

19



OFICINA ESPAÑOLA DE  
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 974 305**

51 Int. Cl.:

**G06F 8/34** (2008.01)

**G05B 19/05** (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **24.04.2017** **E 17167844 (4)**

97 Fecha y número de publicación de la concesión europea: **03.01.2024** **EP 3396480**

54 Título: **Procedimiento para programar un control de seguridad**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:  
**26.06.2024**

73 Titular/es:

**WIELAND ELECTRIC GMBH (100.0%)**  
**Brennerstrasse 10-14**  
**96052 Bamberg, DE**

72 Inventor/es:

**LUDVIK, MARCO**

74 Agente/Representante:

**GONZÁLEZ PECES, Gustavo Adolfo**

**ES 2 974 305 T3**

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

DESCRIPCIÓN

Procedimiento para programar un control de seguridad

5 La invención se refiere a un procedimiento para programar un control de seguridad modular diseñado para encender y apagar de forma segura un consumidor eléctrico de acuerdo con una norma de seguridad, en el que por un procesador es representada en una pantalla una interfaz de programación con una programación lógica, y en una primera subárea de la interfaz de programación es representada una pluralidad de símbolos de bloque funcional, y tras la selección de un símbolo de bloque funcional por parte de un usuario dentro de la programación lógica, una instancia del símbolo de bloque funcional seleccionado es representada en una segunda subárea de la interfaz de programación, y en el que la segunda subárea está concebida de tal manera que las instancias ya seleccionadas de símbolos de bloque funcional son enlazadas lógicamente por el usuario con la ayuda de un diagrama lógico, y por el procesador, a base del diagrama lógico creado y de una configuración de módulos y una asignación de terminales, son creados o bien datos de configuración que están concebidos para ser cargados al control de seguridad modular y ser interpretados y usados por éste durante la ejecución de un firmware depositado allí, o bien es creado un firmware adaptado que está concebido para ser transferido al control de seguridad modular y ser ejecutado allí.

20 Los controles de seguridad se utilizan principalmente en el sector industrial para encender y apagar de forma segura consumidores eléctricos que entrañen riesgos potenciales para personas y materiales. Tales consumidores eléctricos son, por ejemplo, prensas, fresadoras, etc. Para este fin, la alimentación eléctrica del consumidor eléctrico es controlada a través de salidas de conmutación del control de seguridad, que están concebidas para cumplir las normas de seguridad pertinentes, por ejemplo, la norma EN ISO 13849. Para ello, los circuitos internos suelen estar concebidos típicamente de forma redundante, por ejemplo, aquellos para encender y apagar el consumidor eléctrico a través de dos relés conectados en serie, que son controlados por circuitos eléctricos separados.

25 Los controles de seguridad presentan habitualmente un número de terminales para conectar los contactos de trabajo de los transmisores y los actuadores. Los transmisores son, por ejemplo, interruptores de parada de emergencia o dispositivos de monitoreo de posición, como por ejemplo una puerta de protección, una barrera luminosa, una estera de conmutación, etc. Sirven para determinar un estado relevante para la seguridad. Los transmisores son, por ejemplo, relés, contactores, etc., con los que, como se ha explicado anteriormente, se conmuta el consumidor eléctrico. Los transmisores tienen habitualmente una posición "segura" que se determina en función de la aplicación, por ejemplo, barrera de luz no interrumpida, es decir, ninguna persona en la zona de peligro, y una posición "no segura", por ejemplo, posición de puerta de seguridad abierta, es decir, peligro. Típicamente, la alimentación eléctrica del consumidor eléctrico debe interrumpirse cuando el transmisor ya no está en la posición "segura". Pero además, puede haber también transmisores adicionales como, por ejemplo, dispositivos de monitoreo de marcha por inercia, que proporcionan información adicional para determinar una situación no segura u otra información de diagnóstico.

40 Los controles de seguridad a menudo también están estructurados de forma modular, de modo que pueden conectarse varios tipos de transmisores diferentes. Generalmente, un módulo de cabecera, que contiene los procesadores principales (redundantes) y realiza la mayor parte de las funciones de evaluación durante el tiempo de ejecución, se amplía con módulos de expansión adicionales. Éstos pueden realizar funciones especiales como, por ejemplo, proporcionar interfaces o terminales adicionales para transmisores y actuadores. Los módulos de expansión suelen estar conectados al módulo de cabecera a través de un bus de plano posterior.

45 En este tipo de controles de seguridad más grandes, ya sean de estructura modular o no modular, la programación solo con la ayuda de interruptores de ajuste manuales ya no es viable. Por lo tanto, aquí a menudo se recurre a la programación en una interfaz de programación de una pantalla de un ordenador digital, tal como también se conoce, por ejemplo, del ámbito de los controles lógicos programables (PLC). En una interfaz de programación de este tipo, al usuario habitualmente se muestran símbolos de bloque funcional que se seleccionan, por ejemplo, arrastrando y soltando, y se enlazan entre sí con la ayuda de un diagrama lógico. Este enlace lógico se utiliza a continuación en el software de programación para configurar o crear el programa utilizado en el momento de ejecución del control de seguridad. Los pasos de procedimiento descritos anteriormente y a continuación son iniciados de manera correspondiente por el procesador del ordenador.

55 Un procedimiento de este tipo es conocido por los expertos, por ejemplo, por el documento EP1362269B1. Los símbolos de bloque funcional se dividen allí en tres grupos, en concreto, en primer lugar, símbolos a los que está asignado respectivamente un tipo de transmisor; en segundo lugar, símbolos a los que está asignado respectivamente un tipo de actuador; y en tercer lugar, símbolos lógicos que en el marco de la lógica procesan magnitudes intermedias.

