante para la seguridad por consiguiente también puede ser registrada usando un sensor que no esté diseñado a prueba de errores. Esto contribuye a la reducción de los costes.

Para la transformación de una variable de programa no relevante para la seguridad en una variable de programa relevante para la seguridad en principio sólo es necesario definir una condición de asignación y una asignación. Esto permite una programación clara, fácil y por lo tanto rápida de un mando de seguridad, lo que también contribuye a aumentar la seguridad frente a los errores.

Por consiguiente, el objetivo previamente mencionado se ha logrado en su totalidad.

En otra forma de realización de la presente invención, la condición de asignación representa una consulta, a través de la que se determina si una primera variable de programa no relevante para la seguridad seleccionada cumple con los requisitos de confiabilidad exigidos para variables de programa relevantes para la seguridad.

Con esta medida se asegura que a pesar de la transmisión de valores momentáneos de una variable de programa no relevante para la seguridad a una variable de programa relevante para la seguridad sea posible una operación a prueba de errores del mando de seguridad. Para cumplir las funciones de seguridad por lo tanto también es posible usar adicionalmente a las variables de programa relevantes para la seguridad y las magnitudes físicas representadas por las mismas también las variables de programa relevantes para la seguridad y las magnitudes físicas representadas por éstas. Adicionalmente, para incrementar la seguridad frente a los errores, el valor de la primera variable de programa no relevante para la seguridad seleccionada sólo se asigna a la variable de programa relevante para la seguridad seleccionada si se ha cumplido la condición de asignación.

Con miras a un sistema técnico, la confiabilidad se representa como la confiabilidad en lo referente a la prestación correcta y continúa de un servicio acordado dentro de límites de probabilidad y segmentos de tiempo definidos. La confiabilidad por lo tanto es un grado de la confianza que parece justificarse debido a una reducida probabilidad de fracaso, una alta probabilidad de exactitud o una reducida tasa de paradas o errores.

El requisito de confiabilidad exigido a las variables de programa relevantes para la seguridad de cine en primer lugar una probabilidad de exactitud para sus valores momentáneos. El requisito de confiabilidad puede especificar, por ejemplo, que el valor momentáneo de una variable de programa relevante para la seguridad coincida en todo caso con una desviación insignificantemente pequeña con un valor real de una magnitud física. Normalmente, para el cumplimiento de este requisito de confiabilidad se emplean sensores diseñados a prueba de errores. Por lo tanto, los errores en el registro de una magnitud física pueden ser reconocidos y se puede cumplir con la probabilidad de exactitud exigida. Si el requisito de exactitud exigido se cumple o no, se determina ventajosamente mediante la evaluación del valor momentáneo de la primera variable de programa no relevante para la seguridad seleccionada.

En otra forma de realización adicional de la presente invención, como condición de asignación se define una consulta de plausibilidad, mediante la cual se comprueba si la primera variable de programa no relevante para la seguridad seleccionada y una variable de programa adicional seleccionada son compatibles entre sí.

Esta medida permite una gran flexibilidad, ya que la selección discrecional de una variable de programa adicional ofrece un amplio margen para la definición de una consulta de plausibilidad. Además, una consulta de plausibilidad se puede realizar sin mayor esfuerzo mediante el uso de dos variables de programa.

Mediante la comprobación de si las dos variables de programa son compatibles entre sí, se determina si las dos variables de programa son coherentes entre sí. En otras palabras: se determina si las dos variables de programa coinciden. Esta comprobación se basa en la siguiente suposición: en el caso ideal, tanto la primera variable de programa no relevante para la seguridad seleccionada como también la variable de programa adicional seleccionada representan la misma magnitud física, por ejemplo una velocidad o una distancia. Pero también para el caso de que las dos variables de programa representen magnitudes físicas diferentes, las cuales sin embargo se correlacionan a través de una regularidad física, se puede realizar igualmente esta comprobación. En este caso, sin embargo, para una de las dos variables de programa se requiere un cálculo de conversión de sus valores momentáneos tomando en cuenta la regularidad física.

De manera ventajosa, la comprobación se realiza mediante la evaluación de los valores momentáneos de las dos variables de programa.

En la evaluación de los valores momentáneos se comprueba si los valores momentáneos de las dos variables de programa cumplen un criterio definido. Para esto en principio entran en consideración numerosos criterios de orientación diferente. Por una parte se puede usar un criterio que esté dirigido a los valores momentáneos como tales. En este caso se comprueba si los valores momentáneos como tales muestran un comportamiento compatible. Se puede verificar, por ejemplo, si una diferencia formada por los dos valores momentáneos es más pequeña que un valor de umbral definido o si un cociente formado por los dos valores momentáneos se ubica dentro de un intervalo definido, el cual se sitúa alrededor del valor 1. Por otra parte, se puede aplicar un criterio que esté dirigido al comportamiento cronológico de los valores momentáneos. Se comprueba, por lo tanto, si los valores momentáneos muestran un comportamiento cronológicamente compatible. En este caso, para las dos variables de programa se determina una derivación cronológica, de manera ventajosa aproximadamente en forma de un cociente de diferencias. También en el caso de este criterio se puede evaluar de manera correspondiente, según se ha descrito previamente, una diferencia o un cociente. El uso de un criterio dirigido al comportamiento cronológico tiene la ventaja de que es posible determinar en un momento muy temprano una desviación entre las dos variables de programa, mucho antes de que se haga evidente una desviación considerable en los valores momentáneos mismos. Ventajosamente, los valores momentáneos de las dos variables de programa, que se usan para formar la diferencia o el cociente, se ubican respectivamente por pares en un determinado punto de tiempo, antes o dentro de un pequeño intervalo de tiempo referido a ese punto de tiempo.

De manera ventajosa, se combinan varios criterios entre sí. De esta forma es posible establecer consultas de plausibilidad confiables, lo cual resulta en una alta seguridad frente a los errores.

Preferentemente, en el caso de la variable de programa adicional seleccionada se trata de una variable de programa no relevante para la seguridad. En este caso se puede generar una variable de programa relevante para la seguridad individualmente mediante el uso de variables de programa no relevantes para la seguridad, lo cual resulta en una flexibilidad muy grande.

En otra forma de realización de la medida previamente mencionada tanto en el caso de la primera variable de programa no relevante para la seguridad seleccionada como también en la variable de programa no relevante para la seguridad adicional seleccionada se trata respectivamente de una variable de entrada de programa, en donde a la primera variable de programa no relevante para la seguridad seleccionada se le asigna un valor momentáneo que representa un valor de una señal de sensor determinada por medio de un primer sensor, y en donde a la variable de programa no relevante para la seguridad adicional seleccionada se le asigna un valor momentáneo que representa el valor de una segunda señal de sensor determinada con un segundo sensor. Por lo tanto es posible de una manera simple, mediante el uso de dos sensores con un diseño no a prueba de errores, transformar una variable de programa no relevante para la seguridad en una variable de programa relevante para la seguridad.

En otra forma de realización de la medida previamente mencionada, los dos sensores ventajosamente están realizados de forma diversificada. Por lo tanto, los dos sensores captan la misma magnitud física, pero a través de principios de medición diferentes. Por ejemplo, la magnitud física puede ser registrada por un sensor en base a la tensión y con el otro sensor en base a la corriente. Esta medida permite alcanzar un grado muy elevado de seguridad frente a los errores.

Mediante la selección o combinación apropiada de los dos sensores se pueden compensar además las influencias externas como por ejemplo las variaciones de temperatura o similares.

Alternativamente, como variable de programa adicional seleccionada también se puede usar una variable de programa relevante para la seguridad, lo cual contribuye a un incremento adicional de la flexibilidad.

En otra forma de realización de la presente invención se define como condición de asignación una consulta de plausibilidad, mediante la cual se comprueba si la primera variable de programa no relevante para la seguridad seleccionada cumple un criterio de comparación, en donde el criterio de comparación representa una propiedad característica de la primera variable de programa no relevante para la seguridad seleccionada.

Esta medida tiene la ventaja de que la determinación de si se cumple o no el requisito de confiabilidad se puede realizar únicamente a través de la primera variable de programa no relevante para la seguridad seleccionada. No se requiere otra variable de programa. Esta medida permite una gran flexibilidad y una fácil realización. Para el caso de que la primera variable de programa no relevante para la seguridad seleccionada sea una variable de entrada de programa, para la transformación de una variable de programa no relevante para la seguridad en una variable de programa relevante para la seguridad no se requiere, por lo tanto, ningún sensor adicional. Esto permite una transformación muy favorable en cuanto al coste.

En general son posibles diferentes criterios de comparación. Por una parte se puede usar un criterio de evaluación dirigido a los valores momentáneos como tales. Se trata, por ejemplo, de un intervalo dentro del cual se deben ubicar conforme a lo esperado los valores momentáneos de la primera variable de programa no relevante para la seguridad seleccionada. Este intervalo puede ser especificado a través de un valor máximo generalmente esperado y a través de un valor mínimo generalmente esperado. Por otra parte se puede usar un criterio de comparación dirigido al comportamiento cronológico de los valores momentáneos. Por ejemplo, se trata de un intervalo dentro del cual se debe ubicar conforme a lo esperado el gradiente de tiempo determinado para valores momentáneos crecientes. Para los valores momentáneos decrecientes se puede proceder de manera correspondiente.

De manera más ventajosa, el criterio de comparación se compone de varias consultas individuales. Así, por ejemplo, el criterio de comparación puede incluir tanto especificaciones para los valores momentáneos como tales, así como también especificaciones para el comportamiento cronológico de los valores momentáneos. De esta manera es posible comprobar si los valores momentáneos, partiendo de un primer nivel de valores, ocupan un segundo nivel de valores dentro de un período de tiempo definido y si el valor momentáneo existente después de transcurrir dicho período de tiempo se ubica dentro de un intervalo definido. Algo correspondiente también es posible para valores momentáneos decrecientes. También se puede definir un criterio de comparación para valores momentáneos que primero aumentan y después se reducen.

En otra forma de realización de la invención, el programa de aplicación comprende una pluralidad de instrucciones de transformación, en donde las instrucciones de transformación representan la asignación y la condición de asignación, en donde por lo menos una parte de las instrucciones de transformación están contenidas en la primera parte de del programa.

