

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 483 966**

51 Int. Cl.:

G05B 19/05

(2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **20.04.2009** **E 09158215 (5)**

97 Fecha y número de publicación de la concesión europea: **16.07.2014** **EP 2244143**

54 Título: **Método y dispositivo para programar un controlador industrial de una instalación automatizada**

45 Fecha de publicación y mención en BOPI de la traducción de la patente:
08.08.2014

73 Titular/es:

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

72 Inventor/es:

CANTARELLI, MATTEO;
REUSCH, MATTHIAS;
WALTER, HERBERT;
STANKO, FLORIAN y
NAWRATIL, TIMO

74 Agente/Representante:

UNGRÍA LÓPEZ, Javier

ES 2 483 966 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

Método y dispositivo para programar un controlador industrial de una instalación automatizada

- 5 La presente invención se refiere a un método de programar un controlador industrial para controlar una instalación automatizada, incluyendo la instalación al menos un sensor para proporcionar datos de sensor que representan un estado real del sistema de la instalación y al menos un accionador para actuar en el estado real del sistema, e incluyendo el controlador una memoria de datos para almacenar los datos de sensor y una memoria de programa para almacenar un programa de código máquina, incluyendo el método los pasos de
- 10 - proporcionar una herramienta de programación que tiene un editor de programa, una herramienta de depuración y un compilador,
- 15 - crear un programa de control de alto nivel por medio del editor de programa, incluyendo el programa de control de alto nivel una pluralidad de instrucciones de control de nivel alto,
- compilar el programa de control de alto nivel por medio del compilador con el fin de crear un programa de código máquina incluyendo instrucciones de código máquina para leer y procesar repetidas veces los datos de sensor,
- 20 - cargar el programa de código máquina en la memoria de programa, y
- activar la herramienta de depuración para depurar el programa de control de alto nivel, donde las instrucciones de control de nivel alto se muestran en el editor de programa,
- 25 donde el programa de código máquina es ejecutado en el controlador mientras la herramienta de depuración está activada.
- La invención se refiere además a un dispositivo para programar un controlador industrial de una instalación automatizada, incluyendo la instalación al menos un sensor para proporcionar datos de sensor que representan un estado real del sistema de la instalación y al menos un accionador para actuar en el estado real del sistema, e incluyendo el controlador una memoria de datos para almacenar los datos de sensor y una memoria de programa para almacenar un programa de código máquina, incluyendo el dispositivo un editor de programa para crear un programa de control de alto nivel, incluyendo el programa de control de alto nivel una pluralidad de instrucciones de control de nivel alto, un compilador para compilar el programa de control de alto nivel con el fin de crear un programa de código máquina incluyendo instrucciones de código máquina para leer y procesar repetidas veces los datos de sensor, una interfaz para cargar el programa de código máquina en la memoria de programa del controlador, y una herramienta de depuración para depurar el programa de control de alto nivel, donde las instrucciones de control de nivel alto se muestran en el editor de programa.
- 40 Se conoce un método y un dispositivo de este tipo, por ejemplo, por EP 1 870 787 A1.
- En el mundo actual, muchos procesos técnicos se ejecutan automatizados o al menos parcialmente automatizados. Se puede ver ejemplos en emplazamientos de fabricación industriales y las plantas de producción, pero también en situaciones de la vida diaria en la recogida de equipajes en aeropuertos, telesillas en estaciones de esquí, montañas rusas en parques de ocio, etc. Las máquinas y otros componentes de tales instalaciones son operados por controladores industriales que reciben y procesan datos de sensor procedentes de sensores situados en la instalación y que producen señales de control para mover accionadores en la instalación. Por ejemplo, la posición operativa de un brazo de robot puede ser controlada automáticamente en función de los datos de sensor que representan una posición instantánea del brazo de robot y en función de una trayectoria deseada del brazo de robot definida por un programa de control. Las señales de control generadas por el controlador energizan un mecanismo de accionamiento eléctrico que mueve el brazo de robot en la dirección deseada. Las operaciones de control para controlar el brazo de robot pueden ser muy complejas. No obstante, es deseable tener una alta flexibilidad en el proceso de control. Por lo tanto, lo típico es usar controladores programables, es decir, controladores donde el control lógico se determina por un software que se denomina típicamente el programa de aplicación.
- 55 Hay varios lenguajes de programación de tipo especial que se usan típicamente para programar controladores industriales. En particular, el estándar internacional IEC 61131 define varios lenguajes de programación. Estos lenguajes de programación son lenguajes de programación de nivel alto en términos de la presente invención, porque no dan lugar a un programa de código máquina que sea ejecutable en un controlador específico. Más bien, un programa escrito en uno de los lenguajes de programación de nivel alto tiene que ser traducido a un programa de código máquina de nivel inferior de manera que sea ejecutable en un controlador específico. El proceso de traducción puede incluir varias etapas incluyendo código de programa de nivel intermedio. Por motivos de simplicidad, el término "compilador" se usa aquí para cualquier tipo de traductor adecuado incluyendo traductores línea a línea (intérpretes) y otros tipos de traductores. Igualmente, el término "programa de código máquina" se usa aquí para cualquier código de programa de nivel inferior que sea el resultado de un proceso de traducción y que pueda ser ejecutado en un controlador o una máquina equivalente que sea capaz de proporcionar verdaderos datos

de sensor procedentes de un proceso o instalación controlado.

El uso de un lenguaje de programación de nivel alto facilita en gran medida el proceso de programar un controlador industrial, porque el lenguaje de programación de nivel alto proporciona una pluralidad de instrucciones de control de nivel alto que permiten al programador centrarse en el problema de control y el control lógico más bien que en la implementación práctica en un hardware específico de un controlador. Consiguientemente, los lenguajes de programación de nivel alto se usan ampliamente.

No obstante, los programas de control para modernas aplicaciones de control altamente automatizadas pueden ser muy complejos, y a menudo es difícil identificar errores de programación. Por lo tanto, es común usar herramientas de depuración. Una herramienta de depuración o depurador es un programa de ordenador especial que se usa en el proceso de hallar errores de programación en otros programas, tal como el programa de aplicación. Típicamente, una herramienta de depuración permite ejecutar el programa de aplicación, que ha de ser depurado, en un modo de paso a paso (paso único) y/o con lo que se denomina puntos de interrupción. Un punto de interrupción es una orden de parada especial no usada en el transcurso normal del programa de aplicación, orden de parada que permite parar y pausar el programa de aplicación bajo examen en el punto de interrupción de modo que el programador pueda comprobar el estado logrado por el programa de aplicación en dicho punto de interrupción con tiempo suficiente.

EP 1 870 787 A1 antes citada describe un método para supervisar un programa de control cíclico que se ejecuta en un controlador de máquina. Una herramienta de depuración que se ejecuta en un dispositivo de programación genera una petición de supervisión que es transmitida al controlador de máquina. El controlador de máquina registra datos en respuesta a la petición de supervisión y devuelve los datos registrados al dispositivo de programación. Para que sea capaz de detectar datos que solamente son manipulados raras veces, el controlador de máquina registra no solamente los datos asociados con una operación de control específica, sino que también incrementa un contador. Como resultado, el usuario puede analizar los datos proporcionados desde el controlador de máquina según el contador respectivo.

EP 1 119 801 B1 describe un protocolo para supervisar la ejecución de código máquina en un controlador industrial. En una herramienta de depuración, las variables (direcciones de datos) son seleccionadas y asociadas a elementos de código. La información combinada es enviada al controlador antes de comenzar el programa. Una vez que el programa ha sido lanzado y uno de los elementos de código definidos es ejecutado, los valores de las variables asociadas son almacenados y transferidos de nuevo a la herramienta de depuración. La información es utilizada entonces para volver a enlazar los valores de las variables a los segmentos correspondientes en el código de programa de nivel alto.