60 Además de la programación lógica, también son necesarias la selección y disposición de hardware adecuado. En el marco de la programación, debe ser definido o introducido por el usuario para cada actuador o transmisor utilizado a qué terminales del control de seguridad está conectado el respectivo actuador o transmisor. Además, en sistemas de control de seguridad modulares primero hay que especificar qué módulos se van a utilizar y cómo están dispuestos. A este respecto, precisamente en el ámbito de los controles de seguridad es sumamente importante que la entrada sea transparente para el usuario y lo menos propensa posible a errores, ya que las confusiones o asignaciones incorrectas pueden eventualmente tener consecuencias fatales.

En los documentos EP2012201B1 y EP2098926B1 se propone separar la configuración del hardware, es decir, la asignación de los transmisores y actuadores a terminales, de la programación lógica y llevarla a cabo en una interfaz separada en la que se muestra una imagen física del control de seguridad con sus terminales. En el documento EP2098926B1 también se divulga una asignación parcialmente automatizada en la que, en primer lugar, se seleccionan los elementos de hardware necesarios y se depositan en un área intermedia. Tras la selección completa de todos los elementos de hardware, se realiza una asignación automática de estos elementos de hardware a los terminales. Solo después tiene lugar la programación lógica, en la que solo los elementos de hardware previamente seleccionados pueden direccionarse dentro del diagrama lógico. Sin embargo, en este tipo de separación entre la vista del hardware y la programación lógica resulta problemática la falta de un enlace directo. Es que las dos vistas se muestran en páginas de pantalla diferentes, lo que puede representar una fuente adicional de error.

El documento WO02/067065A2 divulga una interfaz de programación para enlazar con la ayuda de un diagrama lógico símbolos de bloque funcional seleccionados.

Por lo tanto, el objetivo de la invención es indicar un procedimiento para programar un control de seguridad modular del tipo mencionado al principio, que haga posible la configuración del lado del hardware de la forma más segura posible.

Este objetivo se consigue según la invención porque a cada uno de los símbolos de bloque funcional está asignado un registro de datos que especifica un requisito de recursos de hardware, comprendiendo los recursos de hardware funcionalidades del control de seguridad modular y/o un número de terminales, y cuando es seleccionado el símbolo de bloque funcional dentro de la programación lógica sobre la base del requisito de recursos de hardware especificado en el registro de datos asignado a este bloque funcional, se comprueba si una configuración de módulos actual del control de seguridad modular puede proporcionar este requisito de recursos de hardware adicionalmente a un requisito de recursos de hardware actual, y si la configuración de módulos actual no puede proporcionar este requisito de recursos de hardware, se cambia la configuración de módulos, en cuyo caso, cuando se cambia la configuración de módulos, a base de los requisitos de recursos de hardware especificados en los respectivos registros de datos, asignados a los símbolos de bloque funcional de todas las instancias ya seleccionadas, de símbolos de bloque funcional en la segunda subárea se determina un requisito total de recursos de hardware, siendo cambiada la configuración de módulos de tal manera que la configuración de módulos cambiada pueda proporcionar el requisito total de recursos determinando de entre todas las combinaciones posibles de módulos aquellas combinaciones que pueden proporcionar el requisito total de recursos de hardware.

La invención parte de la idea de que una programación especialmente segura y que evite errores de la asignación de terminales y el enlace lógico sería posible en particular si la configuración del hardware no fuera realizada por el programador, sino más bien de forma automatizada. Por consiguiente, a diferencia de desarrollos más recientes, ya no se debe prever una interfaz separada para la composición previa de la configuración de hardware. En cambio, la configuración de hardware debe producirse de forma automatizada y, en particular, también en tiempo real durante la creación de la programación lógica. Para ello, a cada uno de los símbolos de bloque funcional que pueden ser seleccionados en el software de programación se le asigna un registro de datos que especifica un requisito de recursos de hardware para este símbolo de bloque funcional. Entonces, cada vez que el usuario selecciona un nuevo símbolo de bloque funcional, es decir, cada vez que se selecciona tal símbolo, se comprueba si la configuración de módulos depositada actualmente puede proporcionar el requisito de recursos de hardware adicionalmente al requisito de recursos de hardware actual. Esto resulta, por un lado, de los recursos de hardware proporcionados por la configuración de módulos actual, que asimismo están depositados en el software por cada módulo, y, por otro lado, de los recursos de hardware ya ocupados por símbolos de bloque funcional previamente seleccionados. Estos últimos, al igual que el requisito de recursos de hardware del símbolo de bloque funcional seleccionado actualmente, pueden determinarse a base de los registros de datos depositados. Si la configuración de módulos actual aún puede proporcionar suficientes recursos de hardware libres para el símbolo de bloque funcional seleccionado actualmente, adicionalmente a los recursos de hardware ya ocupados, se mantiene la configuración de módulos actual. Solo si este no es el caso se modifica la configuración de módulos. De esta manera, la configuración del hardware se crea de forma totalmente automática durante la programación lógica, sin necesidad de intervención del usuario, y además se actualiza continuamente en tiempo real.

Los símbolos de bloque funcional también incluyen bloques de función lógicos, es decir, símbolos de bloque funcional que no se refieren directamente a transmisores o actuadores específicos y sus señales de entrada o salida, sino que procesan señales de entrada o crean señales de salida dentro de la lógica, o que solo generan magnitudes intermedias en la secuencia lógica. Es que también este tipo de funciones puramente lógicas pueden, dado el caso, plantear exigencias al hardware que se adapta entonces en tiempo real mediante las medidas descritas anteriormente.

Los recursos de hardware depositados en los registros de datos para cada símbolo de bloque funcional incluyen funcionalidades del control de seguridad modular. Funcionalidades en relación con los recursos de hardware significa que el hardware debe tener determinadas propiedades para cumplir estas funcionalidades. Por ejemplo, se podría usar un bloque funcional lógico que como señal de entrada requiera una señal de un transmisor incremental. En este caso, debe utilizarse un módulo de cabecera que pueda procesar este tipo de señales y proporcione también las

entradas correspondientes. La funcionalidad "transmisor incremental" es, por tanto, un recurso de hardware requerido por el bloque funcional lógico en cuestión.