Por lo tanto, las instrucciones de transformación están contenidas en la parte del programa en la que se procesan a prueba de errores las variables de programa relevantes para la seguridad y que contiene instrucciones de mando de seguridad para controlar los actores de seguridad. En consecuencia, también en la transformación de una variable de programa no relevante para la seguridad en una variable de programa relevante para la seguridad está dada la seguridad frente a los errores. Esto significa que la variable de programa relevante para la seguridad generada por la transformación es segura frente a los errores, mientras que la variable de programa no relevante para la seguridad continúa no siendo a prueba de errores. De manera ventajosa, todas las instrucciones de transformación están incluidas en la primera parte del programa.

En otra forma de realización adicional de la invención, una primera porción de código que representa la asignación y una segunda porción de código que representa las condiciones de asignación se reúnen en un módulo de programa.

Esta medida afecta al código fuente del programa de aplicación. En consecuencia, las dos porciones de código se resumen en un bloque en el código fuente. Por lo tanto, en el código fuente existe un sitio de transformación dedicado, en el que se resumen en las instrucciones de transformación. Tanto la asignación como también la condición de asignación son relevantes para la seguridad frente a los errores y por lo tanto son comprobadas también en el marco de un procedimiento de homologación a ser realizado por una autoridad inspectora. La formación de bloques dentro del código fuente simplifica el procedimiento de homologación. Preferentemente, el

módulo de programa está incluido en la primera parte de programa.

Un sitio de transformación dedicado le ofrece al creador de un programa de aplicación la posibilidad de obtener sin demasiado esfuerzo una idea de dónde se usa una variable de programa no relevante para la seguridad para generar una variable de programa relevante para la seguridad o, respectivamente, qué variable de programa relevante para la seguridad se deriva de una variable de programa no relevante para la seguridad. Si en un programa de aplicación están previstas varias transformaciones, es decir, si varias variables de programa no relevantes para la seguridad se transforman respectivamente en una variable de programa relevante para la seguridad, el programa de aplicación incluye varios módulos de programa independientes. Para cada transformación existe un sitio de transformación dedicado.

Una ventaja adicional consiste en que el resumen en un módulo de programa permite un encapsulamiento. Un encapsulamiento tiene como resultado que el módulo de programa puede ser reutilizado cuantas veces se quiera a través de una correspondiente llamada como si se tratara de una función.

El resumen de las dos porciones de código en un módulo de programa es realizado ventajosamente de forma automática por el programa de ordenador a través de cuyo uso se crea el programa de aplicación.

En otra forma de realización de la invención, el programa de aplicación se crea mediante el uso de un programa de ordenador que se ejecuta en un ordenador, en donde el programa de ordenador comprende un módulo de visualización que permite que durante la creación del programa de aplicación se pueda visualizar un código fuente para el programa de aplicación mediante una pluralidad de símbolos gráficos de código de fuente que representan a dicho código fuente en una unidad de visualización conectada al ordenador, en donde los símbolos gráficos del código fuente están realizados en una forma de representación básica, comprendiendo dicha pluralidad de símbolos gráficos del código fuente un número de símbolos gráficos de código de transformación que representan un código de transformación incluido en el código fuente, en donde el código de transformación comprende una primera porción de código que representa la asignación y una segunda porción de código que representa la condición de asignación, comprendiendo el programa de ordenador adicionalmente un módulo de reconocimiento, mediante el cual se reconoce por lo menos uno de las dos porciones de código, en donde al existir una porción de código reconocida el módulo de visualización hace que por lo menos un símbolo gráfico de código de transformación, que está incluido en el número de símbolos gráficos de código de transformación, se represente con una forma de representación modificada con respecto a la forma de representación básica.

La representación de un símbolo de transformación gráfico modificado produce una marcación en la representación del programa de aplicación. Por lo tanto, el creador de un programa de aplicación puede localizar sin problemas el volumen incluido en el programa de aplicación que representa el código de transformación, es decir, el sitio de transformación dedicado. Aquellos volúmenes del programa de aplicación que provienen de la transformación de una variable de programa no relevante para la seguridad en una variable de programa relevante para la seguridad, se pueden reconocer, por lo tanto, de forma inmediata. Por consiguiente, en caso de ser necesario, el creador del programa puede someterlos a una comprobación crítica, lo cual representa una ventaja también en vista de un procedimiento de homologación.

La representación de un símbolo de transformación gráfico modificado y por consiguiente la marcación en la representación del programa de aplicación es realizada de manera automática por el respectivo programa de ordenador a través de cuyo uso se crea el programa de aplicación. El creador del programa de aplicación no tiene que realizar ningún acto adicional, más allá de lo que normalmente se requiere para crear un programa de aplicación.

Ventajosamente, se representan en una forma de representación modificada aquellos símbolos de transformación gráficos que representan la primera porción de código. De manera complementaria, también se pueden representar en forma de representación modificada los símbolos de transformación gráficos que representan la segunda porción del código.

Son imaginables en varias formas de realización para representar el código fuente para un programa de aplicación a través de una pluralidad de símbolos gráficos. En una primera forma de realización, el programa de aplicación se crea a través de entradas textuales. Por lo tanto, el programa de aplicación está presente, por ejemplo, en forma de un texto estructurado. Los símbolos gráficos del código fuente corresponden aquí las diferentes letras, cifras y, dado el caso, símbolos especiales. En esta forma de realización es imaginable representar de forma modificada por lo menos algunas letras que representan la primera porción del código. Por ejemplo, esto se puede lograr si las letras se representan en un color modificado en comparación con la forma de representación básica. De manera alternativa o complementaria, dichas letras también se pueden representar con un tipo de letra modificado en comparación con la forma de representación básica, por ejemplo en letras cursivas o en negritas. También es imaginable que a dichas letras se aplique, por ejemplo, una sangría o que a la respectiva línea que contenga las primeras letras se antepongan símbolos correspondientes. Así, por ejemplo, es posible anteponer asteriscos o algún texto en particular, por ejemplo "¡Sigue un código de transformación relevante para la seguridad!". La anteposición de caracteres se realiza mediante la inserción de una línea adicional. De forma complementaria, los caracteres

correspondientes también se pueden insertar detrás de la respectiva línea que contiene las últimas letras.

En una forma de realización preferida, con el módulo de reconocimiento se reconoce la primera porción del código.

El reconocimiento de la primera porción del código y la marcación relacionada con ello de dicha porción del código tiene la ventaja de que la primera variable de programa no relevante para la seguridad seleccionada que participa en la transformación y la variable de programa relevante para la seguridad seleccionada en la representación del programa de aplicación se pueden determinar rápidamente. Por lo tanto, el creador de un programa de aplicación se puede informar ampliamente. Él podrá reconocer de inmediato que variable de programa constituye la base para la transformación y cuál será la variable de programa generada por la transformación.

De manera ventajosa, la primera porción del código se reconoce por el hecho de que en una porción de código coherente se le una variable de programa no relevante para la seguridad y que el valor momentáneo leído en este acto se asigna a una variable de programa relevante para la seguridad.

Mediante el reconocimiento de la primera porción de código, las variables de programa que participan en la transformación son en sí conocidas. Con miras a un apoyo más extenso para el creador de un programa de aplicación, esto permite marcar volúmenes adicionales del programa de aplicación. Preferentemente, también se pueden marcar aquellos volúmenes que representan la condición de asignación. Esto produce una marcación aún más llamativa en la representación del programa de aplicación.

En una forma de realización particularmente ventajosa, se reconocen las dos porciones del código. En este caso, el por lo menos un símbolo de código de transformación gráfico se representa en la forma de representación modificada con respecto a la forma de representación básica, en caso de existir ambas porciones de código. Adicionalmente, también cabe la posibilidad de representar todos los símbolos de código de transformación mediante la forma de representación modificada.

Con miras a la creación de un programa de aplicación y la representación del programa de aplicación vinculada a la misma, es posible una forma de realización adicional. En este caso, el programa de aplicación se realiza en forma de un así llamado diagrama de bloques de función mediante el uso de símbolos gráficos. En esta forma de realización se usan diferentes tipos de símbolos gráficos, por ejemplo rectángulos, que representan bloques de función seleccionados, letras y símbolos de reserva de espacio. En esta forma de realización, los módulos de programa utilizables, y por consiguiente seleccionables, están guardados en una base de datos. Para cada uno de estos módulos de programa existe un pequeño símbolo representativo dispuesto en una barra de iconos. Cuando se selecciona uno de estos símbolos representativos mediante una función de "arrastrar y soltar" y se deposita en una región prevista para ello en una superficie gráfica, en el sitio correspondiente de la superficie gráfica se representará entonces un bloque de función que representa el módulo de programa seleccionado. En la base de datos también se pueden incluir módulos de programa creados durante la elaboración del programa de aplicación. Por ejemplo, el módulo de programa que representa la asignación y las condiciones de asignación.

En consecuencia, en una forma de realización adicional de la presente invención, el programa de aplicación se crea usando un programa de ordenador que se está ejecutando en un ordenador, en donde la creación del programa de aplicación se realiza mediante la selección de un número de módulos de programa entre una pluralidad de módulos de programa predefinidos, en donde el programa de ordenador comprende un módulo de visualización que permite la visualización de una pluralidad de símbolos de módulos de programa gráficos en una unidad de visualización conectada a un ordenador, en donde la pluralidad de símbolos de módulos de programa gráficos comprende un primer número de símbolos de módulos de programa gráficos que representan los módulos de programa previamente definidos, así como un segundo número de símbolos de módulos de programa gráficos que representan el número de módulos de programas seleccionados, en donde por lo menos el primer número de símbolos de módulos de programa gráficos se realizan en una forma de representación básica, en donde el número de módulos de programa previamente definidos comprende un módulo de programa creado durante la creación del programa de aplicación, el cual representa la asignación y la condición de asignación, en donde el módulo de programa se representa por al menos un símbolo del módulo de programa gráfico modificado, en donde el módulo de visualización hace que el símbolo del módulo de programa gráfico modificado se visualice en una forma de representación modificada con respecto a la forma de representación básica.