EP 1 184 758 A2 describe un acercamiento para depurar un programa de aplicación para un controlador industrial, donde un modo de paso único y/o un modo de punto de interrupción pueden ser usados no solamente al nivel del código máquina, sino también al nivel del lenguaje de programación de nivel alto, y en particular a un nivel de diagrama de flujo. Consiguientemente, se puede buscar errores de programación en diferentes niveles, lo que se supone que facilita la detección de errores.

EP 1 184 759 A2 describe un método de programar un controlador industrial, donde se generan diagramas de flujo a partir de rutinas definidas por el usuario proporcionadas en un lenguaje de texto. Este documento también proporciona una vista general de los pasos para programar un controlador industrial usando un lenguaje de programación de nivel alto, un compilador e interfaces de depuración.

DE 10 2004 062 852 A1 describe un controlador industrial que tiene una función de depuración integrada. El depurador integrado permite ejecutar el programa de aplicación paso a paso y/o usando puntos de interrupción, como se ha descrito anteriormente. Sin embargo, puede haber situaciones donde un proceso controlado no pueda ser interrumpido con el fin de evitar, por ejemplo, movimientos no controlados de la máquina. Por lo tanto, DE 10 2004 062 852 A1 propone puntos de interrupción que pueden ser activados o desactivados selectivamente.

DE 10 2004 062 852 A1 afronta así un problema fundamental que es específico de programas de aplicación de depuración para controladores industriales en contraposición a programas informáticos de depuración para aplicaciones de sobremesa, por ejemplo, a saber, limitaciones resultantes de la incapacidad de parar una instalación controlada en cualquier momento. No es insólito que un programa de control complejo para un controlador industrial solamente pueda ser depurado en una ejecución simulada, es decir, sin mover realmente o controlar de otro modo la instalación.

Además, todavía puede ser un proceso muy lento y penoso depurar un programa de control complejo para un controlador industrial usando herramientas de depuración existentes.

En vista de estos antecedentes, un objeto de la presente invención es proporcionar un método y dispositivo alternativos para programar un controlador industrial incluyendo una característica de depuración mejorada para una depuración eficiente en el tiempo de programas de aplicación complejos.

Según un aspecto de la invención, este objeto se logra con un método como el mencionado al principio, determinando la herramienta de depuración una relación inversa entre el código máquina que se ejecuta en el controlador y al menos una instrucción de control de nivel alto mostrada en el editor de programa, leyendo la herramienta de depuración los datos de sensor de la memoria de datos, asignando la herramienta de depuración los datos de sensor procedentes de la memoria de datos a la al menos única instrucción de control de nivel alto, enlazando por ello la al menos única instrucción de control de nivel alto a un estado real del sistema durante la ejecución del programa de código máquina, donde la herramienta de depuración visualiza los datos de sensor en el editor de programa en una posición próxima a la instrucción de programa de nivel alto, y donde la herramienta de depuración incluye una base de datos incluyendo una pluralidad de símbolos gráficos cada uno de los cuales ilustra uno de una pluralidad de estados predefinidos del sistema de la instalación, presentando la herramienta de depuración uno de la pluralidad de símbolos gráficos en la posición próxima a la instrucción de programa de nivel alto en función de los datos de sensor asignados.

Según otro aspecto de la invención, este objeto se logra con un dispositivo como el mencionado al principio, donde la herramienta de depuración está configurada para iniciar la ejecución del programa de código máquina en el controlador, la herramienta de depuración está configurada para determinar una relación inversa entre el código máquina que se ejecuta en el controlador y al menos una instrucción de control de nivel alto mostrada en el editor de programa, la herramienta de depuración está configurada además para leer los datos de sensor de la memoria de datos, la herramienta de depuración está configurada para asignar los datos de sensor procedentes de la memoria de datos a la al menos única instrucción de control de nivel alto, enlazando por ello la al menos única instrucción de control de nivel alto a un estado real del sistema durante la ejecución del programa de código máquina, la herramienta de depuración está configurada para visualizar los datos de sensor en el editor de programa en una posición próxima a la instrucción de programa de nivel alto, y donde la herramienta de depuración incluye una base de datos incluyendo una pluralidad de símbolos gráficos cada uno de los cuales ilustra uno de una pluralidad de estados predefinidos del sistema de la instalación, presentando la herramienta de depuración uno de la pluralidad de símbolos gráficos en la posición próxima a la instrucción de programa de nivel alto en función de los datos de sensor asignados.

La herramienta de depuración del nuevo método y dispositivo no se basan en el uso de puntos de interrupción y/o un modo de paso a paso, aunque también se podría usar estas funciones de depuración convencionales, donde sea apropiado. Sin embargo, un aspecto central es establecer una relación inversa entre el código máquina que se ejecuta realmente en el controlador y al menos una y preferiblemente varias instrucciones de control de nivel alto mostradas en el editor de programa. Establecer esta relación inversa incluye identificar qué grupo de instrucciones de código máquina ejecutadas en el controlador corresponde a una instrucción de control de nivel alto concreta de modo que la al menos única instrucción de control de nivel alto mostrada en el editor de programa pueda ser enlazada a un estado real del sistema durante la ejecución del programa de código máquina. Determinar la relación inversa puede implicar, en cierta medida, un paso de descompilar el programa de código máquina con el fin de volver a traducir las instrucciones de código máquina a la al menos única instrucción de control de nivel alto. Sin embargo, podría haber implementaciones que no requieran un paso de descompilación. Por ejemplo, la base para la relación inversa se podría haber establecido ya en el proceso de compilación integrando referencias apropiadas al código máquina.

Establecer la relación inversa permite enlazar la instrucción de control de nivel alto y un estado específico del sistema de la instalación controlada, estado específico del sistema que es el resultado de las operaciones de control iniciadas por dichas instrucciones de control de nivel alto. En otros términos, la nueva herramienta de depuración correlaciona la al menos única instrucción de control de nivel alto y su efecto en el estado real del sistema de la instalación controlada.

La nueva herramienta de depuración establece automáticamente la nueva relación inversa y, preferiblemente, para cualquier instrucción de control de nivel alto del programa de aplicación.

Otro aspecto de la nueva herramienta de depuración es la asignación de datos de sensor reales leídos de la memoria de datos del controlador a la al menos única instrucción de control de nivel alto identificada como resultado de la relación inversa. Ambos aspectos proporcionan conjuntamente el enlace entre el estado real del sistema en un momento específico de tiempo durante el proceso de control y la instrucción de control de nivel alto responsable de este estado específico del sistema.

Según otro aspecto, la herramienta de depuración visualiza los datos de sensor, y en particular una variable de datos compleja, en el editor de programa en una posición próxima a la instrucción de programa de nivel alto. Esto permite un proceso de depuración muy conveniente y eficiente, porque el programador puede ver tanto instrucciones de control de nivel alto como el efecto de las respectivas instrucciones en la instalación real al mismo tiempo, es decir de un vistazo. Los datos de sensor reales son reflejados al código de programa de nivel alto, lo que hace muy fácil identificar errores de programación.

Según otro aspecto, la herramienta de depuración incluye una base de datos incluyendo una pluralidad de símbolos

gráficos cada uno de los cuales ilustra uno de una pluralidad de estados predefinidos del sistema de la instalación, presentando la herramienta de depuración uno (o más) de la pluralidad de símbolos gráficos en la posición próxima a la instrucción de programa de nivel alto en función de los datos de sensor asignados.