Los recursos de hardware incluyen además un número de terminales. El hardware usado debe proporcionar siempre suficientes terminales de entrada o salida, especialmente si dentro de la programación lógica se seleccionan símbolos de bloque funcional que se asignan directamente a transmisores o actuadores específicos o a sus señales. De este modo, los recursos de hardware especifican de manera ventajosa para cada símbolo de bloque funcional cuántos terminales se requieren para el respectivo símbolo, y preferiblemente además cuántos terminales de qué tipo (si el hardware no permite que los terminales se utilicen arbitrariamente).

La configuración de módulos incluye de manera ventajosa, por una parte, el tipo y, por otra, el número de módulos del sistema control de seguridad modular. Por ejemplo, son concebibles como tipos módulos de cabecera con diferentes funcionalidades, así como los llamados módulos de pasarela como interfaces, por ejemplo, a sistemas de bus de campo, o módulos de expansión de entrada o salida que proporcionan terminales de entrada o salida adicionales. De manera ventajosa, la configuración de módulos especifica exactamente cuántos módulos de qué tipo están previstos. También es especialmente ventajoso especificar su disposición exacta en una fila de módulos conectados entre sí a través de un bus.

Además de la configuración de módulos, también se almacena de manera ventajosa una asignación de los terminales de los módulos del control de seguridad modular. Esto significa que no solo se especifica qué módulos se usan, sino también cómo se asignan los terminales individuales de estos módulos a los transmisores y actuadores. Esta asignación también se adapta convenientemente en tiempo real. Para ello, tras la selección de un símbolo de bloque funcional, en cuyo registro de datos asignado está depositado un número de terminales como requisito de recursos de hardware, se modifica una asignación actual de los terminales de los módulos del control de seguridad modular. De esta manera, para el programador no solo resulta de forma totalmente automática el hardware que ha de usarse, sino también su cableado concreto.

Si, debido a la falta de recursos de hardware en la configuración de módulos actual, se cambia la configuración de módulos, en una realización ventajosa del procedimiento, sobre la base de los requisitos de recursos de hardware especificados en los respectivos registros de datos asignados a los símbolos de bloque funcional de todas las instancias ya seleccionadas de símbolos de bloque funcional en la segunda subárea de la interfaz de programación se determina un requisito total de recursos de hardware, y la configuración de módulos se modifica entonces de tal manera que la configuración de módulos cambiada pueda proporcionar el requisito total de recursos. Por lo tanto, la nueva configuración de módulos cambiada se ha seleccionado de manera que proporcione los recursos completos requeridos para todos los símbolos de bloque funcional.

Generalmente, entrarán en consideración varias configuraciones de módulos posibles. Preferiblemente, la configuración de módulos cambiada es, por tanto, la configuración de módulos que proporciona la menor cantidad posible de recursos de hardware. En otras palabras: Al determinar la nueva configuración de módulos cambiada, se produce una optimización automática de modo que solo se seleccionan y añaden los módulos absolutamente necesarios y no se proporciona un número innecesariamente grande de recursos de hardware no asignados entonces. Por lo tanto, de todas las configuraciones de módulos posibles, se forma un mínimo a través de los recursos de hardware necesarios, por así decirlo, y se selecciona y usa la configuración de módulos correspondiente a este mínimo. De este modo se minimiza el consumo de recursos y energía del control de seguridad durante el funcionamiento.

De manera ventajosa, además, las condiciones de configuración de módulos también están depositadas sobre la base de las limitaciones de hardware, por lo que, si la configuración de módulos cambiada no cumple las condiciones de configuración de módulos, se cancela la representación del símbolo del bloque funcional en la segunda subárea de la interfaz de programación. De esta manera, se evita que el usuario diseñe lógicas en la programación lógica que no son reproducidas por el hardware disponible. Así, por ejemplo, algunos símbolos de módulos lógicos solo pueden usarse una vez, ya que el hardware disponible solo puede proporcionar la funcionalidad correspondiente como máximo una vez. Esto se deposita como una condición de configuración de módulos correspondiente en el software. El proceso de selección llevado a cabo por el usuario se cancela entonces, es decir, la selección se deshace inmediatamente o ni siquiera se permite en la interfaz de programación. Al seleccionar mediante arrastrar y soltar, por ejemplo, el anclaje en el diagrama lógico podría ser imposible, dado el caso, con un mensaje al usuario. Además de las condiciones depositadas debido a limitaciones de hardware, de manera ventajosa, también pueden ser ajustadas de manera análoga las condiciones depositadas manualmente por el usuario.

Dado que el procedimiento adapta en tiempo real el hardware utilizado a los símbolos de bloque funcional seleccionados, de manera ventajosa esto también se produce a la inversa cuando se elimina un símbolo de bloque funcional. Para ello, de manera ventajosa, tras el borrado de un símbolo de bloque funcional de la segunda subárea por el usuario, se habilita en la configuración de módulos actual el requisito de recursos de hardware especificado en el registro de datos asignado a este bloque funcional. Si al mismo tiempo se especifica la asignación de terminales (véase más arriba), también se vuelve a modificar de manera ventajosa la asignación de los terminales.

Tras la habilitación descrita, de manera ventajosa, a su vez, a base de los requisitos de recursos de hardware, especificados en los respectivos registros de datos asignados a los símbolos de bloque funcional, de todas las instancias ya seleccionadas de símbolos de bloque funcional en la segunda subárea se determina un requisito total de recursos de hardware, y se comprueba además si otra configuración de módulos con una cantidad menor de recursos de hardware puede proporcionar el requisito total de recursos de hardware. Si sobre la base de la comprobación puede determinarse otra configuración de módulos de este tipo, la configuración de módulos actual se cambia a la otra configuración de módulos. En otras palabras: La optimización de la configuración de módulos descrita anteriormente tiene lugar de forma ventajosa también en caso de eliminarse un símbolo de bloque funcional, de modo que los módulos que ya no son necesarios se eliminan inmediatamente.