A través de esta medida, el creador de un programa de aplicación puede referirse de manera confiable a aquellos volúmenes del programa de aplicación que se derivan de la transformación y que junto a los volúmenes del programa de aplicación que representan las funciones de seguridad propiamente dichas, son relevantes en cuanto a la seguridad frente a los errores. A este respecto, se deben observar en particular los volúmenes que afectan a la transformación, debido a que aquí en la primera parte del programa se emplean variables de programa que se derivan de variables de programa no relevantes para la seguridad. En lo que respecta a la extensión de los símbolos de módulos de programa gráficos modificados, es imaginable que se modifique solamente el símbolo de reserva de posición a través del cual se represente el respectivo módulo de programa que representa la asignación y la condición de asignación. De manera complementaria, todos los bloques de funciones creados en el programa de aplicación, que representan este módulo de programa, también pueden ser modificados. Ventajosamente, también

en esta forma de realización se reconoce la primera porción de código.

De manera ventajosa, se usa el símbolo de módulo de programa gráfico modificado cuando para el módulo de programa se reconoce que el mismo abarca tanto la primera porción de código como también una segunda porción de código. Mediante el uso de un símbolo de módulo de programa gráfico modificado, el creador de un programa de aplicación es advertido sobre un sitio de transformador potencial. Si se modifican todos los bloques funcionales creados en el programa de aplicación que representan dicho módulo de programa, esto produce una marcación de los sitios de transformación concretos. También cuando el programa de aplicación se crea mediante entradas textuales, la marcación resulta en que se resalten los sitios de transformación concretos.

En una forma de realización adicional de la presente invención, durante la ejecución del programa de aplicación partiendo de la primera variable de programa no relevante para la seguridad seleccionada se genera una variable de programa no relevante para la seguridad duplicada.

Esta forma de realización tiene la ventaja de que para cada uno de los dos canales de procesamiento, que tienen que estar presentes para un procesamiento libre de errores de las variables de programa relevantes para la seguridad en un mando de seguridad, se provee una variable de programa no relevante para la seguridad independiente que se transforma respectivamente de forma autónoma en una variable de programa relevante para la seguridad.

La variable de programa no relevante para la seguridad duplicada se produce ventajosamente mediante la lectura del valor momentáneo de la primera variable de programa no relevante para la seguridad seleccionada en dos áreas de memoria independientes, de las cuales una está prevista para la variable de programa no relevante para la seguridad y la otra para la variable de programa no relevante para la seguridad duplicada.

Ventajosamente, la variable de programa no relevante para la seguridad duplicada es generada de forma automática por el programa de ordenador que se usa para la creación del programa de aplicación. El creador del programa de aplicación no tiene que realizar ninguna otra acción, aparte de las que normalmente se requieren para la creación de un programa de aplicación.

La generación automática de las variables de programa no relevantes para la seguridad duplicada es se desarrolla ventajosamente de la siguiente manera: Mediante el módulo de visualización se crea un código de fuente para el programa de aplicación. Si existe una porción de código reconocida, el módulo de reconocimiento genera un código de duplicación que se integra en el código fuente. Basado en este código de duplicación, durante la posterior ejecución del programa de aplicación se genera de forma automática la variable de programa no relevante para la seguridad duplicada a partir de la primera variable de programa no relevante para la seguridad seleccionada. Sin embargo, también es posible una forma de realización, en la que no sea necesaria una creación explícita de un código de duplicación. En este caso, en la memoria fija (*firmware*) que corre en el mando de seguridad se encuentra almacenada una rutina que realiza de forma automática una duplicación o que establece las instrucciones requeridas para ello, respectivamente, si debido a una asignación se debe procesar una variable de programa no relevante para la seguridad en la primera parte del programa.

En otra forma de realización de la medida previamente mencionada, tanto para la primera variable de programa no relevante para la seguridad seleccionada como también para la variable de programa no relevante para la seguridad duplicada se procesa la condición de asignación.

Por lo tanto, se fija la siguiente secuencia: En un primer paso se genera la variable de programa no relevante para la seguridad duplicada. En un segundo paso se comprueba entonces para las dos variables de programa si respectivamente se ha cumplido un requisito de confiabilidad exigido para variables de programa relevantes para la seguridad. A través de esta secuencia de procesamiento se pueden reconocer los errores que se presenten en la duplicación, lo cual contribuye a incrementar la seguridad frente a los errores.

En otra forma de realización de la presente invención, la primera variable no relevante para la seguridad seleccionada es una variable de entrada de programa, en donde el valor momentáneo que se asigna a la variable de entrada de programa representa un valor de la señal de sensor que se genera con un sensor no diseñado a prueba de errores.

Esta medida tiene la ventaja de que se pueden realizar funciones de seguridad usando sensores de menor coste, no diseñados a prueba de errores. No es indispensable el uso de un costoso sensor diseñado a prueba de errores. Esto permite reducir el costo de realización de los mandos de seguridad.

Preferentemente, también el valor momentáneo de la variable de programa adicional seleccionada se produce mediante el uso de un sensor no diseñado a prueba de errores. Preferentemente, el sensor no diseñado a prueba de errores que es uno con el que se registra una magnitud física que se requiere en el contexto de las funciones estándar a ser realizadas por el mando de seguridad. Esto tiene la ventaja que además de los sensores requeridos para realizar las funciones estándar no es necesario emplear otros sensores. Por lo tanto, el mando de seguridad se

puede realizar de una manera favorable en cuanto al coste.

Según se ha explicado previamente, tanto la primera variable de programa no relevante para la seguridad seleccionada como también la variable de programa adicional seleccionada pueden ser respectivamente una variable de entrada de programa. Alternativamente, estas dos variables de programa también pueden ser respectivamente una variable intermedia de programa. El valor momentáneo de una variable intermedia de programa se determina, por ejemplo, dependiendo del valor momentáneo de una variable de entrada de programa. En ambas variables de programa también se puede tratar respectivamente de una variable de salida de programa. El valor momentáneo de una variable de salida de programa se determina, por ejemplo, dependiendo del valor momentáneo de una variable de entrada de programa o del valor momentáneo de una variable intermedia de programa, y representa el valor de una señal de salida de mando, con la que se controla un actor. La primera variable de programa no relevante para la seguridad seleccionada y la variable de programa adicional seleccionada no tienen que ser necesariamente del mismo tipo de variables de programa. También puede existir una de las combinaciones imaginables.

Se entiende que las características previamente descritas y las que aún serán explicadas más abajo se pueden usar no solamente en la combinación respectivamente señalada, sino también en otras combinaciones o de forma individual, sin salirse del marco de la presente invención.

Ventajosamente, la variable relevante para la seguridad seleccionada se puede usar como magnitud intermedia, dependiendo de la cual se determina, por ejemplo, una señal de salida de mando relevante para la seguridad.

En las explicaciones previas y en las subsiguientes, bajo el término "programa de aplicación" se entiende que un programa de aplicación comprende y por consiguiente representa tanto código fuente como también código de máquina. O formulado de otra manera: el programa de aplicación es representado por el código fuente y el código de máquina.

Cabe señalar además que los términos "no relevante para la seguridad" y "relevante para el proceso" en lo anterior y en los subsiguiente se emplean como sinónimos.

Ejemplos de realización de la invención se representan en los dibujos y se explicarán más detalladamente en la siguiente descripción. En los dibujos:

La Fig. 1 es una representación esquemática del nuevo dispositivo en conexión con un mando de seguridad, para el cual se debe crear un programa de aplicación;

la Fig. 2 es un diagrama de flujo simplificado para explicar el nuevo procedimiento;

la Fig. 3 es una representación simplificada de una primera superficie gráfica para la elaboración de un programa de aplicación; y

la Fig. 4 es una representación simplificada de una segunda superficie gráfica para la elaboración de un programa de aplicación.

En la Fig. 1, un dispositivo de acuerdo con la invención se designa en su totalidad con el número de referencias 10.

El dispositivo 10 comprende un ordenador convencional 12 con una unidad de visualización 14, en el cual se ejecuta un programa de ordenador 16. El programa de ordenador 16 permite la creación de un programa de aplicación para un mando de seguridad. En el lenguaje técnico, por lo tanto, muchas veces también recibe el nombre de "herramienta de programación". El ordenador 12 puede estar realizado como PC y la unidad de visualización 14 puede estar realizada como un monitor.

En la Fig. 1 se representa un circuito de seguridad designado en su totalidad con el número de referencia 18, el cual presenta un mando de seguridad 20 que está diseñado para controlar una instalación automatizada designada en su totalidad con el número de referencia 22. La instalación automatizada 22 comprende una pluralidad de actores 24 y una pluralidad de sensores 26. De manera ejemplar se representa un consumidor 28 comprendido en la instalación 22 y que puede ser, por ejemplo, un robot.

El mando de seguridad 20 está estructurado de forma redundante en dos canales, con la finalidad de alcanzar la seguridad requerida frente a los errores para controlar procesos críticos en cuanto a la seguridad. De manera representativa para la construcción en dos canales, en la Fig. 1 se representan dos procesadores separados entre sí, es decir, un primer procesador 30 y un segundo procesador 32. Los dos procesadores 30, 32 están comunicados a través de un interface de comunicación bidireccional 34 para poder controlarse mutuamente e intercambiar datos. Preferentemente, los dos canales del mando de seguridad 20 y los dos procesadores 30, 32 están construidos de forma diversa, es decir diferentes entre sí, para prevenir errores sistemáticos en la mayor medida posible.

Con el número de referencia 36 se designa una unidad de entrada/salida que está conectada con cada uno de los dos procesadores 30, 32. La unidad de entrada/salida 36 recibe una pluralidad de señales de entrada de mando 38 de la pluralidad de sensores 26 y las dirige en un formato de datos adaptado a cada uno de los dos procesadores 30, 32. Suponiendo que las señales de entrada de mando son señales análogas, los valores análogos de las señales de entrada de mando se transforman en valores digitales a través de una transformación A/D, las cuales se asignan entonces como valores momentáneos a variables de entrada de programa. Adicionalmente, la unidad de entrada/salida 36 dependiendo de los procesadores 30, 32 genera una pluralidad de señales de salida de mando 40 con las que se controla la pluralidad de actores 24. Suponiendo que las señales de salida de mando son señales análogas, los valores momentáneos de las variables de salida de programa disponibles como valores digitales se transforman para ello en valores análogos mediante una transformación D/A, los cuales representan entonces los valores de las señales de salida de mando.