5 Esto es especialmente ventajoso, si se usan variables de datos complejas con el fin de reflejar estados de nivel alto del sistema al programa código. Por ejemplo, un símbolo gráfico del robot que representa el brazo de robot arriba o abajo puede ser usado con el fin de simbolizar un estado correspondiente del sistema del brazo de robot. Sin embargo, este refinamiento también podría ser usado para datos de sensor menos complejos, tal como un interruptor de encendido/apagado que pueda ser representado por un bit solamente.

10 Este refinamiento proporciona una interfaz muy conveniente y fácil de usar para el programador, porque los estados reales del sistema representados por los datos de sensor reales son reflejados en la programación de nivel alto.

15 Se apreciará que la nueva herramienta de depuración facilita un proceso de depuración eficiente, porque el programador puede comprobar fácilmente el efecto real de una instrucción de control de nivel alto concreta en la instalación controlada. Establecer el nuevo enlace entre el estado real del sistema y una instrucción de control de nivel alto concreta describe la causa en la cadena de causa y efecto entre el programa de control de alto nivel y la instalación controlada. Cuanto más complejo es el programa de aplicación, más facilita la nueva herramienta de depuración el proceso de depuración. Consiguientemente, el objeto anterior se logra completamente.

20 En un refinamiento preferido de la invención, la herramienta de depuración lee repetidas veces los datos de sensor de la memoria de datos, mientras el programa de código máquina es ejecutado en el controlador.

25 Según este refinamiento, la herramienta de depuración proporciona datos de sensor actualizados no solamente para un momento específico de tiempo, sino para períodos de tiempo más largos durante la ejecución del programa de control. Puede ser más fácil considerar las variaciones a largo plazo en el comportamiento de la instalación controlada en el proceso de depuración.

30 Según otro refinamiento preferido, el programa de código máquina es ejecutado en el controlador en tiempo real.

Según este refinamiento, el programa de código máquina no es ejecutado ni con puntos de interrupción ni con un modo de paso a paso específico. Más bien, el programa de código máquina es ejecutado sustancialmente de la misma forma que es ejecutado en el proceso de control verdadero.

35 Ejecutar el programa de código máquina en condiciones de la vida real proporciona una imagen realista del proceso de control y por lo tanto mejora más el proceso de depuración. Esto es especialmente verdadero para programas de aplicación y operaciones de control que tienen que ver con fuerzas naturales tales como gravedad, porque dichas fuerzas no pueden ser paradas o ralentizadas.

40 En otro refinamiento, una historia de datos de sensor leídos de la memoria de datos es asignada a la al menos única instrucción de control de nivel alto. Como una alternativa o además, la herramienta de depuración lee los datos de sensor de la memoria de datos y los asigna a la al menos única instrucción de control de nivel alto en función de una condición de disparo predefinida.

45 Ambos refinamientos pueden facilitar el análisis, por parte del usuario, de los datos recogidos del sensor en el caso de que los datos de sensor cambien muy rápidamente mientras el programa de código máquina es ejecutado en el controlador y/o en un bucle de programa que esté siendo examinado. Una historia de datos de sensor podría ser cualquier forma de una recogida de datos que muestre los valores actuales y anteriores (históricos) de los datos de sensor en el transcurso del tiempo. Una condición de disparo predefinida es cualquier condición que pueda ser definida por el usuario o automáticamente con el fin de identificar un cierto momento del tiempo para leer los datos de sensor deseados. Así, la condición predefinida proporciona una señal de disparo para leer los datos de sensor que son asignados a la al menos única instrucción de control de nivel alto.

50 Sin embargo, en otro refinamiento podría ser ventajoso ejecutar el programa de código máquina en el controlador en forma de paso a paso.

55 Este refinamiento es ventajoso si hay que supervisar y considerar una gran cantidad de datos de sensor en el proceso de depuración, tal como en escenarios controlados muy complejos. Utilizar una ejecución paso a paso del programa de control, si es posible en un escenario de la vida real, facilita la visión por parte del programador de todos los datos relevantes del sensor.

60 En otro refinamiento, el programa de control de alto nivel incluye una pluralidad de instrucciones de control de nivel alto en un orden secuencial que corresponde a una secuencia temporal durante la ejecución del programa de código máquina, asignando la herramienta de depuración los datos de sensor de la memoria de datos a la al menos única instrucción de control de nivel alto según la secuencia temporal.

Según este refinamiento, una y la misma variable de programa de nivel alto mostrada en el editor de programa puede tener valores diferentes sustancialmente en uno y el mismo tiempo (o al menos en el mismo ciclo operativo) dependiendo de la posición dentro del programa de control de alto nivel donde se use la variable de programa de nivel alto. Por ejemplo, una variable booleana puede tener el valor de "verdadero" después de que la al menos única instrucción de control de nivel alto haya sido ejecutada, pero la misma variable booleana puede tener el valor de "falso" antes de la ejecución de dicha instrucción de control de nivel alto. Este refinamiento refleja dicho cambio en el valor de una variable de programa de nivel alto como resultado de la secuencia temporal en la ejecución del programa de código máquina. Este refinamiento facilita más una depuración eficiente de un programa complejo de control de alto nivel explotando datos de sensor reales según la secuencia temporal o el comportamiento de temporización del programa de código máquina en el controlador. En otros términos, este refinamiento es ventajoso, porque toma en cuenta el comportamiento de temporización de la instalación controlada en el proceso de depuración.

Según otro refinamiento, se selecciona un número limitado de instrucciones de control de nivel alto a partir de la pluralidad de instrucciones de control de nivel alto, y la herramienta de depuración determina la relación inversa entre el código máquina que se ejecuta en el controlador y las instrucciones de control de nivel alto para el número limitado solamente.

En este refinamiento, solamente un número limitado de instrucciones de control de nivel alto están disponibles para el nuevo proceso de depuración. La limitación podría parecer desventajosa a primera vista, pero se ha observado que la herramienta de depuración puede ser operada más eficientemente, porque se precisa menos tiempo de procesado para establecer la relación inversa. En otros términos, el proceso de depuración se concentra en un número limitado de instrucciones de control de nivel alto que son de especial interés, y la relación inversa se determina para las instrucciones de control seleccionadas solamente. La relación inversa puede ser establecida mucho más rápidamente para el número limitado de instrucciones de control.

En otro refinamiento, la al menos única instrucción de control de nivel alto incluye una variable de programa de nivel alto, asignando la herramienta de depuración los datos de sensor procedentes de la memoria de datos a la variable de programa de nivel alto.

Este refinamiento proporciona una asignación muy simple y directa de datos de sensor reales a la al menos única instrucción de control de nivel alto. Consiguientemente, este refinamiento permite una implementación bastante fácil de la nueva herramienta de depuración. En otro refinamiento, sin embargo, la herramienta de depuración puede asignar los datos de sensor de la memoria de datos a una instrucción de control compleja incluyendo tanto programar órdenes como programar variables. Este último refinamiento incrementa la flexibilidad y el rendimiento de la nueva herramienta de depuración.

Según otro refinamiento, la al menos única instrucción de control de nivel alto corresponde a una pluralidad de datos de sensor de la memoria de datos, donde la herramienta de depuración determina una variable de datos compleja a partir de la pluralidad de datos de sensor, representando la variable de datos compleja un estado del sistema de nivel más alto de la instalación, y donde la herramienta de depuración asigna la variable de datos compleja a la instrucción de control de nivel alto.