Un producto de programa de ordenador que puede cargarse directamente a la memoria interna de un ordenador digital con una pantalla comprende de manera ventajosa secciones de código de software que, cuando se ejecutan en el ordenador, capacitan al ordenador para ejecutar uno de los procedimientos descritos anteriormente, visualizándose la interfaz de programación en la pantalla del ordenador. Un producto de programa de ordenador de este tipo es un software de programación que por medio del hardware del ordenador, como una pantalla, e interfaces como, por ejemplo, un teclado y un ratón o una pantalla táctil, hace posible la creación de una programación por medio de la representación gráfica descrita de terminales del control de seguridad y pone a disposición las herramientas correspondientes para crear enlaces lógicos.

De manera ventajosa, un producto de programa de ordenador de este tipo se carga a la memoria interna de un ordenador digital. Dicho ordenador puede ser un ordenador personal estándar, pero con una interfaz adecuada también es concebible adaptar dicho software para tabletas o incluso teléfonos inteligentes. También otro tipo de sistemas propios pueden estar equipados con el software correspondiente. Todos estos sistemas se reúnen bajo el concepto de ordenador digital.

Un producto de programa de ordenador que puede cargarse directamente a la memoria interna de un control de seguridad, comprende de manera ventajosa secciones de código de software creadas por medio del procedimiento descrito. En este caso, la programación creada por medio del software para el ordenador digital se carga a la memoria del control de seguridad, de manera que puede ejecutarse de forma independiente y autónoma.

Un control de seguridad, concebido para encender y apagar un consumidor eléctrico de forma segura de acuerdo con una normativa de seguridad, está programado de manera ventajosa con el procedimiento descrito.

Las ventajas logradas con la invención consisten en particular en que, mediante una configuración de hardware en tiempo real durante el enlace lógico, de forma totalmente automática los módulos requeridos son configurados en cuanto a su tipo y número y son asignados sus terminales. De esta manera, el programador puede realizar directamente la programación lógica sin tener que ocuparse de la implementación del hardware. Cualquier limitación de hardware se le indican en la programación lógica. Con el procedimiento en tiempo real propuesto se elimina por completo la selección manual de módulos y la asignación de terminales propensas a errores, y el software las realiza de forma totalmente automática. El programador siempre dispone de información inmediata sobre cómo afecta un cambio lógico a la configuración de módulos actual y a la asignación de terminales. Esto último se le puede indicar, por ejemplo, en un subárea separada o en una pestaña separada de la interfaz de programación. De este modo se evitan errores al programar el control de seguridad modular y se aumenta la seguridad del sistema programado.

Ejemplos de realización de la invención se explican con más detalle con la ayuda de dibujos. En estos, muestran:

- La figura 1 una representación esquemática de un control de seguridad modular,
- la figura 2 una representación esquemática del control de seguridad modular en un modo de funcionamiento con un interruptor redundante,
- la figura 3 una representación esquemática del control de seguridad con un ordenador digital equipado para la programación,
- la figura 4 una representación esquemática de una interfaz de programación

Las piezas que son idénticas están designados por los mismos signos de referencia en todos los dibujos.

La figura 1 muestra esquemáticamente un control de seguridad 1 modular que comprende un módulo de cabecera 2, un módulo de expansión de entrada 4 y un módulo de expansión de salida 6, que están conectados entre sí en el lado de los datos a través de un bus de plano posterior 8. En el módulo de cabecera 2 se encuentra una unidad de control central 10 que procesa las señales de entrada y genera señales de salida a partir de ellas. El control de seguridad 1 está concebido de acuerdo con la norma EN ISO 13849, de modo que la unidad de control 10 comprende dos microcontroladores redundantes, que no se muestran en detalle.

El control de seguridad 1 presenta una multiplicidad de terminales 12 - 38, de los cuales seis terminales 12 - 22 están dispuestos en el módulo de cabecera 2, cuatro terminales 24 - 30 en el módulo de expansión de entrada 4 y cuatro terminales 32 - 38 en el módulo de expansión de salida 6. La unidad de control 10 puede controlar o invocar cada uno de los terminales 12 - 38 a través del bus de plano posterior 8 y, dado el caso, otras unidades procesadoras en los

módulos de expansión 4, 6.

Los terminales 32 - 38 del módulo de expansión de salida 6 pueden conectarse por pares a la fuente de alimentación de un actuador no mostrado en detalle, que conmuta un consumidor eléctrico que tiene un potencial de peligro y, por tanto, debe apagarse de forma segura. Para ello, en la conexión entre los terminales 32, 34 o 36, 38 están conectados en serie respectivamente dos relés 40, 42 o 44, 46. Para mantener el flujo de corriente, ambos relés 40, 42 o 44, 46 deben estar en estado cerrado, es decir que son redundantes. Los relés 40, 42 y 44, 46 son controlados independientemente por los microcontroladores redundantes de la unidad de control 10.

Para interrumpir o al menos influir en el funcionamiento del consumidor eléctrico en caso de peligro, pueden conectarse al control de seguridad 1 diversos transmisores, mostrados a modo de ejemplo en la siguiente figura 2 pero no en la figura 1. Pueden ser, por ejemplo, interruptores de parada de emergencia, barreras luminosas, interruptores de posición de puertas, etc. Para conectar estos transmisores, el control de seguridad 1 presenta, además de los terminales 24 - 30 del módulo de expansión de entrada 4, también terminales 12 - 22 que pueden alimentar eléctricamente los transmisores. En algunos de estos terminales 12 - 22, en el ejemplo de realización está siempre presente una pequeña tensión continua positiva, pero esto no es obligatorio, por ejemplo, en las llamadas salidas de colector abierto o triestado. Alternativamente, en otros de estos terminales 12 - 22 también pueden estar previstas señales de prueba con impulsos de prueba pulsados cíclicos para pruebas de línea y cortocircuito, que ajustan respectivamente durante un breve período 0 voltios por encima de la tensión de base obtenida de otro modo. Alternativamente, los impulsos de prueba también pueden emitirse en respuesta a un disparador, como una solicitud de prueba manual o automática. Todos los terminales 12 - 38 pueden estar concebidos a discreción en cuanto al hardware, por ejemplo, como terminales de tornillo o de muelle.