Con el número de referencia 42 se designa una memoria de programa, en la que se almacena un programa de aplicación en forma de código de máquina. El programa de aplicación y por lo tanto el código de máquina se crean mediante el dispositivo 10. Si la memoria de programa 42 está configurada como tarjeta de chip, ello permite cambiar fácilmente el código de máquina y por consiguiente el programa de aplicación, incluso sin una conexión directa al ordenador 12. Alternativamente, la memoria de programa 42 también puede estar diseñada como una memoria integrada de manera fija en el mando de seguridad 20, tal como por ejemplo una memoria EEPROM.

El programa de ordenador 16 suministrar una superficie de usuario 44 en la unidad de visualización 14. La superficie de usuario 44 le permite a un programador o técnico la creación de un programa de aplicación. El programador crea el programa de aplicación introduciendo datos en el ordenador 12 a través de una unidad de entrada conectada. La unidad de entrada puede ser, por ejemplo, un teclado o un ratón. Dependiendo del lenguaje de programación que se use para la creación del programa de aplicación, se diferencian los conceptos de introducción. Si como lenguaje de programación se usan por ejemplo el lenguaje de programación Structured Text, entonces el programa de aplicación se crea mediante entradas textuales. Si por el contrario se usa el lenguaje de programación Function Block Diagram, entonces el programador crea el programa de aplicación selecciona módulos de programa predefinidos usando el ratón, los cuales se representan, por ejemplo, en la superficie de usuario 44 mediante símbolos gráficos.

Independientemente del lenguaje de programación que se use, de acuerdo con el nuevo procedimiento el programador debe realizar diversas entradas. Estas entradas comprenden la definición de un número de variables de programa relevantes para la seguridad 46, la definición de un número de variables de programa no relevantes para la seguridad 48, la selección de una variable de programa relevante para la seguridad 50 entre el número de variables de programa relevantes para la seguridad 46, la selección de una primera variable de programa no relevante para la seguridad 52 entre el número de variables de programa no relevantes para la seguridad 48, la definición de una condición de asignación 54, la definición de una asignación 56 y la creación de instrucciones de mando 58. La asignación 56 desde el punto de vista técnico de la programación corresponde a una instrucción, mediante la cual la primera variable de programa no relevante para la seguridad seleccionada 52 se asigna a la variable de programa relevante para la seguridad seleccionada 50. Durante la ejecución del programa de aplicación, los valores momentáneos de la variable de programa no relevante para la seguridad de esta manera se asignan a la variable de programa relevante para la seguridad. La asignación tiene el carácter de un mapeo.

El programa de ordenador 16 comprende un módulo de visualización 60. Con el módulo de visualización 60 se registran y se evalúan las entradas realizadas por el programador a través de la unidad de entrada. Por una parte, el módulo de visualización 60 genera un código fuente 62 que representa la respectiva entrada, siendo almacenado en una memoria de código fuente 64. Referido a un código fuente completo, que existe después de que el programador haya realizado sus entradas, en el caso del código fuente 62 se trata de un volumen de código fuente. Por otra parte, el módulo de visualización 60 produce la visualización o representación, respectivamente, de símbolos gráficos 66 en una forma de representación básica. Los símbolos gráficos 66 se representan las entradas realizadas por el programador y por consiguiente también el código fuente 62 generado por el módulo de visualización 60. Por lo tanto, es posible representar el programa de aplicación o el código fuente para el programa de aplicación, respectivamente, en la unidad de visualización 14.

El programa de ordenador 16 comprende además un módulo de reconocimiento 68 en el que se evalúa el código fuente 62. El código fuente generado en su totalidad por el módulo de visualización 60 contiene un código de transformación que comprende una primera porción de código y una segunda porción de código. La primera porción de código representa la asignación 56 y la segunda porción de código representa la condición de asignación 54. Si el módulo de reconocimiento 68 ahora reconoce la primera porción de código, entonces por una parte el módulo de visualización 60 recibe un mensaje de reconocimiento 70. Partiendo del mensaje de reconocimiento 70, el módulo de visualización 60 hace que por lo menos uno de los símbolos gráficos 66 se represente en una forma de representación modificada con respecto a la forma de representación básica. Por otra parte, el módulo de reconocimiento 68 genera un código de duplicación 72. El código de duplicación 72 se suministra a la memoria de código fuente 64 y por consiguiente se integra en el código fuente. A este respecto son imaginables diferentes formas de realización. El código de duplicación 72 puede ser una parte de programa autónoma, que existe de forma independiente de la primera parte de programa 74 y de la segunda parte de programa 78. El código de duplicación 72 también puede formar parte de la primera parte de programa 74 o ser parte de la segunda parte de programa 78.

Una vez que el programador haya realizado todas las entradas requeridas para su programa de aplicación, el código fuente estará disponible de forma completa en la memoria de código fuente 64. El código fuente comprende una primera parte de programa 74 en la que se procesan a prueba de errores las variables de programa relevantes para la seguridad. La primera parte de programa 74 a su vez comprende un módulo de programa 76, en el que se resume la primera porción de código y la segunda porción de código. Debido a la funcionalidad asociada con el módulo de programa 76, el módulo de programa 76 se puede denominar como módulo de transformación. Adicionalmente, el código fuente comprende una segunda parte de programa 78, en la que se procesan las variables de programa no relevantes para la seguridad, en donde para las variables de programa no relevantes para la seguridad dentro de la segunda parte de programa 78 no se requiere un procesamiento a prueba de errores. La primera parte de programa 74 comprende el código fuente que representa las instrucciones de seguridad que se requieren para las funciones de seguridad a ser cumplidas por el mando de seguridad 20. Las instrucciones de seguridad comprenden no sólo instrucciones de mando de seguridad, sino también instrucciones de transformación, en donde las instrucciones de transformación representan la asignación 56 y la condición de asignación 54. Las instrucciones de transformación se resumen en el módulo de programa 76. Las instrucciones de transformación resumidas en el módulo de programa 76 son independientes con respecto a las instrucciones de mando de seguridad propiamente dichas. La segunda parte de programa 78 comprende el código fuente que representa las instrucciones estándar que se requieren para las funciones estándar a ser cumplidas por el mando de seguridad 20.

El código fuente 80 completo almacenado en la memoria de código fuente 64 se traduce en código de máquina mediante un compilador 82. Preferentemente, en el mismo se asegura adicionalmente con una comprobación de redundancia cíclica CRC (Cyclic Redundancy Check).

El código de máquina traducido por el compilador 82 se almacena en la memoria de programa 42. Para un procesamiento a prueba de errores de variables de programa relevantes para la seguridad, en la memoria de programa 42 se almacena un primer código de máquina 84 y un segundo código de máquina 86. El primer código de máquina 84 está destinado al primer procesador 30 y el segundo código de máquina 86 está destinado al segundo procesador 32. El primer código de máquina 84 comprende un primer código de seguridad 88 y un código estándar 90. El primer código de seguridad 88 comprende, por una parte, aquellas instrucciones de seguridad que deben ser procesadas por el primer procesador 30 en el marco de las funciones de seguridad a ser cumplidas por el mando de seguridad 20. Por otra parte, el primer código de seguridad 88 comprende aquellas instrucciones de seguridad que deben ser procesadas por el procesador 30 en el marco de la transformación que está definida por la asignación 56 y la condición de asignación 54. En total, el primer código de seguridad 88 por lo tanto comprende instrucciones de mando de seguridad e instrucciones de transformación. El código estándar 90 comprende aquellas instrucciones de seguridad que deben ser procesadas por el procesador 30 en el marco de las funciones estándar a ser cumplidas por el mando de seguridad 20. El segundo código de máquina 86 comprende un segundo código de seguridad 92. De manera correspondiente a las descripciones relacionadas con el primer código de seguridad 88, el segundo código de seguridad 92 comprende aquellas instrucciones de seguridad, es decir, instrucciones de mando de seguridad e instrucciones de transformación, que deben ser procesadas por el segundo procesador 32. El programa de aplicación creado por el programador comprende el código fuente completo 80 y los dos códigos de máquina 84, 86.

Dependiendo del avance de procesamiento del programa de aplicación, en el primer procesador 30 por una parte se procesa una primera instrucción de seguridad actual 94 y por otra parte se procesa una instrucción estándar actual 96. De manera esencialmente simultánea, en el segundo procesador 32 se procesa una segunda instrucción de seguridad actual 98. Tanto en la primera instrucción de seguridad actual 94 como también en la segunda instrucción de seguridad actual 98 se puede tratar de una instrucción de mando de seguridad o de una instrucción de transformación.

En el marco del procesamiento de la instrucción estándar actual 96, en la que se trata de una instrucción de mando estándar y por lo tanto de una instrucción de mando no relevante para la seguridad, se intercambian primeros datos no relevantes para la seguridad 100 entre el procesador 30 y una unidad de entrada/salida 36. A este respecto, al primer procesador 30 se suministran datos mediante el uso de variables de entrada de programa, cuyos valores momentáneos representan señales de entrada de mando no relevantes para la seguridad 102 que son generadas por sensores no relevantes para la seguridad 104. En los sensores no relevantes para la seguridad 104 se trata de sensores que por ejemplo registran magnitudes de entrada requeridas para la regulación de un accionamiento. A este respecto se puede tratar, por ejemplo, de números de revoluciones, ángulos o velocidades. Los sensores no relevantes para la seguridad 104 no están diseñados a prueba de errores. A la unidad de entrada/salida 36 se suministran datos mediante el uso de variables de salida de programa, cuyos valores momentáneos representan señales de salida de mando no relevantes para la seguridad 106 que son suministrados a actores no relevantes para la seguridad 108 para controlar los mismos. Los actores no relevantes para la seguridad 108 pueden ser, por ejemplo, motores y cilindros de regulación. Los valores momentáneos de las variables de salida de programa no relevantes para la seguridad se determinan dependiendo de las variables de entrada de programa no relevantes para la seguridad de acuerdo con las instrucciones estándar. A este respecto puede ser necesario determinar magnitudes intermedias, cuyos valores momentáneos se asignan variables intermedias de programa. Los valores momentáneos de las variables intermedias de programa se suministran y se almacenan de forma intermedia en una memoria de trabajo 112 mediante segundos datos no relevantes para la seguridad 110.