Este refinamiento proporciona un acercamiento muy flexible y de alto rendimiento para un proceso de depuración eficiente. Combinar una pluralidad de diferentes datos de sensor en una variable de datos compleja y asignar dicha variable de datos compleja a la instrucción de control de nivel alto permite centrarse en el comportamiento del sistema general en el proceso de depuración en lugar de supervisar en detalle una pluralidad de datos de sensor. Por ejemplo, la variable de datos compleja puede representar la posición de un brazo de robot (superior o inferior, izquierda o derecha), donde una consideración cualitativa o generalizada puede corresponder a una pluralidad de datos de sensor procedentes de una pluralidad de diferentes codificadores de posición en varios ejes del robot. El uso de la variable de datos compleja en el proceso de depuración permite supervisar la posición general del brazo de robot, que es mucho más fácil y más eficiente que supervisar la pluralidad de diferentes datos de sensor, a condición de que la posición general del robot sea la cuestión principal en el proceso de depuración.

Otras características y detalles de realizaciones de la invención se explican más adelante con respecto a los dibujos, en los que:

La figura 1 representa una realización de una instalación automatizada incluyendo un controlador programable según la presente invención.

La figura 2 representa una realización del nuevo método y el nuevo dispositivo en una ilustración simplificada.

La figura 3 representa un editor de programa incluyendo instrucciones de control de nivel alto según el método y el dispositivo de la figura 2.

La figura 4 representa el editor de programa de la figura 3 conjuntamente con datos de sensor reales según una

realización preferida.

Y la figura 5 representa otra realización preferida de un editor de programa incluyendo instrucciones de control de nivel alto y símbolos de robot que representan datos reales del sistema.

La figura 1 representa una ilustración simplificada de una instalación 10 que es controlada y programada según una realización preferida de la presente invención. A modo de ejemplo, la instalación 10 incluye un robot 11 que tiene un brazo móvil de robot 12. Los movimientos del brazo de robot 12 son controlados por un controlador programable 14. El controlador 14 recibe señales de sensores que están configurados para medir o detectar cantidades físicas que representan un estado real del sistema tal como, por ejemplo, una posición o una velocidad del brazo de robot 12 en un momento definido del tiempo. Las señales de sensor son transmitidas al controlador 14 de modo que el controlador 14 reciba datos de sensor que representen el estado real respectivo del sistema medido o detectado por el sensor. A modo de ejemplo, un sensor 16 se representa en la figura 1 cerca de un dispositivo de accionamiento rotativo 18. El sensor 16 está configurado para detectar una posición de movimiento instantáneo del dispositivo de accionamiento 18. El dispositivo de accionamiento 18 está configurado para mover el brazo de robot 12 en respuesta a señales de control procedentes del controlador 14. El controlador 14 controla el dispositivo de accionamiento 18 en función de las instrucciones de control programadas según la presente invención y en función de los datos de sensor recibidos del sensor 16. No es necesario afirmar que la instalación 10 podría incluir una pluralidad de sensores y accionadores de varios tipos.

En la presente realización, el controlador 14 también recibe una señal de un interruptor de puerta de protección 20 que está configurado para supervisar una posición cerrada de una puerta de protección 22. El interruptor de puerta de protección 20 es un ejemplo típico de un sensor relacionado con la seguridad usado en modernas instalaciones automatizadas con el fin de evitar situaciones peligrosas. Otros sensores relacionados con la seguridad podrían incluir pulsadores de emergencia-apagado, redes de luz o sistemas de cámaras a prueba de fallos, tal como el sistema de cámara SafetyEYE® del cesionario. Según realizaciones preferidas de la invención, el controlador 14 está configurado para procesar datos de sensor relacionados con la seguridad y datos de sensor no relacionados con la seguridad (estándar) con el fin de controlar el robot 11 en función de ambos tipos de información.

En la presente realización, el controlador 14 no solamente controla el dispositivo de accionamiento 18, sino también contactores 24, 26 que están configurados para interrumpir el suministro de potencia al robot 11 con el fin de proporcionar una parada segura en respuesta a los datos de sensor procedentes de sensores relacionados con la seguridad, tal como un interruptor de puerta de protección 20, si es preciso. Se apreciará, sin embargo, que la presente invención también se puede usar para controladores programables que no procesen datos de sensor relacionados con la seguridad.

La figura 2 representa una realización preferida de una herramienta de programación 30 para programar el controlador 14. El controlador programable 14 también se muestra con detalles adicionales.

Como se puede ver en la figura 2, el controlador 14 incluye aquí dos microprocesadores redundantes 32, 34. Los microprocesadores 32, 34 están configurados para procesar instrucciones de control y datos de sensor con una redundancia de dos canales. Los dos microprocesadores comparan sus respectivos resultados de procesado y se supervisan uno a otro con el fin de proporcionar una condición a prueba de fallos requerida para operaciones de control relacionadas con la seguridad. No es necesario afirmar que el controlador 14 podría tener una redundancia de tres canales y/u otros tipos de funciones de prueba incorporadas con el fin de cualificarse como un controlador para controlar operaciones de control relacionadas con la seguridad según las categorías 3 o 4 del estándar europeo EN 954-1 o requisitos similares según las normas internacionales ISO 13849-1 o IEC 61508. Sin embargo, como ya se ha indicado anteriormente, el nuevo método y dispositivo también podrían ser usados para programar controladores que no estén cualificados como controladores de seguridad, aunque es preferible esto último dado que el nuevo método y dispositivo son especialmente adecuados para depurar programas de control relacionados con la seguridad.

El controlador 14 incluye una memoria de programa 36 para recibir un programa de código máquina 38 que incluye una pluralidad de instrucciones de código máquina 40. Se contempla que la memoria de programación 36 pueda incluir varias memorias de programa redundantes para almacenar varios programas de código máquina redundantes, uno para cada microprocesador redundante 32, 34. Por motivos de simplicidad, sin embargo, solamente se muestra aquí una memoria de programa 36, que proporciona los programas de código máquina para ambos microprocesadores 32, 34.

El controlador 14 también incluye una memoria de datos 38 para almacenar datos de sensor y para almacenar datos de accionador. Los datos de sensor son recibidos de sensores 16, 20 y son procesados por microprocesadores redundantes 32, 34 según las instrucciones de código máquina almacenadas en la memoria de programación 36. Como resultado del procesado de datos, se generan datos de accionador y se transmiten a los accionadores remotos 18, 24, 26 con el fin de mover apropiadamente los accionadores 18, 24, 26. Se apreciará que los datos de sensor almacenados en la memoria de datos 38 también podrían incluir datos de sensor preprocesados, tal como datos de 1 bit resultantes de un sensor redundante de 2 bits preprocesado.

La herramienta de programación 30 incluye un editor de programa 42, que se implementa preferiblemente en un ordenador personal de tipo general (PC). El editor de programa 42 permite escribir y editar un programa de control de alto nivel 44 incluyendo una pluralidad de instrucciones de control de nivel alto 46. Según realizaciones preferidas, el editor de programa 42 permite programar en uno o varios lenguajes de programación de nivel alto definidos en el estándar internacional IEC 61131.

La herramienta de programación 30 también incluye un compilador 48 para compilar el programa de control de alto nivel 44 con el fin de crear el programa de código máquina 38. El programa de código máquina 38 es transferido posteriormente a la memoria de programa 36 mediante una interfaz 50, que podría ser cualquier interfaz adecuada que permita transferir datos entre la herramienta de programación basada en PC 30 y el controlador 14.

Como apreciarán los expertos en la técnica, el compilador 48 y la interfaz 50 pueden incluir componentes de hardware y/o componentes de software del PC que también proporcione el editor de programa 42.