La figura 2 muestra un ejemplo de aplicación del control de seguridad 1 en estado bueno. Por razones de claridad, no se muestran todas las partes del control de seguridad 1 y solo se muestra un fragmento del control de seguridad.

La figura 2 muestra un interruptor de parada de emergencia 48 que comprende dos canales redundantes con dos interruptores 50, 52 conectados mecánicamente. Los interruptores 50, 52 son equivalentes, es decir, ambos están cerrados en el estado bueno y abiertos cuando se solicita la función de seguridad. Aquí, el interruptor de parada de emergencia 48 está conectado al control de seguridad 1 de tal manera que el terminal 16 está conectado al terminal 26 a través de uno de los interruptores 50 y el terminal 14 está conectado al terminal 24 a través del otro de los interruptores 52. Esto da como resultado la combinación de señales positivo, positivo, en los terminales 24, 26 cuando está habilitada, y la combinación de señales cero, cero cuando se solicita la función de seguridad. Otras combinaciones de señales indican un error. En estos dos últimos casos, se inicia una interrupción de la alimentación eléctrica del circuito eléctrico, enlazado al interruptor de parada de emergencia 48, del módulo de expansión de salida 6 mediante la programación lógica descrita a continuación, por ejemplo entre los terminales 36 y 38.

El ejemplo de las figura 1 y 2 solo sirve para ilustrar el modo de funcionamiento del control de seguridad 1 y, por tanto, está muy simplificado. En realidad, el sistema será mucho más complejo, es decir, habrá muchos más módulos y módulos de otros tipos (por ejemplo, los llamados módulos pasarela para interfaces con buses de campo), y habrá un gran número de otros terminales en diversos módulos que pueden conmutar una multiplicidad de consumidores eléctricos y a los que se puede conectar una multiplicidad de transmisores y actuadores de diversos tipos. Además, los terminales no tienen que cumplir necesariamente una sola función, sino que, dado el caso, también pueden usarse de forma variable como terminales de entrada o de salida.

La asignación de terminales y, por tanto, en última instancia, las combinaciones de señales que cabe esperar en los terminales, así como el enlace lógico entre las señales de transmisor y de actuador, deben estar depositados en la unidad de control 10 del control de seguridad 1, de modo que con la señalización correcta inicie el apagado de los consumidores eléctricos correctos. Esto se hace transfiriendo los parámetros correspondientes, es decir, la configuración correspondiente, a un programa de ordenador depositado en la unidad de control 10 (el firmware).

La creación de esta configuración se hace en un ordenador digital 54 mostrado en la figura 3. El ordenador 54 presenta aparatos de entrada como un teclado 56 y un ratón 58, así como una pantalla 60 como aparato de salida. Se conecta al control de seguridad 1 a través de una interfaz serie o paralela o, por ejemplo, a través de Ethernet. En el ordenador 54, un usuario crea y comprueba la configuración en una interfaz gráfica de programación utilizando la combinación prevista de transmisor y actuador y el enlace lógico deseado y, a continuación, se transfiere a la unidad de control 10 del control de seguridad 1. En una forma de realización alternativa no mostrada, el ordenador digital 54 también puede estar realizado como una tableta con pantalla táctil. Cuando en el siguiente ejemplo se habla de un clic o movimiento del ratón, en el caso de una tableta se realizaría pulsando y arrastrando el dedo de manera correspondiente.

Tal interfaz de programación 62 mostrada en una pantalla 60 se muestra en la figura 4. La interfaz de programación 62 tiene una primera subárea 64 en la que se muestran los símbolos de bloques de función 66, 68, 70. Los símbolos de bloque funcional 66, 68, 70 se agrupan en tres grupos. El primer grupo comprende símbolos de bloque funcional 66, a los que están asignados un transmisor, por ejemplo, un interruptor de parada de emergencia monocanal, un interruptor de parada de emergencia bicanal, etc. Estos símbolos de bloque funcional 66 simbolizan una señal de entrada de tal, y en la programación lógica, la redundancia implementada en el lado del hardware (véase más arriba)

ya no es visible para el usuario, ya que el monitoreo de errores la realiza el propio firmware del control de seguridad 1, incluso sin programación separada. Durante la programación, el usuario solo debe definir cómo funciona el accionamiento de dicho transmisor. Por ejemplo, en el caso del interruptor de parada de emergencia de dos canales que se muestra en la figura 2, aunque éste comprende en realidad dos canales redundantes, solo se mostraría una única señal de salida en la programación lógica de la interfaz de programación 62, que se enlaza posteriormente en la lógica.

En consecuencia, el segundo grupo de símbolos de bloque funcional 68 comprende símbolos que simbolizan señales de salida a actuadores. Lo anterior se aplica aquí de forma análoga en este caso. Por último, el tercer grupo comprende símbolos de bloque funcional 70, que son bloques lógicos. Pueden ser combinaciones lógicas simples como "AND" u "OR", evaluadores más complejos como "flip-flops" (biestables), generadores de reloj o símbolos de bloque funcional 70 específicos de la aplicación para retardos de apagado, prensas, etc. En ejemplos de realización no mostradas, los símbolos de bloque funcional 66, 68, 70 también pueden subagruparse según otros criterios relacionados con el contenido.

El usuario selecciona, como parte de la programación lógica, un símbolo de bloque funcional 66, 68, 70. La selección se realiza moviendo el puntero del ratón, no representado en detalle, hasta el símbolo de bloque funcional 66, 68, 70 deseado y simplemente haciendo clic en el símbolo de bloque funcional 66, 68, 70 deseado. La selección mediante clic del ratón provoca la creación de una instancia 72 - 84 del símbolo de bloque funcional 66, 68, 70, que se adjunta al puntero del ratón.