Si en la primera instrucción de seguridad actual 94 se trata de una instrucción de mando de seguridad y por lo tanto de una instrucción de mando relevante para la seguridad, entonces en el marco de su procesamiento se intercambian primeros datos relevantes para la seguridad 114 entre el primer procesador 30 y la unidad de entrada/salida 36. A este respecto, al primer procesador 30 se suministran datos mediante el uso de variables de entrada de programa relevantes para la seguridad, cuyos datos momentáneos representan señales de entrada de mando relevantes para la seguridad 116 que son generadas por sensores relevantes para la seguridad 118. En los sensores relevantes para la seguridad 118 se trata, por ejemplo, de pulsadores de parada de emergencia, mandos a dos manos, puertas de seguridad, aparatos para vigilar el número de revoluciones u otros sensores para registrar parámetros relevantes para la seguridad. A la unidad de entrada/salida 36 se suministran datos mediante el uso de variables de salida de programa relevantes para la seguridad, cuyos valores momentáneos representan señales de salida de mando relevantes para la seguridad 120 que se suministran a actores relevantes para la seguridad 122 para controlar los mismos. En los actores relevantes para la seguridad 122 se trata, por ejemplo, de así llamados contactores o coyuntores, cuyos contactos de trabajo están dispuestos en la conexión entre un suministro de corriente 124 y el consumidor 28. A través de los actores relevantes para la seguridad 122, se puede desconectar el suministro de corriente 124 del consumidor 28, mediante lo cual es posible que al presentarse un correspondiente funcionamiento de error, por lo menos el consumidor 28 puede ser puesto en un estado seguro. Los valores momentáneos de las variables de salida de programa relevantes para la seguridad se determinan dependiendo de las variables de entrada de programa relevantes para la seguridad de acuerdo con las instrucciones de mando de seguridad. A este respecto puede ser necesario determinar magnitudes intermedias relevantes para la seguridad, cuyos valores momentáneos se suministran a variables intermedias de programa relevantes para la seguridad. Los valores momentáneos de las variables intermedias de programa relevantes para la seguridad se suministran y se almacenan de forma intermedia a la memoria de trabajo 112 mediante segundos datos relevantes para la seguridad 126.

Si en la segunda instrucción de seguridad actual 98 se trata de una instrucción de mando de seguridad y por lo tanto de una instrucción de mando relevante para la seguridad, entonces se procede de forma correspondiente a la primera instrucción de seguridad actual 94 procesada en el primer procesador 30. En lo referente a la segunda instrucción de seguridad actual 98, se usan de manera correspondiente terceros datos relevantes para la seguridad 128, que corresponden a los primeros datos relevantes para la seguridad 114, y cuartos datos relevantes para la seguridad 130, que corresponden a los segundos datos relevantes para la seguridad 126.

Las descripciones precedentes, de acuerdo con las cuales tanto el primer procesador 30 como también el segundo procesador 32 generan valores para las señales de salida de mando relevantes para la seguridad 120, no significa que los valores generados por estos dos procesadores al mismo tiempo se emitan como señales de salida de mando 120. Las descripciones precedentes sólo tienen la intención de reflejar la estructura redundante del mando de seguridad 20 en lo referente a las funciones de seguridad a ser cumplidas. Ambos procesadores 30, 32 están diseñados para determinar valores para las señales de salida de mando 120. Durante un funcionamiento libre de errores del mando de seguridad 20, sólo se emiten como señales de salida de mando 120 los valores determinados por un procesador, por ejemplo el primer procesador 30.

Al mando de seguridad 20 se pueden conectar otras unidades periféricas 132 a través de la unidad de entrada/salida 36 que se requieren en el marco de las funciones de seguridad y las funciones estándar a ser cumplidas por el mando de seguridad 20. Se puede tratar, por ejemplo, de un interruptor selector de modos de funcionamiento o de un pulsador de autorización. Pero también se puede tratar de una unidad de visualización.

A través de la unidad de entrada/salida 36, entre el mando de seguridad 20 y los sensores relevantes para la seguridad 118, los actores relevantes para la seguridad 122 I, en caso de ser necesario, también entre las otras unidades periféricas 132 se intercambian señales de prueba 134. A través de las señales de prueba 134, en el mando de seguridad 20 se puede determinar si las unidades y componentes conectados a la misma trabajan sin errores, lo cual es necesario, debido a que se tiene que garantizar un estado seguro de la instalación 22 controlada, tan pronto se presente un error de funcionamiento en uno de los aparatos conectados con el mando de seguridad 20.

Si en la primera instrucción de seguridad actual 94 y en la segunda instrucción de seguridad actual 98 se trata de una instrucción de transformación, entonces en los dos procesadores 30, 32 se procesa una de las etapas que se requieren para que un valor momentáneo se asigne a la primera variable de programa relevante para la seguridad seleccionada 52 dependiendo de la condición de asignación 54 de la variable de programa relevante para la seguridad seleccionada 50. Para ello, un valor momentáneo original 136 disponible en una etapa de tiempo definida de la primera variable de programa no relevante para la seguridad 52 se suministra una unidad de duplicación 138. Partiendo de la primera variable de programa no relevante para la seguridad seleccionada 52, en la unidad de duplicación 138 se genera una variable de programa no relevante para la seguridad duplicada. Para ello, el valor momentáneo original 136 se guarda en dos áreas de memoria separadas. Por motivos de claridad, dichas dos áreas de memoria no se representan. Las dos áreas de memoria se leen de manera independiente, de tal manera que el valor momentáneo original 136 se suministra a una primera unidad de prueba 140 y un valor momentáneo duplicado 142 se suministra a una segunda unidad de prueba 144. En la segunda unidad de prueba 144 por lo tanto se evalúa y por ende se procesa una variable de programa no relevante para la seguridad duplicada.

Las dos unidades de pruebas 140, 144 forman conjuntamente una unidad de plausibilidad 146. La unidad de duplicación 138 y la unidad de plausibilidad 146 pueden considerarse resumidas como una unidad de transformación. En este punto cabe mencionar que en el caso de la unidad de duplicación 138 en las unidades de pruebas 140, 144 y en la unidad de plausibilidad 146 se trata en todas ellas de unidades funcionales y no de unidades estructurales dentro de la memoria de trabajo 112.

En las dos unidades de pruebas 140, 144 se procesa de manera independiente de la condición de asignación 54. En la primera unidad de pruebas 140 para la primera variable de programa no relevante para la seguridad seleccionada 52 y en la segunda unidad de prueba 144 para la variable de programa no relevante para la seguridad duplicada. Por consiguiente, tanto para la primera variable de programa no relevante para la seguridad seleccionada 52 como también para la variable de programa no relevante para la seguridad duplicada se determina si el requisito de confiabilidad antes mencionado se ha cumplido respectivamente.

Para la condición de asignación 54 son posibles dos formas de realización. En una primera forma de realización se define una consulta de plausibilidad como condición de asignación, mediante la cual se comprueba si la primera variable de programa no relevante para la seguridad seleccionada 52 y una variable de programa adicional seleccionada son compatibles entre sí. De manera correspondiente se comprueba si la variable de programa no relevante para la seguridad duplicada y la variable de programa adicional seleccionada son compatibles entre sí. Si en la variable de programa adicional seleccionada se trata de una variable de programa no relevante para la seguridad, entonces a las dos unidades de pruebas 140, 144 se suministra un primer valor momentáneo 148 disponible en la etapa de tiempo definida. A este respecto, se trata de un valor momentáneo de una variable de programa no relevante para la seguridad adicional seleccionada. Si en la primera unidad de pruebas 140 se determina que la primera variable de programa no relevante para la seguridad seleccionada 52 y la variable de programa no relevante para la seguridad adicional seleccionada son compatibles entre sí, entonces el valor momentáneo original 136 se asigna a la variable de programa relevante para la seguridad seleccionada 50 y se suministra al primer procesador 30. Si en la segunda unidad de pruebas 144 se determina que la variable de programa no relevante para la seguridad duplicada y la variable de programa no relevante para la seguridad adicional seleccionada son compatibles entre sí, entonces el valor momentáneo duplicado 142 se asigna a una variable de programa relevante para la seguridad que corresponde a la variable de programa relevante para la seguridad seleccionada 50, que se ha creado en el segundo código de seguridad 92 y que se procesa en el segundo procesador 32, y se suministra al segundo procesador 32.

Si por el contrario la variable de programa adicional seleccionada es una variable de programa relevante para la seguridad, a la primera unidad de prueba 140 se suministra entonces por medio de quintos datos relevantes para la seguridad 150 un segundo valor momentáneo disponible en la etapa de tiempo definida. A este respecto, se trata de un valor momentáneo de una variable de programa relevante para la seguridad adicional seleccionada. De manera correspondiente, a la segunda unidad de pruebas 144 se suministra por medio de sextos datos relevantes para la seguridad 152 un tercer valor momentáneo disponible en la etapa de tiempo definida. A este respecto se trata de un valor momentáneo de una variable de programa relevante para la seguridad correspondiente, creada en el segundo código de seguridad 92. En la primera unidad de pruebas 140 se determina entonces si la primera variable de programa no relevante para la seguridad 52 y la variable de programa relevante para la seguridad adicional seleccionada son compatibles entre sí. En la segunda unidad de pruebas 144 se determina luego si la primera variable de programa no relevante para la seguridad seleccionada 52 y la correspondiente variable de programa relevante para la seguridad usada en el segundo código de seguridad 92 son compatibles entre sí. La asignación y emisión del valor momentáneo original 136 y del valor momentáneo duplicado 144 se realiza de manera correspondiente al caso en que la variable de programa adicional seleccionada es una variable de programa no relevante para la seguridad.