La herramienta de programación 30 incluye además una herramienta de depuración 52, que también se implementa como un componente de software y/o hardware en el PC en realizaciones preferidas. La herramienta de depuración 52 está configurada para leer datos de sensor que representan estados reales del sistema de la instalación de memoria de datos 38 mediante un enlace de comunicación 54. En algunas realizaciones, la herramienta de depuración 52 es capaz de leer los datos de sensor una y otra vez dentro de cuadros de tiempo definidos, de modo que una historia de datos de sensor sea leída realmente, como se ha indicado en el número de referencia 55.

La herramienta de depuración 52 también está configurada para identificar qué instrucciones de código máquina 40 son ejecutadas actualmente en el microprocesador 32 mediante un enlace de comunicación 56. En una realización, el código máquina ejecutado en los microprocesadores 32 y/o 34 podría estar configurado para escribir un identificador que identifique las instrucciones de código máquina actualmente ejecutadas en la memoria de datos 38 y/o en una memoria separada (no representada aquí), de modo que las instrucciones de código máquina actuales puedan ser identificadas fácilmente por la herramienta de depuración en base al identificador leído mediante el enlace 56.

Se contempla que herramienta de depuración 52 pueda estar configurada para proporcionar un campo 57 para definir una condición de disparo, que dispare el proceso de lectura a través de los enlaces 54 y/o 56. El campo 57 puede ser usado ventajosamente por un programador con el fin de definir una condición de disparo, tal como "Si brazo de robot 12 izquierdo = verdadero, entonces leer datos de sensor de memoria 38", por ejemplo.

La herramienta de depuración 52 interconecta los datos de sensor procedentes de la memoria de datos 38 con las instrucciones de control de nivel alto 46 mostradas en el editor de programa 42 de una manera que se explica con más detalle más adelante. Por lo tanto, la herramienta de depuración 52 está configurada para determinar una relación inversa entre el código máquina actualmente ejecutado en el microprocesador 32 y las instrucciones de control de nivel alto mostradas en el editor de programa 42 al mismo tiempo. En realizaciones preferidas, la herramienta de depuración 52 está conectada a una base de datos 58 incluyendo una pluralidad de símbolos gráficos, ilustrando cada símbolo gráfico uno de una pluralidad de estados predefinidos del sistema de robot 11. Por ejemplo, la base de datos 58 puede incluir símbolos gráficos que representen el robot 11 con el brazo de robot 12 en posiciones diferentes (superior, inferior, izquierda, derecha, etc). La herramienta de depuración 52 puede estar configurada para seleccionar uno de la pluralidad de símbolos gráficos en función de los datos de sensor reales recibidos mediante el enlace de comunicación 54 con el fin de reflejar simbólicamente un estado actual del sistema de robot 11 al programa de control de alto nivel 44, como se representa en las figuras 3 a 5 a modo de ejemplo.

La figura 3 representa el editor de programa 42 incluyendo una pluralidad de instrucciones de control de nivel alto 46. La pluralidad de instrucciones de control de nivel alto 46 forma un programa de control de alto nivel, que se escribe en un lenguaje de texto en la presente realización. También se contempla, sin embargo, usar un lenguaje de programación gráfica tal como el denominado diagrama de bloques de función definido en el estándar internacional IEC 61131. La figura 3 representa el editor de programa 42 sin herramienta de depuración 52 activada, es decir, sin la nueva función de depuración.

La figura 4 representa el mismo editor de programa 42, pero con un número limitado de instrucciones de control de nivel alto 46 y con la herramienta de depuración 52 activada. Como se puede ver en la figura 4, símbolos alfanuméricos 60, 62 y símbolos gráficos 64, 66 son visualizados conjuntamente con las instrucciones de control de nivel alto 46. Los símbolos 60, 62 visualizan un valor actual de la variable de programa de nivel alto 68, que es una variable llamada STATE en la realización representada. El valor actual de la variable de programación 68 es determinado por la herramienta de depuración 52 a partir de datos de sensor actuales.

Como se puede ver en la ilustración de la figura 4, la variable de programación 68 puede tener valores diferentes en el transcurso del proceso de control como resultado de las operaciones de control que se ejecuten y/o como resultado de los datos de sensor que cambien por otras razones. La herramienta de depuración 52 visualiza el valor actual respectivo de la variable de programa 68 según una línea de tiempo correspondiente a la secuencia temporal

69 de las instrucciones de control en el proceso de control. Por lo tanto, la variable de programación 68 podría tener un valor actual de "5" en una instrucción de control de nivel alto, y podría tener un valor actual de "6" en una instrucción de control posterior, como se representa en los símbolos 60, 62 en la figura 4.

- 5 Los símbolos 60, 62 visualizan valores alfanuméricos, que podrían ser útiles para variables de programa de nivel alto del tipo BYTE, PALABRA, ENTERO o REAL, que son tipos de datos comunes en lenguajes de programación según IEC 61131. En contraposición, los símbolos 64, 66 son símbolos gráficos que pueden ser usados ventajosamente para presentar un estado actual de una variable booleana. En la realización representada en la figura 4, los símbolos 64, 66 representan una "luz verde" que podría estar "encendida" o "apagada". De nuevo, sin embargo, el estado actual de cualquier variable de programación booleana simbolizada por símbolos 64, 66 se determina según la secuencia temporal de las operaciones de control y en función de datos de sensor reales.
- 10

- La figura 5 representa otro ejemplo de editor de programa 42 incluyendo un programa de control de alto nivel 44 compuesto de instrucciones de control de nivel alto 46. El programa 44 incluye dos variables booleanas 70, 72, y el estado actual respectivo de cada variable de programación 70, 72 en el transcurso del programa de control se indica por medio de símbolos gráficos 74, 76 que muestran aquí un pictograma del brazo de robot 12. Se contempla que los símbolos gráficos 74, 76 puedan mostrar una posición actual de brazo de robot 12 que se determina a partir de una pluralidad de diferentes datos de sensor. La herramienta de depuración 52 puede estar configurada para determinar una variable de datos compleja a partir de la pluralidad de datos de sensor. Por ejemplo, la variable de datos compleja puede corresponder a una pluralidad de datos de sensor que representen una pluralidad de información posicional para cada unión del brazo de robot 12. Los símbolos 74, 76 pueden ser una imagen del brazo de robot 12 en las posiciones respectivas, representando cada imagen una posición diferente y, en consecuencia, un valor diferente de la variable de datos compleja.
- 15
- 20

- 25 En las realizaciones preferidas, el programa de código máquina 38 es ejecutado en el controlador 14 sin puntos de interrupción y en tiempo real. Sin embargo, también se contempla que se pueda usar un modo de paso a paso y/o el uso de puntos de interrupción además del nuevo concepto de depuración, donde sea apropiado.