La instancia 72 - 84 se genera porque pueden requerirse varios símbolos de bloque funcional 66, 68, 70 del mismo tipo y, por tanto, dado el caso, debe seleccionarse varias veces seguidas el mismo símbolo de bloque funcional 66, 68, 70. La instancia 72 - 84 puede ser idéntica en su representación al símbolo de bloque funcional 66, 68, 70, pero no tiene por qué serlo. En el ejemplo de realización, las instancias 72 - 84 son significativamente más anchas y planas que el símbolo de bloque funcional 66, 68, 70 para, dado el caso, poder mostrar texto en las mismas. Si el símbolo de bloque funcional 66, 68, 70 contiene un pictograma caracterizador, en formas de realización, este pictograma también puede representarse en la instancia 72 - 84, dado el caso, de forma ampliada o reducida de tamaño.

Para especificar el enlace lógico sirve un diagrama lógico creado por el programador con la ayuda de las instancias 72 - 84 y representado en la segunda subárea 86 de la interfaz de programación 62. En el diagrama lógico, la programación se realiza con la ayuda de símbolos gráficos que se vinculan lógicamente por trazado de líneas. Las señales de entrada de los transmisores, representadas por las instancias 72, 74, 76 de los símbolos de bloque funcional 66 son procesadas por bloques lógicos en las instancias 78, 80 y procesadas formando señales de salida para actuadores, representados por la instancia 84. Las instancias 72 - 84 se depositan tras seleccionar los símbolos de bloque funcional 66, 68, 70 (véase más arriba) mediante un sencillo procedimiento de arrastrar y soltar. Mediante la simple conexión por líneas, las señales pueden ser transferidas de la manera deseada. Mediante una hábil selección y combinación de los símbolos de bloque funcional 66, 68, 70 y la correspondiente conexión de las instancias 72 - 84, se genera un diagrama lógico 88 que define la lógica de conmutación.

Para evitar al máximo posible los errores en el posterior cableado y disposición del hardware, también se definen automáticamente en tiempo real una configuración de módulos del control de seguridad 1 y una asignación de terminales para el proceso de programación lógica descrito anteriormente. Para ello, en el software que genera la interfaz de programación 62 para cada símbolo de bloque funcional 66, 68, 70 está depositado un registro de datos, que especifica el requisito de recursos de hardware de este símbolo de bloque funcional 66, 68, 70. Esto incluye tanto el número requerido de terminales de entrada y salida 12 - 38 como determinadas funciones del control de seguridad 1 modular. Por ejemplo, el símbolo de bloque funcional 70 especial mencionado anteriormente puede especificar un requisito de recursos de hardware en el sentido de que también debe usarse un tipo de módulo de cabecera 2 que proporcione la funcionalidad de prensa correspondiente. Tal símbolo de bloque funcional 70 podría, por ejemplo, tener el requisito de recursos de hardware "dos terminales de entrada; dos terminales de salida; funcionalidad de prensa" depositado en el registro de datos asignado.

Al mismo tiempo, en el software, para todos los tipos de módulo 2, 4 y 6 existentes están depositados los recursos de hardware que los proporcionan, es decir, el número y tipo de terminales y funcionalidades específicas. Paralelamente, el software crea ahora durante la programación lógica automáticamente una configuración de módulos, es decir, una combinación del número, tipo y disposición de módulos 2, 4, 6. Esto se produce de tal manera que cada vez que durante cada proceso de arrastrar y soltar un símbolo de bloque funcional 66, 68, 70 se comprueba si la configuración de módulos almacenada actualmente en el software todavía puede proporcionar suficientes recursos de hardware libres. Como se ha descrito, el software conoce los recursos de hardware de cada módulo y a base de las instancias 72 - 84 depositadas en el diagrama lógico 88 y sus respectivos requisitos de recursos asignados para determinar cuántos de estos recursos ya están asignados. (Ejemplo: Al inicio del proceso de programación lógica, cuando la primera instancia 72 - 84 se almacena en la segunda subárea 86, la configuración de módulos no se compondrá de ningún módulo 2, 4, 6. Por consiguiente, no tiene recursos de hardware libres)

Si de la comprobación resulta que no están disponibles recursos de hardware libres, se cambia la configuración de módulos. Esto se hace de tal manera que sobre la base de todas las instancias 72 - 84 ancladas actualmente y sus

requisitos de recursos de hardware depositados se determina un requisito total de recursos de hardware. A continuación, de entre las combinaciones posibles de módulos 2, 4, 6, se determinan aquellas combinaciones que puedan proporcionar el requisito total de recursos de hardware. De entre estas combinaciones determinadas, se selecciona la que proporcione el menor requisito de recursos de hardware y se guarda como la nueva configuración de módulos actual. La respectiva configuración de módulos actual se muestra al usuario en una pestaña separada, no representada, en la pantalla 60, de modo que el usuario puede seguir en tiempo real el cambio en la configuración de módulos debido a sus ajustes en el diagrama lógico 88. Esto puede tener lugar, por ejemplo, en forma de texto o de gráfico.

La comprobación se realiza durante el proceso de arrastrar y soltar. Si el software no puede determinar una posible configuración de módulos que pueda proporcionar el requisito total de recursos de hardware, no es posible que el usuario deposite el símbolo de bloque funcional 66, 68, 70 seleccionado actualmente como una instancia 72 - 84 en la segunda subárea 86. La determinación de una configuración de módulos adecuada también está sujeta a otras condiciones marco que están depositadas en el software. Por un lado, se trata de condiciones de configuración basadas en el hardware, por ejemplo, que un determinado tipo de módulo solo puede estar presente en una determinada cantidad, o bien son condiciones que pueden ser ajustadas por un usuario, por ejemplo, que el control de seguridad 1 debe poder conectarse a un determinado tipo de bus de campo. Estas condiciones de configuración de módulos siempre se tendrán en cuenta en la determinación de una configuración de módulos adecuada.

Además de la configuración de módulos, el software también crea de forma totalmente automática una asignación correspondiente de los terminales 12 - 38 de los módulos 2, 4, 6 en la configuración de módulos actual. Para ello, cada vez que se seleccione un símbolo de bloque funcional 66, 68, 70 que tenga depositado un número de terminales como requisitos de recursos de hardware, se asigna y almacena una asignación de terminales. Los terminales 12 - 38 libres del control de seguridad 1 pueden, por ejemplo, asignarse en orden ascendente o puede usarse una función de asignación más compleja.