En una segunda forma de realización, como condición de asignación se define una consulta de plausibilidad, mediante la cual se comprueba si la primera variable de programa no relevante para la seguridad seleccionada 52 cumple con un criterio de comparación, en donde dicho criterio de comparación representa una propiedad característica de la primera variable de programa no relevante para la seguridad seleccionada 52. En este caso, a la primera unidad de pruebas 140 se suministran datos de comparación mediante los quintos datos relevantes para la seguridad 150, con los que se compara el valor momentáneo original 136. A la segunda unidad de pruebas 144 se suministran datos de comparación correspondientes mediante los sextos datos relevantes para la seguridad 152. Si en la primera unidad de pruebas 140 se determina que la primera variable de programa no relevante para la seguridad seleccionada 52 cumple con el criterio de comparación, entonces el valor momentáneo original 136 se asigna a la variable de programa relevante para la seguridad seleccionada 50 y se emite al primer procesador 30. Si en la segunda unidad de pruebas 144 se determina que la variable de programa no relevante para la seguridad duplicada cumple con el criterio de comparación, entonces el valor momentáneo duplicado 142 se asigna a una variable de programa relevante para la seguridad, guardada en el segundo código de seguridad y correspondiente a la variable de programa relevante para la seguridad seleccionada 50, y se emite al segundo procesador 32.

Si para la prueba de la condición de asignación 54 en las dos unidades de pruebas 140, 144 se deben evaluar varios valores momentáneos o cambios cronológicos en los valores momentáneos, entonces en las dos unidades de pruebas 140, 144 se almacenan de forma intermedia los valores momentáneos disponibles para varias etapas de

tiempo sucesivas.

En la Fig. 1 se describe una primera forma de realización, en la que tanto el primer código de seguridad 88 como también el segundo código de seguridad 92 contienen las instrucciones de transformación. Esto no debe tener un efecto limitador. También es imaginable una forma de realización en la que sólo el primer código de seguridad 88 contiene las instrucciones de transformación y el segundo código de seguridad 92 no. También es imaginable una forma de realización en la que uno o ambos códigos de seguridad solamente contienen aquellas instrucciones de transformación que representan la condición de asignación 54. En lo que respecta al código de duplicación 72, también son imaginables varias formas de realización. Es imaginable que sólo el primer código de seguridad 88 contenga instrucciones correspondientes. Sin embargo, también es posible que los dos códigos de seguridad 88, 92 contengan instrucciones correspondientes.

El diagrama de flujo representado en la Fig. 2 muestra en principio la forma de proceder para la creación del programa de aplicación de acuerdo con el nuevo procedimiento.

De acuerdo con la etapa 170, se define un número de variables de programa relevantes para la seguridad 46. En una siguiente etapa 172, se define un número de variables de programa no relevantes para la seguridad 48. En una etapa subsiguiente 174 se selecciona una variable de programa relevante para la seguridad 50 entre un número de variables de programa relevantes para la seguridad 46. En un siguiente paso 176 se selecciona una primera variable de programa no relevante para la seguridad 52 entre un número de variables de programa no relevantes para la seguridad 48. A la primera variable de programa no relevante para la seguridad seleccionada 52, se le asigna repetidamente un valor momentáneo durante la ejecución del programa de aplicación. En un siguiente paso 178 se define una condición de asignación 54 que se procesa durante la ejecución del programa de aplicación. Para el caso de que como condición de asignación se define una consulta de plausibilidad, en el paso 178 se selecciona además una variable de programa adicional. En un paso subsiguiente 180 se define una asignación 56 que asigna la primera variable de programa no relevante para la seguridad 52 a la variable de programa relevante para la seguridad 50. Durante la ejecución del programa de aplicación, el valor momentáneo de la primera variable de programa no relevante para la seguridad seleccionada 52 se asigna dependiendo de la condición de asignación 54 a la variable de programa relevante para la seguridad seleccionada 50. Si se debe transformar otra variable de programa no relevante para la seguridad en una variable de programa relevante para la seguridad, es decir, si se debe realizar una asignación adicional, lo cual se determina en un paso subsiguiente 182, se repiten nuevamente de las etapas 274 a 180. Si por el contrario ya no se va a realizar ninguna transformación adicional y por lo tanto tampoco una asignación adicional, entonces a continuación de la etapa 182 se ejecuta una etapa 184. En la etapa 184 se crean instrucciones de mando.

En la Fig. 3 se muestra una primera superficie de usuario gráfica 190 que representa un primer concepto de programación. En general, el programa de aplicación se crea usando un programa de ordenador 16 que se ejecuta en un ordenador 12. La primera superficie de usuario gráfica 190 le permite a un programador introducir entradas textuales requeridas para la creación de un programa de aplicación. El módulo de visualización 60 comprendido en el programa de ordenador 16 hace que durante la creación del programa de aplicación se visualice por lo menos el código fuente 62 o por lo menos por extractos una parte del código fuente completo 80 en la unidad de visualización 14 conectada al ordenador 12. En las explicaciones subsiguientes se describirá de manera simplificada un extracto de un código fuente.

En la Fig. 3 se representa el extracto del código fuente y por consiguiente del programa de aplicación, lo cual se indica mediante puntos suspensivos 192 y un número de rectángulos 194, en donde cada rectángulo representa una línea de programa del código fuente y por lo tanto del programa de aplicación. En general, la primera superficie de usuario gráfica 190 representada en la unidad de visualización 14 comprende una pluralidad de símbolos gráficos de código fuente 196 que representan el código fuente. A este respecto, los símbolos gráficos de código fuente están realizados en una forma de representación básica. La pluralidad de símbolos gráficos de código fuente 196 comprende un número de símbolos gráficos de código de transformación 198. El número de símbolos gráficos de código de transformación 198 representa un código de transformación contenido en el código fuente. El código de transformación a su vez comprende una primera porción de código 200, que representa la asignación 56, y una segunda porción de código 202, que representa las condiciones de asignación 54. A través del módulo de reconocimiento 68 se reconoce por lo menos una de las dos porciones de código 200, 202. Si existe una porción de código reconocida, el módulo de visualización 60 hace que por lo menos un símbolo gráfico de código de transformación 204, que está contenido en el número de símbolos gráficos de código de transformación 198, se represente en una forma de representación modificada con respecto a la forma de representación básica. En la Fig. 3, esto se indica mediante un rayado provisto en los dos rectángulos designados con el número de referencia 204.

La primera porción de código 200 representa instrucciones de transformación que a su vez representan la asignación 56. La segunda porción de código 202 representa instrucciones de transformación que a su vez representan la condición de asignación 54. Según se indica en la Fig. 3 mediante un bloque representado en línea punteada, las dos porciones de código 200, 202 están resumidas en el módulo de programa 76.

Mediante la representación de por lo menos un símbolo gráfico de transformación 204 en una forma de representación modificada con respecto a la forma de representación básica, dentro del código fuente y por consiguiente también dentro del programa de aplicación se crea un sitio de transformación detectado.

En la Fig. 4 se representa una segunda superficie de usuario gráfica 210, la cual representa un segundo concepto de programación. También en este segundo concepto de programación, el programa de aplicación se crea mediante el uso de un programa de ordenador 16 que se ejecuta en un ordenador 12. En este segundo concepto de programación, la creación del programa de aplicación sin embargo se realiza mediante la selección de un número de módulos de programa entre una pluralidad de módulos de programa predefinidos. Para poder realizar esta selección, el módulo de visualización 60 produce la visualización de una pluralidad de símbolos gráficos de módulos de programa 212 en la segunda superficie de usuario gráfica 210 y por lo tanto en la unidad de visualización 14. La pluralidad de símbolos gráficos de módulos de programa 212 comprende a su vez un primer número de símbolos gráficos de módulos de programa 214 que representan los módulos de programa predefinidos, así como un segundo número de símbolos gráficos de módulos de programa 216 que representan el número de módulos de programa seleccionados. La selección de módulos de programa predefinidos se realiza, por ejemplo, mediante el uso de un ratón por medio de una función de "arrastrar y soltar" (Drag & Drop) y se indica en la Fig. 4 a través de dos flechas 218. Por lo menos el primer número de símbolos gráficos de módulos de programa 214 están realizados en una forma de representación básica. El número de módulos de programa predefinidos comprende un módulo de programa 220 creado durante la creación del programa de aplicación, el cual representa la asignación 56 y la condición de asignación 54. El módulo de programa creado 220 se representa por medio de por lo menos un símbolo gráfico de módulo de programa modificado 222. El módulo de visualización 60 hace que el símbolo gráfico de módulo de programa 222 se visualice en una forma de representación modificada con respecto a la forma de representación básica. Esto se indica en la Fig. 4 a través de un rayado. El módulo de programa creado 220 corresponde al módulo de programa 76 previamente descrito.

La creación de un módulo de programa en el segundo concepto de programación se representa de la siguiente manera: En un campo de introducción específicamente provisto para ello, el programador fija la funcionalidad del módulo de programa a ser creado. Para ello, el programador introduce, por ejemplo en forma de entradas textuales, aquellas instrucciones de mando que deberán ser ejecutadas por el módulo de programa a ser creado. En este estadio ya se establece un símbolo gráfico de módulo de programa para el módulo de programa a ser creado. Si el módulo de reconocimiento 68 ahora reconoce una de las dos porciones de código 200, 202 dentro de las instrucciones de mando introducidas por el programador, el símbolo gráfico de módulo de programa ya establecido se modifica, mediante lo cual se visualiza en una forma de representación modificada con respecto a la forma de representación básica.

REIVINDICACIONES

1. Procedimiento para la creación de un programa de aplicación para un mando de seguridad (20) que está diseñado para controlar una instalación automatizada (22) con una pluralidad de sensores (26) y una pluralidad de actores (24), en donde el programa de aplicación comprende una primera parte de programa (74), en la que se procesan a prueba de errores variables de programa relevantes para la seguridad, y que comprende por lo menos una segunda parte de programa (78), en la que se procesan variables de programa no relevantes para la seguridad, en donde para las variables de programa no relevantes para la seguridad dentro de la segunda parte de programa (78) no se requiere un procesamiento a prueba de errores, **caracterizado por** las siguientes etapas:

- Definir un número de variables de programa relevantes para la seguridad (46),
- definir un número de variables de programa no relevantes para la seguridad (48),
- seleccionar una variable de programa relevante para la seguridad (50) entre el número de variables de programa relevantes para la seguridad (46),
- seleccionar una primera variable de programa no relevante para la seguridad (52) entre el número de variables de programa no relevantes para la seguridad (48), en donde a la primera variable de programa no relevante para la seguridad (52) se asigna repetidamente un valor momentáneo durante la ejecución del programa de aplicación,
- definir por lo menos una condición de asignación (54) que se procesa durante la ejecución del programa de aplicación,
- definir una asignación (56) que asigne la primera variable de programa no relevante para la seguridad seleccionada (52) a la variable de programa relevante para la seguridad seleccionada (50), en donde el valor momentáneo de la primera variable de programa no relevante para la seguridad seleccionada (52) durante la ejecución del programa de aplicación se asigna dependiendo de la condición de asignación (54) a la variable de programa relevante para la seguridad seleccionada (50).