REIVINDICACIONES

1. Un método de programar un controlador industrial (14) para controlar una instalación automatizada (10), incluyendo la instalación al menos un sensor (16, 20) para proporcionar datos de sensor que representan un estado real del sistema de la instalación y al menos un accionador (18, 24, 26) para actuar en el estado real del sistema, e incluyendo el controlador (14) una memoria de datos (38) para almacenar los datos de sensor y una memoria de programa (36) para almacenar un programa de código máquina (38), incluyendo el método los pasos de:
 - proporcionar una herramienta de programación (30) que tiene un editor de programa (42), una herramienta de depuración (52) y un compilador (48),
 - crear un programa de control de alto nivel (44) por medio del editor de programa (42), incluyendo el programa de control de alto nivel (44) una pluralidad de instrucciones de control de nivel alto (46),
 - compilar el programa de control de alto nivel (44) por medio del compilador (48) con el fin de crear un programa de código máquina (38) incluyendo instrucciones de código máquina (40) para leer y procesar repetidas veces los datos de sensor,
 - cargar el programa de código máquina (38) en la memoria de programa (36), y
 - activar la herramienta de depuración (52) para depurar el programa de control de alto nivel (44), donde las instrucciones de control de nivel alto (46) se muestran en el editor de programa (42), donde el programa de código máquina (38) es ejecutado en el controlador (14) mientras la herramienta de depuración (52) es activada, **caracterizado** porque la herramienta de depuración (52) determina una relación inversa entre el código máquina (38) que se ejecuta en el controlador (14) y al menos una instrucción de control de nivel alto (46) mostrada en el editor de programa (42), leyendo la herramienta de depuración (52) los datos de sensor de la memoria de datos (38), asignando la herramienta de depuración (52) los datos de sensor de la memoria de datos (38) a la al menos única instrucción de control de nivel alto (46), enlazando por ello la al menos única instrucción de control de nivel alto a un estado real del sistema durante la ejecución del programa de código máquina (38), donde la herramienta de depuración (52) visualiza los datos de sensor en el editor de programa (42) en una posición próxima a la instrucción de programa de nivel alto (46) y la herramienta de depuración incluye una base de datos (58) incluyendo una pluralidad de símbolos gráficos (74, 76) de los que cada uno ilustra uno de una pluralidad de estados predefinidos del sistema de la instalación (10), presentando la herramienta de depuración (52) uno de la pluralidad de símbolos gráficos (74, 76) en la posición próxima a la instrucción de programa de nivel alto (46) en función de los datos de sensor asignados.
2. El método de la reivindicación 1, donde la herramienta de depuración (52) lee repetidas veces los datos de sensor de la memoria de datos (38), mientras el programa de código máquina (38) es ejecutado en el controlador (14).
3. El método de la reivindicación 1 o 2, donde el programa de código máquina (38) es ejecutado en el controlador en tiempo real.
4. El método de una de las reivindicaciones 1 a 3, donde una historia (55) de datos de sensor es leída de la memoria de datos (38) y asignada a la al menos única instrucción de control de nivel alto (46).
5. El método de una de las reivindicaciones 1 a 4, donde la herramienta de depuración (52) lee los datos de sensor de la memoria de datos (38) y los asigna a la al menos única instrucción de control de nivel alto (46) en función de una condición de disparo predefinida (57).
6. El método de la reivindicación 1 o 2, donde el programa de código máquina (38) es ejecutado en el controlador (14) en forma de paso a paso.
7. El método de una de las reivindicaciones 1 a 6, donde el programa de control de alto nivel (44) incluye una pluralidad de instrucciones de control de nivel alto (46) en un orden secuencial que corresponde a una secuencia temporal durante la ejecución del programa de código máquina (38), asignando la herramienta de depuración (52) los datos de sensor desde la memoria de datos (38) a la al menos única instrucción de control de nivel alto (46) según la secuencia temporal.
8. El método de una de las reivindicaciones 1 a 7, donde se selecciona un número limitado de instrucciones de control de nivel alto (46) a partir de la pluralidad de instrucciones de control de nivel alto (46), y donde la herramienta de depuración (52) determina la relación inversa entre el código máquina (38) que se ejecuta en el controlador (14) y las instrucciones de control de nivel alto (46) para el número limitado solamente.
9. El método de una de las reivindicaciones 1 a 8, donde la al menos única instrucción de control de nivel alto (46) incluye una variable de programa de nivel alto (68), asignando la herramienta de depuración (52) los datos de sensor procedentes de la memoria de datos (38) a la variable de programa de nivel alto (68).

10. El método de una de las reivindicaciones 1 a 9, donde la al menos única instrucción de control de nivel alto (46) se refiere a una pluralidad de datos de sensor procedentes de la memoria de datos (38), donde la herramienta de depuración (52) determina una variable de datos compleja (74) a partir de la pluralidad de datos de sensor, representando la variable de datos compleja un estado del sistema de nivel más alto de la instalación (10), y donde la herramienta de depuración asigna la variable de datos compleja a la instrucción de control de nivel alto (46).
11. Un programa de ordenador incluyendo código de programa configurado para llevar a la práctica un método según una de las reivindicaciones 1 a 10 con todos los pasos del método, cuando el código de programa es ejecutado en un ordenador que actúa como una herramienta de programación para programar un controlador industrial para controlar una instalación automatizada.
12. Un dispositivo para programar un controlador industrial (14) de una instalación automatizada (10), incluyendo la instalación (10) al menos un sensor (16, 20) para proporcionar datos de sensor que representan un estado real del sistema de la instalación (10) y al menos un accionador (18, 24, 26) para actuar en el estado real del sistema, e incluyendo el controlador (14) una memoria de datos (38) para almacenar los datos de sensor y una memoria de programa (36) para almacenar un programa de código máquina (38), incluyendo el dispositivo
- un editor de programa (42) para crear un programa de control de alto nivel (44), incluyendo el programa de control de alto nivel (44) una pluralidad de instrucciones de control de nivel alto (46),
 - un compilador para compilar el programa de control de alto nivel (44) con el fin de crear un programa de código máquina (38) incluyendo instrucciones de código máquina (40) para leer y procesar repetidas veces los datos de sensor,
 - una interfaz (50) para cargar el programa de código máquina (38) en la memoria de programa (36) del controlador (14), y
 - una herramienta de depuración (52) para depurar el programa de control de alto nivel (44), donde las instrucciones de control de nivel alto (46) se muestran en el editor de programa (42),
- caracterizado** porque la herramienta de depuración (52) está configurada para iniciar la ejecución del programa de código máquina (38) en el controlador (14), la herramienta de depuración (52) está configurada para determinar una relación inversa entre el código máquina (38) que se ejecuta en el controlador (14) y al menos una instrucción de control de nivel alto (46) mostrada en el editor de programa (42), la herramienta de depuración (52) está configurada además para leer los datos de sensor de la memoria de datos (38), la herramienta de depuración (52) está configurada para asignar los datos de sensor desde la memoria de datos (38) a la al menos única instrucción de control de nivel alto (46), enlazando por ello la al menos única instrucción de control de nivel alto (46) con un estado real del sistema durante la ejecución del programa de código máquina (38), la herramienta de depuración (52) está configurada para visualizar los datos de sensor en el editor de programa (42) en una posición próxima a la instrucción de programa de nivel alto (46), y la herramienta de depuración incluye una base de datos (58) incluyendo una pluralidad de símbolos gráficos (74, 76) de los que cada uno ilustra uno de una pluralidad de estados predefinidos del sistema de la instalación (10), visualizando la herramienta de depuración (52) uno de la pluralidad de símbolos gráficos (74, 76) en la posición próxima a la instrucción de programa de nivel alto (46) en función de los datos de sensor asignados.