El mismo proceso tiene lugar de forma análoga cuando se eliminan instancias 72 - 84 del diagrama lógico 88. También entonces, se comprueba si es posible una configuración de módulos más óptima, es decir, que proporcione menos recursos de hardware. En caso afirmativo, esta configuración de módulos se deposita como configuración de módulos actual. Entonces, dado el caso, se reasignan los terminales. Si se eliminan instancias 72 - 84 que requieren un número de terminales como recursos de hardware, los terminales 12 - 38 asignados de forma correspondiente se habilitan en la asignación de terminales actual.

Una vez completada la programación lógica, el usuario puede ver tanto la configuración de módulos (ejemplo: Un módulo de cabecera con función de prensa, dos módulos de expansión de entrada, un módulo de pasarela) como la asignación de terminales (ejemplo: Parada de emergencia bicanal en el terminal A1, A2, etc.) en el software. De esta manera, dispone inmediatamente de toda la información para ensamblar y cablear el hardware. El software indica también los números de pedido de las piezas de hardware correspondientes y permite realizar pedidos directamente al fabricante. En algunas formas de realización, el usuario puede realizar incluso posteriormente reasignaciones manuales, lo que, sin embargo, puede volver a aumentar de nuevo el riesgo de errores.

Sobre la base del diagrama del circuito lógico 88 creado y de la configuración de módulos y la asignación de terminales creadas automáticamente, el software de programación, que también inicia la visualización de la interfaz de programación 62, crea a partir de la representación gráfica automáticamente los datos de configuración que contienen la información descrita anteriormente. A continuación, éstos son transferidos al control de seguridad 1 y son interpretadas y usadas por éste durante la ejecución del firmware allí almacenado. En otras realizaciones, el software de programación también puede crear directamente un firmware adecuado correspondientemente, que entonces se transfiere al control de seguridad 1 y se ejecuta allí.

**50 Lista de signos de referencia**

- 1 Control de seguridad
- 2 Módulo de cabecera
- 4 Módulo de expansión de entrada
- 55 6 Módulo de expansión de salida
- 8 Bus de plano posterior
- 10 Unidad de control
- 12 - 38 Terminal
- 40 - 46 Relé
- 60 48 Interruptor de parada de emergencia
- 50, 52 Interruptores
- 54 Ordenador
- 56 Teclado
- 58 Ratón
- 65 60 Pantalla
- 62 Interfaz de programación

	64	Primera subárea
	66, 68, 70	Símbolo del bloque funcional
	72 – 84	Instancia
	86	Segunda subárea
5	88	Diagrama lógico

## REIVINDICACIONES

- 5 1. Procedimiento para programar un control de seguridad (1) modular diseñado para encender y apagar de forma segura un consumidor eléctrico de acuerdo con una norma de seguridad, en el que por un procesador es representada en una pantalla (60) una interfaz de programación (62) con una programación lógica, y
- 10 en una primera subárea (64) de la interfaz de programación (62) es representada una pluralidad de símbolos de bloque funcional (66, 68, 70) que simbolizan señales de entrada de transmisores o señales de salida a actuadores, comprenden o bloques funcionales lógicos;
- 15 tras la selección de uno de los símbolos de bloque funcional (66, 68, 70) por parte de un usuario dentro de la programación lógica, una instancia (72–84) del símbolo de bloque funcional (66, 68, 70) seleccionado es representada en una segunda subárea (86) de la interfaz de programación (62), en el que la segunda subárea (86) está concebida de tal manera que las instancias (72–84) ya seleccionadas de símbolos de bloque funcional (66, 68, 70) son enlazadas lógicamente por el usuario con la ayuda de un diagrama lógico (86), y por el procesador, a base del diagrama lógico (68) creado y de una configuración de módulos y una asignación de terminales, son creados o bien datos de configuración que están concebidos para ser cargados al control de seguridad (1) modular y ser interpretados y usados por éste durante la ejecución de un firmware depositado allí, o bien es creado un firmware adaptado que está concebido para ser transferido al control de seguridad (1) modular y ser ejecutado allí.
- 20 **caracterizado porque**
- a cada uno de los símbolos de bloque funcional (66, 68, 70) está asignado un registro de datos que especifica un requisito de recursos de hardware, comprendiendo los recursos de hardware funcionalidades del control de seguridad (1) modular y/o un número de terminales, y
- 25 cuando es seleccionado el símbolo de bloque funcional (66, 68, 70) dentro de la programación lógica sobre la base del requisito de recursos de hardware especificado en el registro de datos asignado a este bloque funcional (66, 68, 70), se comprueba si una configuración de módulos actual del control de seguridad (1) modular puede proporcionar este requisito de recursos de hardware adicionalmente a un requisito de recursos de hardware actual, y
- 30 si la configuración de módulos actual no puede proporcionar este requisito de recursos de hardware, se cambia la configuración de módulos,
- en cuyo caso, cuando se cambia la configuración de módulos, a base de los requisitos de recursos de hardware, especificados en los respectivos registros de datos asignados a los símbolos de bloque funcional (66, 68, 70), de todas las instancias (72 – 84) ya seleccionadas de símbolos de bloque funcional (66, 68, 70) en la segunda subárea se determina un requisito total de recursos de hardware,
- 35 siendo cambiada la configuración de módulos de tal manera que la configuración de módulos cambiada pueda proporcionar el requisito total de recursos determinando de entre todas las combinaciones posibles de módulos (2, 4, 6) aquellas combinaciones que pueden proporcionar el requisito total de recursos de hardware.
- 40 2. Procedimiento según la reivindicación anterior, en el que la configuración de módulos incluye el tipo y el número de módulos (2, 4, 6) del sistema control de seguridad (1) modular.
3. Procedimiento según la reivindicación 2, en el que tras la selección de un símbolo de bloque funcional (66, 68, 70), en cuyo registro de datos asignado está depositado un número de terminales como requisito de recursos de hardware, se modifica una asignación actual de los terminales (12 – 38) de los módulos (2, 4, 6) del control de seguridad (1) modular.
- 45 4. Procedimiento según una de las reivindicaciones anteriores, en el que la configuración de módulos cambiada es la configuración de módulos que proporciona la menor cantidad posible de recursos de hardware, formando de todas las configuraciones de módulos posibles un mínimo a través de los recursos de hardware necesarios, y se selecciona y usa la configuración de módulos correspondiente a este mínimo.
- 50 5. Procedimiento según una de las reivindicaciones anteriores, en el que a base de las limitaciones de hardware están depositadas las condiciones de configuración de módulos,
- en el que, si la configuración de módulos cambiada no cumple las condiciones de configuración de módulos, se cancela la representación del símbolo del bloque funcional (66, 68, 70) en la segunda subárea (86) de la interfaz de programación (62).
- 55 6. Procedimiento según una de las reivindicaciones anteriores, en el que tras el borrado de una instancia (72 – 84) de un símbolo de bloque funcional (66, 68, 70) de la segunda subárea (68) por el usuario, se habilita en la configuración de módulos actual el requisito de recursos de hardware especificado en el registro de datos asignado a este símbolo de bloque funcional (66, 68, 70).
- 60 7. Procedimiento según la reivindicación 6, en el que tras la habilitación, a base de los requisitos de recursos de hardware de todas las instancias (72 – 84) ya seleccionadas de símbolos de bloque funcional (66, 68, 70), estando especificados en los respectivos registros de datos asignados a los símbolos de bloque funcional (66, 68, 70), en la segunda subárea (86) se determina un requisito total de recursos de hardware, y
- 65