2. Procedimiento de acuerdo con la reivindicación 1, **caracterizado por que** la condición de asignación (54) representa una consulta mediante la cual se determina si la primera variable de programa no relevante para la seguridad seleccionada (52) cumple con un requisito de confiabilidad exigido para variables de programa relevantes para la seguridad.

3. Procedimiento de acuerdo con cualquiera de las reivindicaciones precedentes, **caracterizado por que** como condición de asignación (54) se define una consulta de plausibilidad, mediante la cual se comprueba si la primera variable de programa no relevante para la seguridad seleccionada (52) y una variable de programa adicional seleccionada son compatibles entre sí.

4. Procedimiento de acuerdo con cualquiera de las reivindicaciones precedentes, **caracterizado por que** como condición de asignación (54) se define una consulta de plausibilidad, mediante la cual se comprueba si la primera variable de programa no relevante para la seguridad seleccionada (52) cumple con un criterio de comparación, en donde el criterio de comparación representa una propiedad característica de la primera variable de programa no relevante para la seguridad seleccionada (52)

5. Procedimiento de acuerdo con cualquiera de las reivindicaciones precedentes, **caracterizado por que** el programa de aplicación comprende una pluralidad de instrucciones de transformación, en donde las instrucciones de transformación representan la asignación (56) y la condición de asignación (54), estando por lo menos una parte de las instrucciones de transformación contenidas en la primera parte de programa (74).

6. Un procedimiento de acuerdo con cualquiera de las reivindicaciones precedentes, **caracterizado por que** una primera porción de código (200) que representa la asignación (56) y una segunda porción de código (202) que representa la condición de asignación (54) se resumen en un módulo de programa (76, 220).

7. Procedimiento de acuerdo con cualquiera de las reivindicaciones precedentes, **caracterizado por que** el programa de aplicación se crea mediante el uso de un programa de ordenador (16) que se ejecuta en un ordenador (12), en donde el programa de ordenador (16) comprende un módulo de visualización (60), en donde el módulo de visualización (60) hace que durante la creación del programa de aplicación se visualice un código fuente (62) para el programa de aplicación por medio de una pluralidad de símbolos gráficos de código fuente (196) que representan a dicho código fuente en una unidad de visualización (14) conectada al ordenador (12), en donde los símbolos gráficos de código fuente (196) están realizados en una forma de representación básica, en donde la pluralidad de símbolos gráficos de código fuente (196) comprende un número de símbolos gráficos de código fuente (198) que representan un código de transformación contenido en el código fuente (62), en donde el código de transformación comprende una primera porción de código (200) que representa la asignación (56) y una segunda porción de código (202) que representa la condición de asignación (54), en donde el programa de ordenador (16) comprende además un módulo de reconocimiento (68), mediante el cual se reconoce por lo menos una de las dos porciones de código (200, 202), en donde al existir una porción de código reconocida el módulo de visualización (60) hace que por lo menos un símbolo gráfico de código de transformación (204), el cual está contenido en el número de símbolos gráficos de código de transformación (198), se represente en una forma de representación modificada con respecto a la forma

de representación básica.

8. Procedimiento de acuerdo con la reivindicación 7, **caracterizado por que** con el módulo de reconocimiento (68) se reconoce la primera porción de código (200).

9. Procedimiento de acuerdo con cualquiera de las reivindicaciones precedentes, **caracterizado por que** el programa de aplicación se crea mediante el uso de un programa de ordenador (16) que se ejecuta en un ordenador (12), en donde la creación del programa de aplicación se realiza mediante la selección de un número de módulos de programa entre una pluralidad de módulos de programa predefinidos, en donde el programa de ordenador (16) comprende un módulo de visualización (60) que produce la visualización de una pluralidad de símbolos gráficos de módulo de programa (212) en una unidad de visualización (14) conectada al ordenador (12), en donde la pluralidad de símbolos gráficos de módulo de programa (212) comprende un primer número de símbolos gráficos de módulo de programa (214) que representan los módulos de programa predefinidos, y comprende un segundo número de símbolos gráficos de módulo de programa (216) que representan el número de módulos de programa seleccionados, en donde por lo menos el primer número de símbolos gráficos de módulo de programa (214) están realizados en una forma de representación básica, en donde el número de módulos de programa predefinidos comprende un módulo de programa (220) creado durante la creación del programa de aplicación, el cual representa la asignación (56) y la condición de asignación (54), en donde el módulo de programa creado (220) se representa a través de por lo menos un símbolo gráfico de módulo de programa modificado (222), en donde el módulo de visualización (60) hace que el símbolo gráfico de módulo de programa modificado (222) se visualice en una forma de representación modificada con respecto a la forma de representación básica.

10. Procedimiento de acuerdo con cualquiera de las reivindicaciones precedentes, **caracterizado por que** durante la ejecución del programa de aplicación, a partir de la primera variable de programa no relevante para la seguridad seleccionada (52) se genera una variable de programa no relevante para la seguridad duplicada.

11. Procedimiento de acuerdo con la reivindicación 10, **caracterizado por que** tanto para la primera variable de programa no relevante para la seguridad seleccionada (52) como también para la variable de programa no relevante para la seguridad duplicada se procesa la condición de asignación (54).

12. Procedimiento de acuerdo con cualquiera de las reivindicaciones precedentes, **caracterizado por que** la primera variable de programa no relevante para la seguridad seleccionada (52) es una variable de entrada de programa, en donde el valor momentáneo que se asigna a la variable de entrada de programa representa un valor de una señal de ascensor (102) que es generada con un sensor no diseñado a prueba de errores (104).

13. Dispositivo para crear un programa de aplicación para un mando de seguridad (20) que está diseñado para controlar una instalación automatizada (22) con una pluralidad de sensores (26) y una pluralidad de actores (24), en donde el programa de aplicación comprende una primera parte de programa (74), en la que se procesan a prueba de errores variables de programa relevantes para la seguridad, y que comprende por lo menos una segunda parte de programa (78), en la que se procesan variables de programa no relevantes para la seguridad, en donde para las variables de programa no relevantes para la seguridad dentro de la segunda parte de programa (78) no se requiere un procesamiento a prueba de errores, **caracterizado por** unidades (12, 14, 16) para definir un número de variables de programa relevantes para la seguridad (46) y para seleccionar una variable de programa relevante para la seguridad (50) entre un número de variables de programa relevantes para la seguridad (46), para definir un número de variables de programa no relevantes para la seguridad (48) y para seleccionar una primera variable de programa no relevante para la seguridad (52) entre el número de variables de programa no relevantes para la seguridad (48), en donde a la primera variable de programa no relevante para la seguridad (52) durante la ejecución del programa de aplicación se asigna repetidamente un valor momentáneo, para definir por lo menos una condición de asignación (54) que se procesa durante la ejecución del programa de aplicación, y para definir una asignación (56) que asigna la primera variable de programa no relevante para la seguridad seleccionada (52) a la variable de programa relevante para la seguridad seleccionada (50), en donde el valor momentáneo de la primera variable de programa no relevante para la seguridad seleccionada (52), durante la ejecución del programa de aplicación se asigna dependiendo de la condición de asignación (54) a la variable de programa relevante para la seguridad seleccionada (50).

14. Programa de ordenador con medios de código de programa para realizar un procedimiento de acuerdo con cualquiera de las reivindicaciones 1 a 12, cuando el programa de ordenador (16) se ejecuta en un ordenador (12).