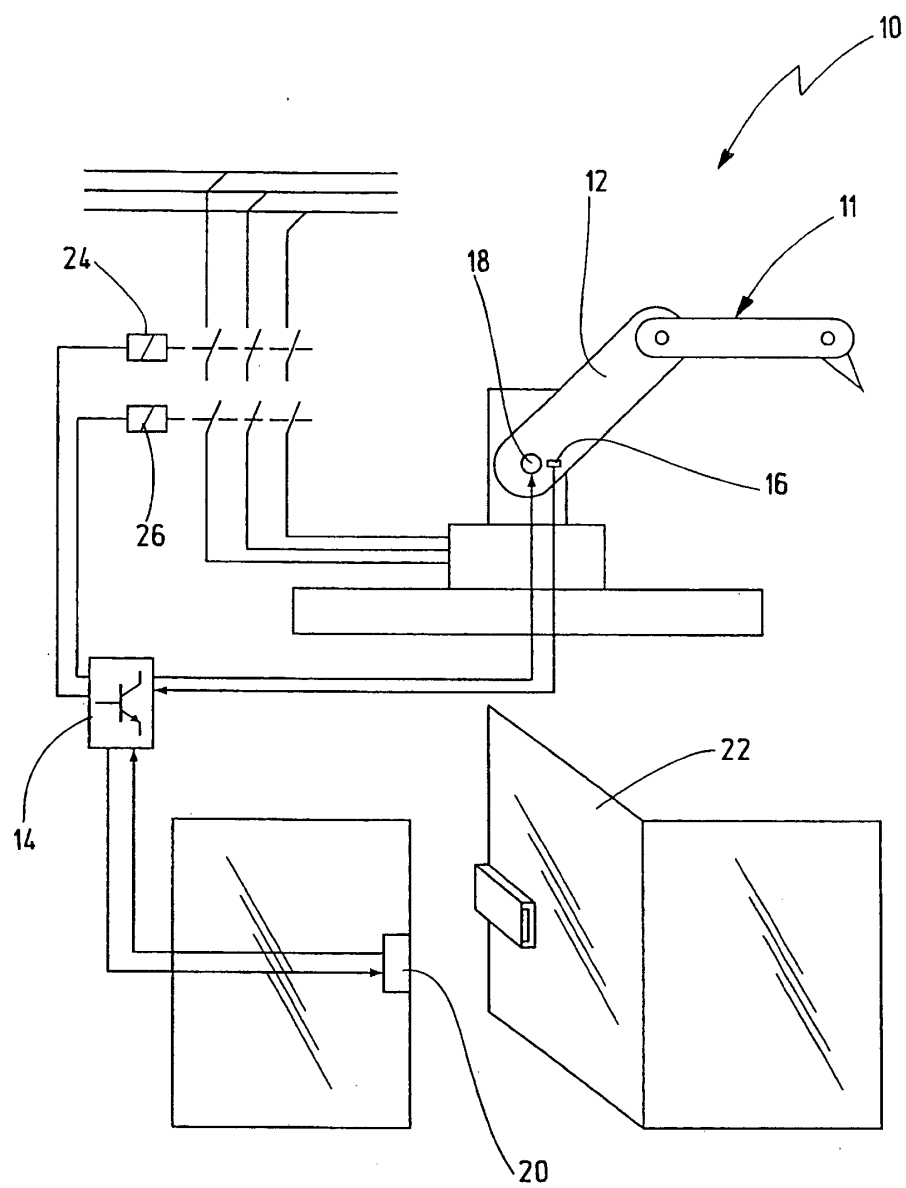


Fig.1

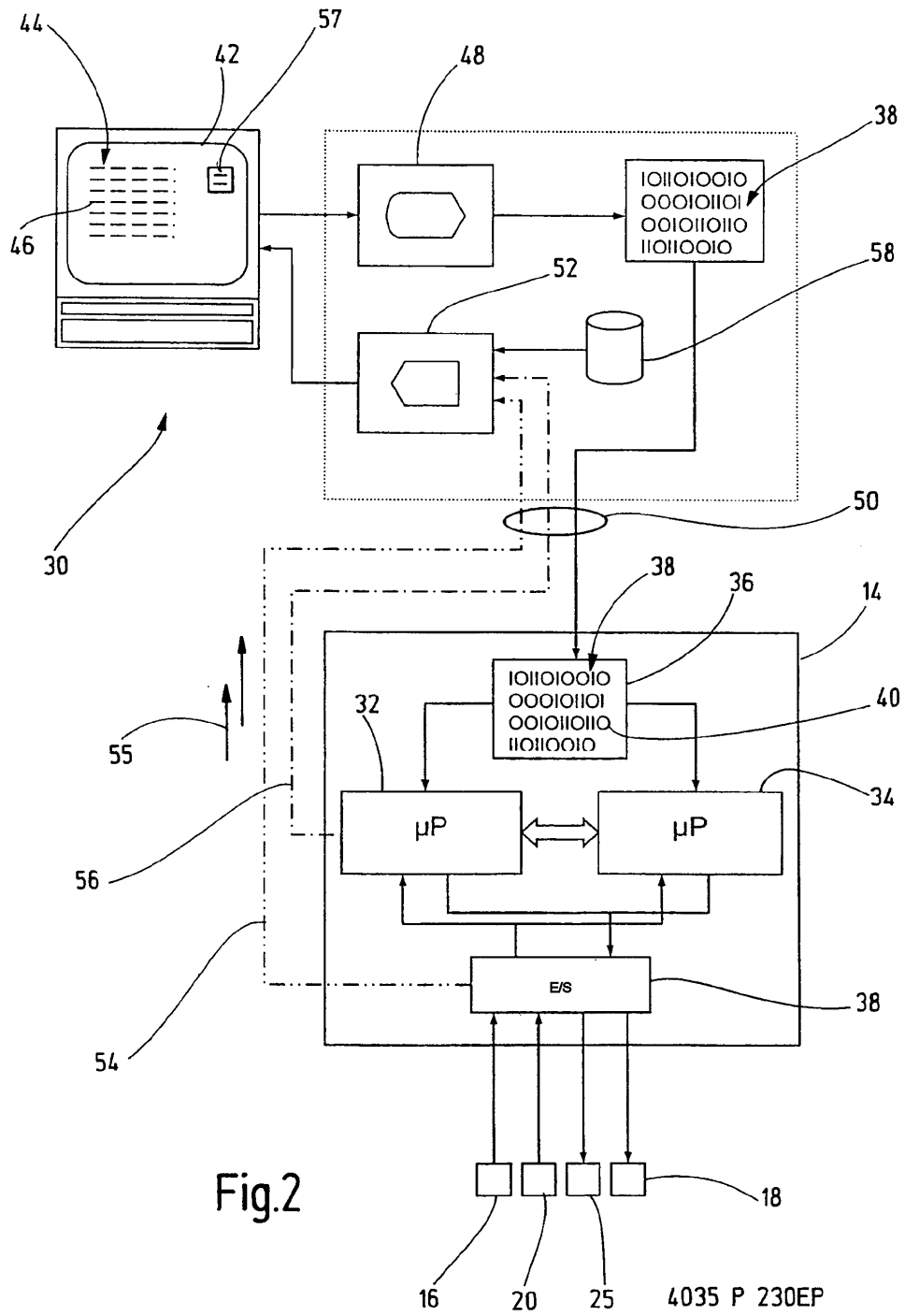


Fig.2

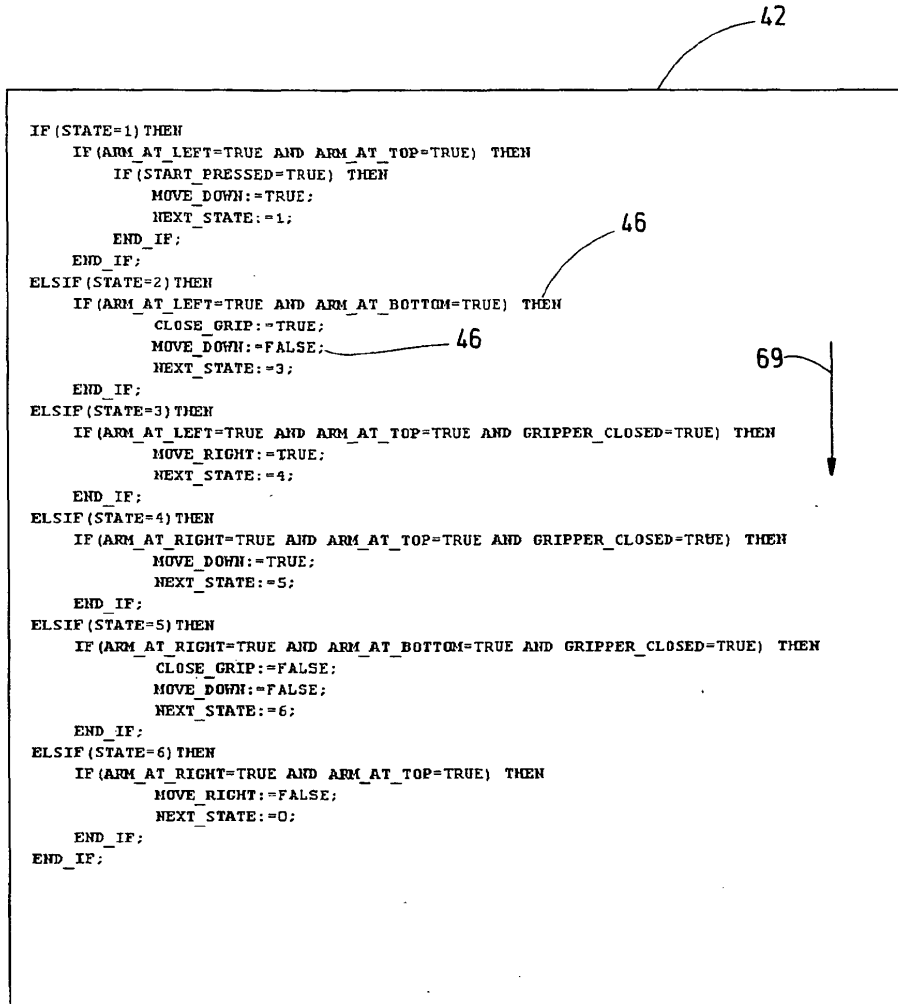