- se comprueba si otra configuración de módulos con una cantidad menor de recursos de hardware puede proporcionar el requisito total de recursos de hardware, formando de todas las configuraciones de módulos posibles un mínimo a través de recursos de hardware requeridos,
- 5 y
- si sobre la base de la comprobación puede determinarse otra configuración de módulos de este tipo, que corresponda a este mínimo, la configuración de módulos actual se cambia a la otra configuración de módulos.
- 10 8. Producto de programa de ordenador que puede cargarse directamente a la memoria interna de un ordenador digital (54) y que comprende secciones de código de software que, cuando se ejecutan en el ordenador (54) , capacitan al ordenador (54) para ejecutar el procedimiento según una de las reivindicaciones anteriores.
- 15 9. Ordenador digital (54), a cuya memoria interna está cargado el producto de programa de ordenador según la reivindicación 8.
10. Producto de programa de ordenador que puede cargarse directamente a la memoria interna de un control de seguridad (1) y que comprende datos de configuración o un firmware adaptado que fueron creados por medio del procedimiento según una de las reivindicaciones 1 a 7.
- 20 11. Control de seguridad (1), diseñado para encender y apagar de forma segura un consumidor eléctrico de acuerdo con una norma de seguridad, programado con el procedimiento según una de las reivindicaciones 1 a 7.

FIG. 1

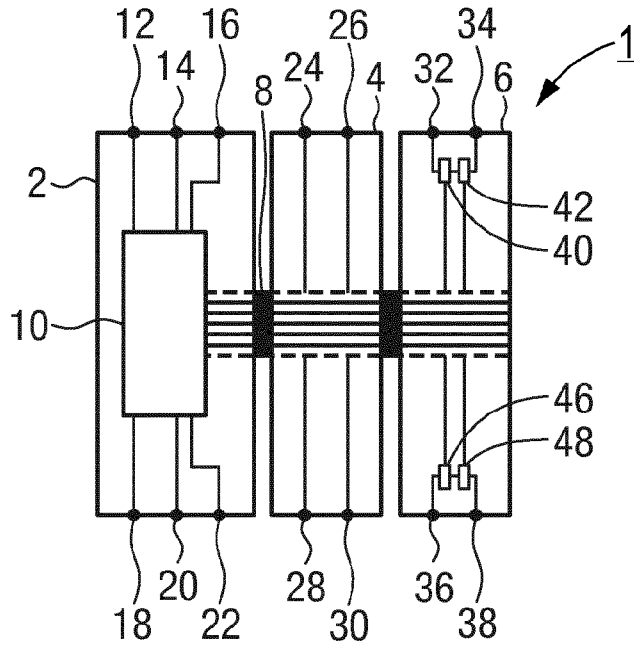


FIG. 2

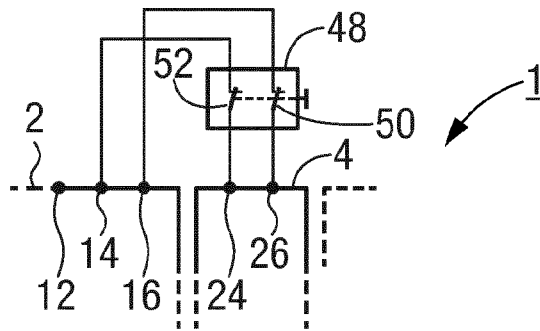


FIG. 3

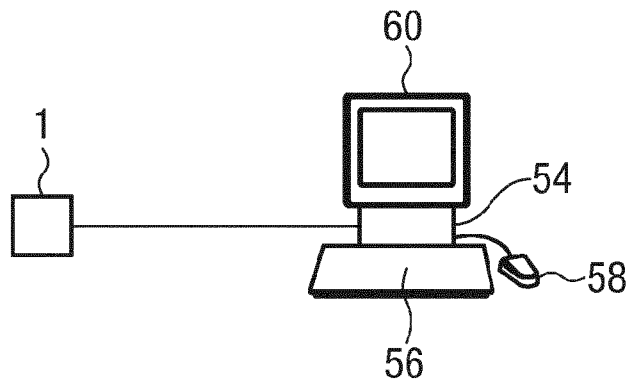


FIG. 4