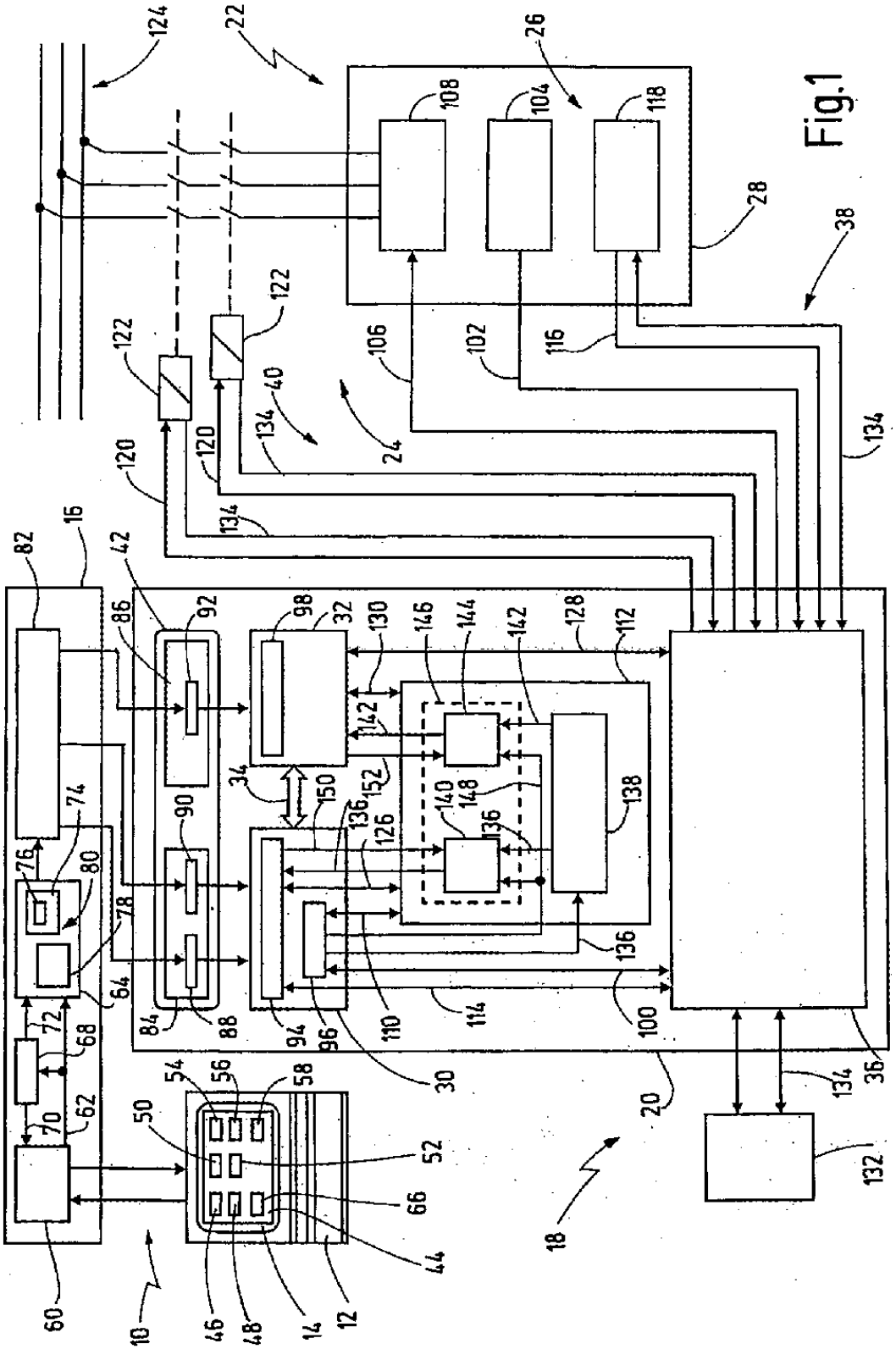

Fig.1

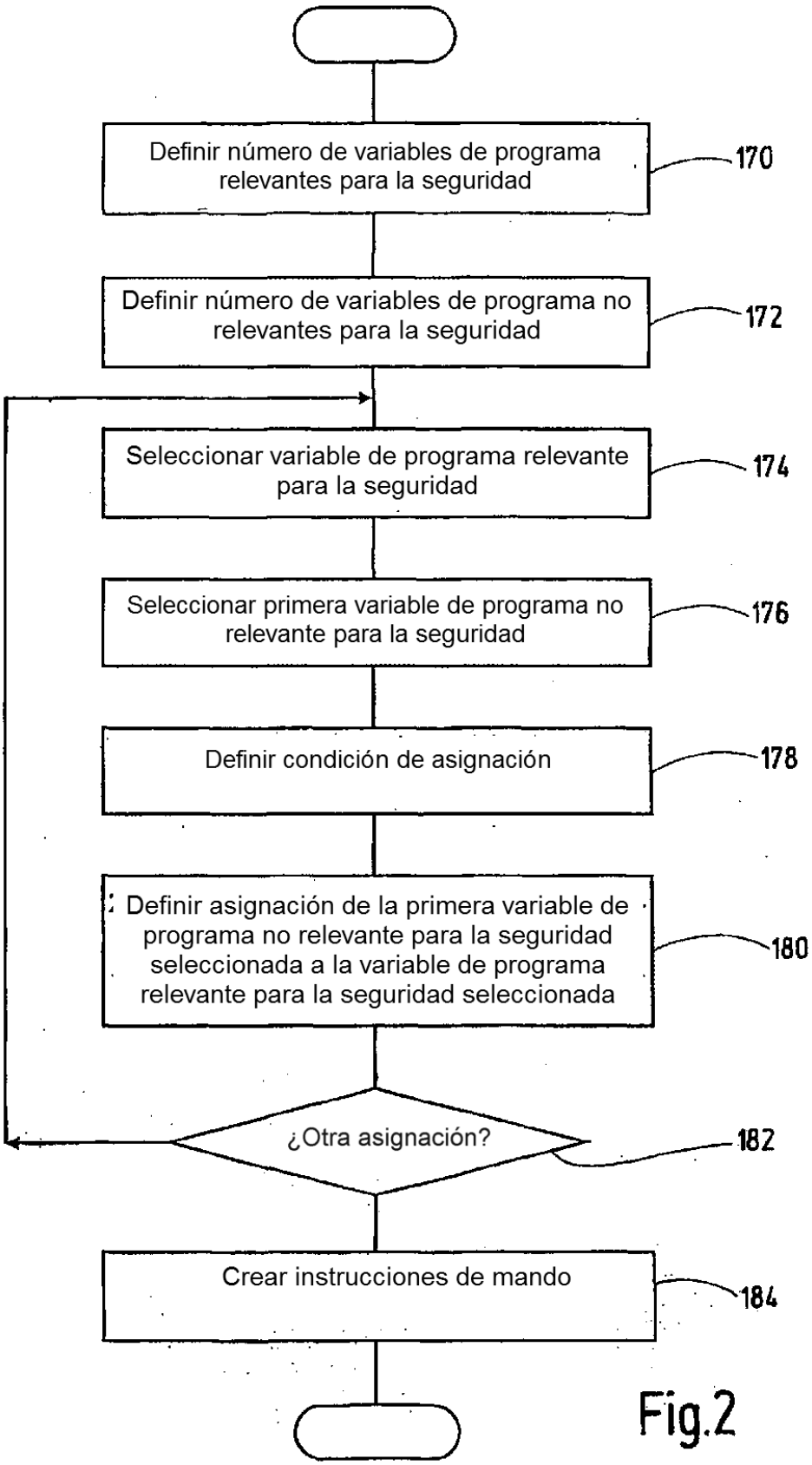
Definir número de variables de programa relevantes para la seguridad
170
Definir número de variables de programa no relevantes para la seguridad
172
Seleccionar variable de programa relevante para la seguridad
174
Seleccionar primera variable de programa no relevante para la seguridad
176
Definir condición de asignación
178
Definir asignación de la primera variable de programa no relevante para la seguridad seleccionada a la variable de programa relevante para la seguridad seleccionada
180
¿Otra asignación?
182
Crear instrucciones de mando
184

Fig.2

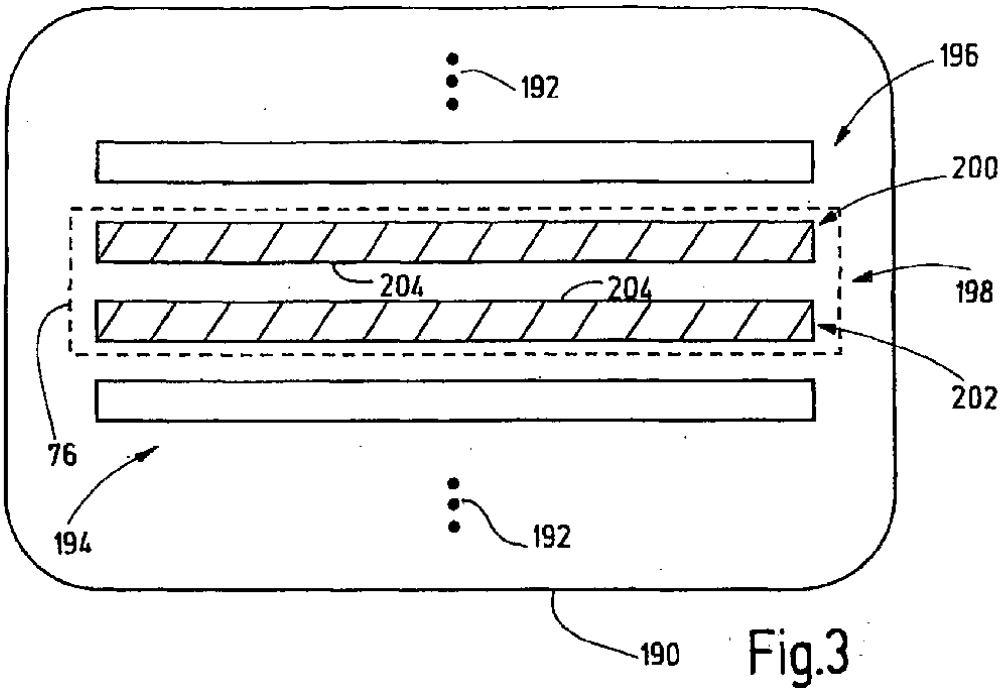
192
196
200
204
204
198
202
76
194
192
190

Fig.3

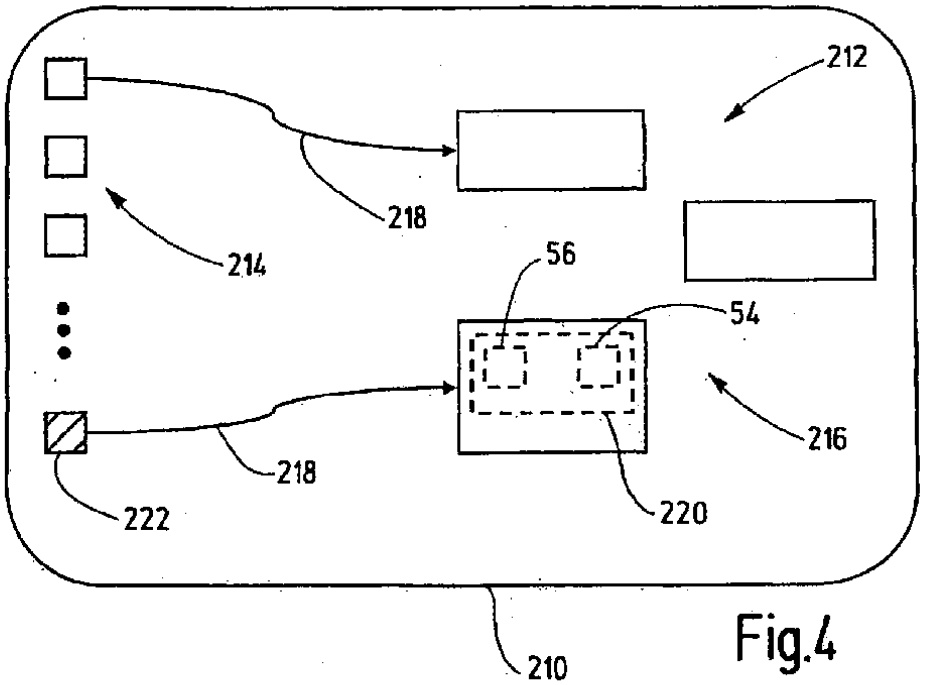
212
218
214
56
54
216
218
222
220
210

Fig.4