Fig.3

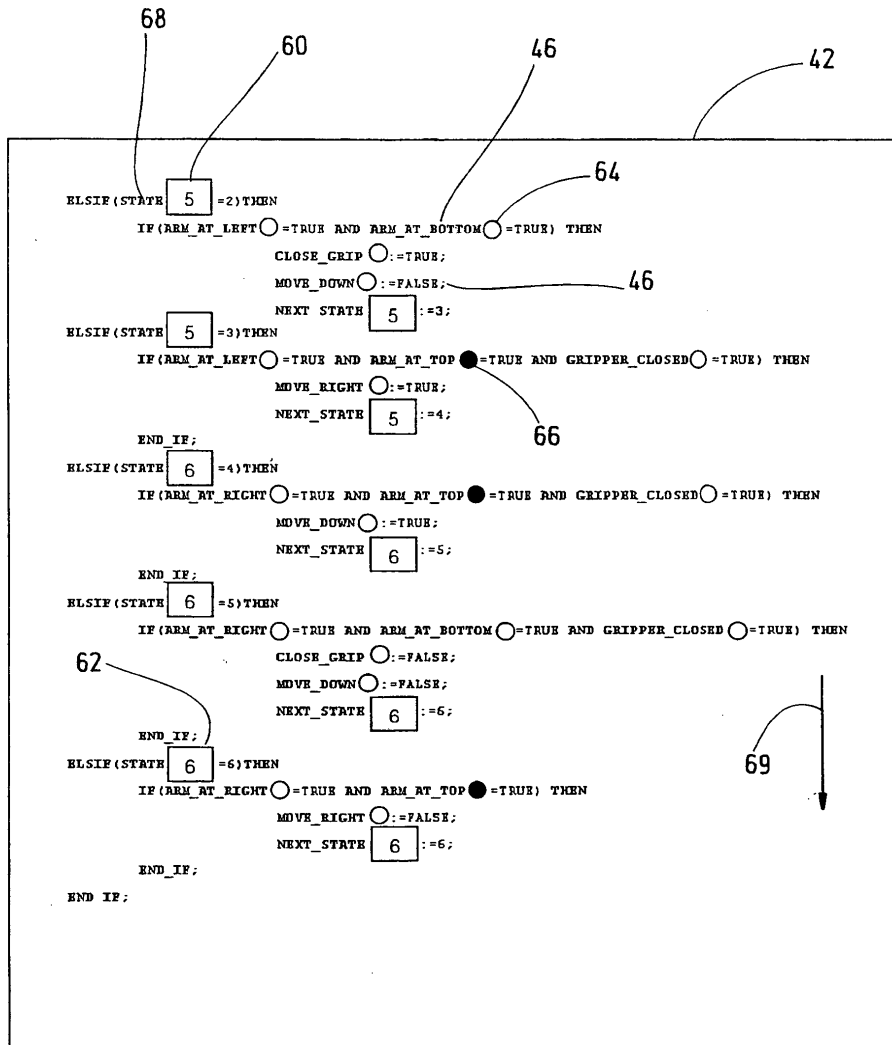


Fig.4

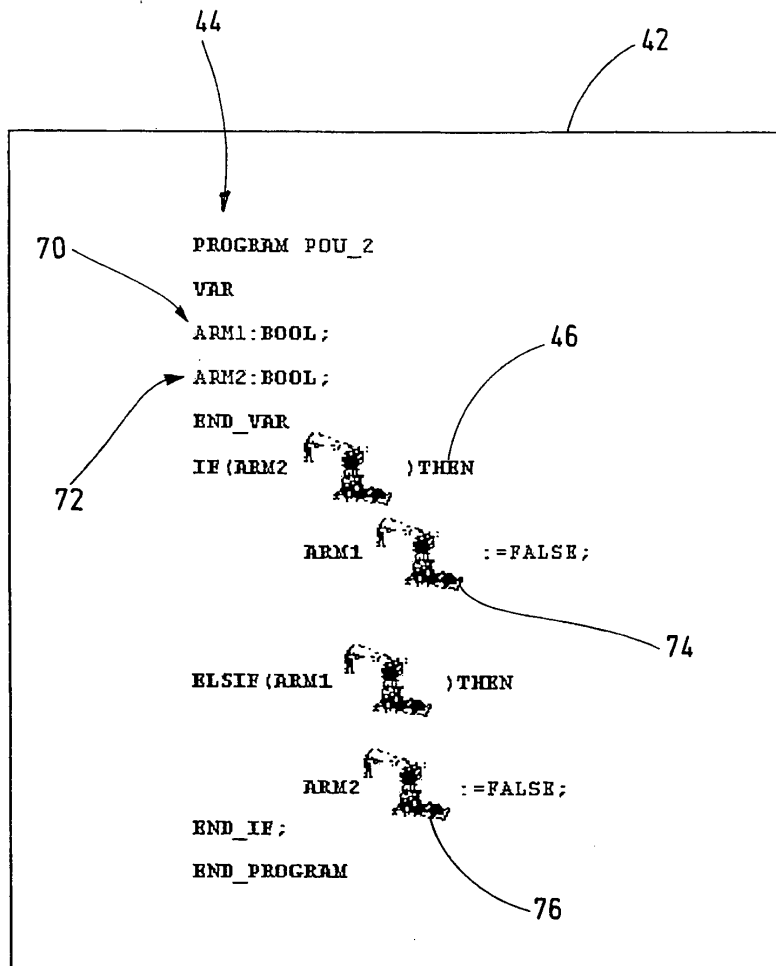


Fig.5