

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 394 015**

51 Int. Cl.:

G06F 3/041 (2006.01)

G06F 3/048 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **25.09.1997 E 97943551 (8)**

97 Fecha y número de publicación de la solicitud europea: **21.10.1998 EP 0871935**

54 Título: **Pizarra blanca electrónica con interfaz de usuario multifuncional**

30 Prioridad:

04.10.1996 US 725458

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

15.01.2013

73 Titular/es:

**POLYVISION CORPORATION (100.0%)
3970 Johns Creek Ct., Suite 325
Suwanee, GA 30024, US**

72 Inventor/es:

**GEAGHAN, BERNARD O.;
ATWOOD, STEPHEN y
SPRAGUE, JAMES**

74 Agente/Representante:

VALLEJO LÓPEZ, Juan Pedro

ES 2 394 015 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

Pizarra blanca electrónica con interfaz de usuario multifuncional

Campo de la invención

5 La presente invención se refiere en general al campo de los dispositivos de entrada electrónicos amigables con el usuario para su uso con ordenadores de finalidad general.

Antecedentes

10 Las pizarras blancas son un medio bien conocido para facilitar las reflexiones personales y las discusiones en grupo proporcionando una superficie conveniente sobre la que se pueden realizar notas, dibujos, gráficos u otras anotaciones. Como con las pizarras tradicionales, las pizarras blancas permiten que se realicen anotaciones en múltiples colores y se borren a continuación. Las pizarras blancas ofrecen varias ventajas sobre las pizarras incluyendo una superficie blanca limpia que proporciona un mayor contraste que el fondo verde tradicional de las pizarras. Adicionalmente, la escritura en una pizarra blanca es más fácil para muchos que sobre la pizarra tradicional. Por ejemplo, la superficie de escritura suave de la pizarra blanca permite un uso más fácil de marcadores de punta suave borrables usados sobre pizarras blancas, mientras que la superficie de la pizarra proporciona una superficie de alguna manera áspera para retener la tiza usada para la escritura en tales superficies. Además, muchos usuarios prefieren la pizarra blanca a la pizarra simplemente debido a que el marcador se puede agarrar más fácilmente que la tiza y no mancha las manos del usuario cuando se agarran.

15 Recientemente, se han desarrollado pizarras blancas para permitir que las escrituras y anotaciones del usuario introducidas sobre la pizarra blanca sean transmitidas a un ordenador personal para su almacenamiento, visualización y manipulación. Tales dispositivos permiten que las imágenes y anotaciones realizadas sobre la pizarra blanca sean guardadas en un ordenador, para ser visualizadas, para ser impresas, transmitidas o manipuladas. En tanto que tales dispositivos incrementan la versatilidad y utilidad de la pizarra blanca tradicional, continúa existiendo la necesidad de una pizarra blanca que mejore los mecanismos para introducir imágenes y anotaciones en un ordenador por medio de la pizarra blanca, y que permita que se realicen las operaciones posteriores sobre los datos almacenados recibidos desde la pizarra blanca.

20 El documento US5500935 desvela un sistema de ordenador usado con hardware de entrada especial para proporcionar los beneficios de la computerización a la escritura manual así como a la información introducida desde el teclado para satisfacer un amplio espectro de necesidades de proceso de la información comúnmente usada, escrita. El hardware de entrada consiste en una tableta de digitalización transparente montada por encima de una pantalla de visualización de panel plano montada por encima de modo que el usuario pueda escribir manualmente directamente sobre la pantalla de visualización con un lápiz adjunto. Las funciones del programa de aplicación incluyen la documentación de apuntes y planificación de trabajo, escritura manual o creación de archivos de texto y manipulación, funciones de enlace de autor a teclado para producción de documentación, anotación manual al texto, creación de notas manuales a partir del texto y creación de documentos de texto a partir del material de origen escrito manualmente.

25 El hardware especial de entrada consiste en un dispositivo electrónico tal como una tableta de dibujo, panel activo o pizarra blanca con una superficie de trabajo y funciones accesibles que incluyen nueva operación (modo dibujo), borrar, mover, reducir, imprimir y así sucesivamente; también se puede seleccionar el color del rotulador.

30 El documento US 5063600 desvela un sistema de ordenador usado con hardware de entrada especial para proporcionar los beneficios de la computerización a la escritura manual así como a la información introducida desde el teclado para satisfacer un amplio espectro de necesidades de proceso de la información comúnmente usada, escrita. El hardware de entrada consiste en una tableta de digitalización transparente montada por encima de una pantalla de visualización de panel plano montada por encima de modo que el usuario pueda escribir manualmente directamente sobre la pantalla de visualización con un lápiz adjunto. El usuario puede elegir también el modo, o bien escogiendo los apartados desde menús desplegados o bien activando botones gráficos tocándolos. Las funciones del programa de aplicación incluyen la documentación de apuntes y planificación de trabajo, escritura manual o creación de archivos de texto y manipulación, funciones de enlace de autor a teclado para producción de documentación, anotación manual al texto, creación de notas manuales a partir del texto y creación de documentos de texto a partir del material de origen escrito manualmente.

35 El documento US-A-5072076 se refiere a una tableta digitalizadora que se puede usar con una punta no ligada que puede ser tan simple como un lapicero o un rotulador de punta suave. Se puede crear un dibujo que corresponderá al dibujo real de la punta y sacado a una hoja escrita.

40 El artículo XP000473784, publicado en BRIDGES BETWEEN WORLDS. AMSTERDAM, 24 – 29 de abril de 1993; [PROCEEDINGS OF THE CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS. (INTERCHI)], READING, ADDISON WESLEY, Estados Unidos, vol. – páginas 391 - 398, 24-04-1993; PEDERSEN E R ET AL: "TIVOLI: AN ELECTRONIC WHITEBOARD FOR INFORMAL WORKGROUP MEETINGS" desvela una pizarra blanca electrónica que comprende un deslizante blanco para dibujar en él y paneles de control a lo largo de los

bordes inferior e izquierdo, un modo de borrado, así como funciones de “guardar”, “Nueva Presentación” e “Imprimir”. La funcionalidad núcleo incluye hacer trazados con un lápiz simple y edición, páginas múltiples, guardar y recuperar, imprimir e importar imágenes.

Sumario de la invención

5 Es un objetivo principal de la presente invención proporcionar un sistema de pizarra blanca de acuerdo con la reivindicación 1.

Las entradas gráficas del usuario se pueden almacenar también en el ordenador para recuperación, manipulación e impresión posterior. El material almacenado previamente se puede revisar posteriormente a conveniencia del usuario. El borrado de las entradas es también capturado y almacenado.

10 Ventajosamente, los sistemas de pizarra blanca que operan de acuerdo con la presente invención permiten al usuario trabajar con la pizarra blanca de dos maneras. Primero, se pueden almacenar las entradas gráficas del usuario en un archivo para recuperación posterior. Segundo, los archivos de la sesión almacenados previamente se pueden recuperar para ver, imprimir y/o copiar páginas a otras aplicaciones y para almacenar, manipular, visualizar e imprimir la información. Las funciones precedentes se proporcionan preferentemente por un software de aplicación de pizarra blanca que opera sobre el PC.

15 El software de aplicación de pizarra blanca implementa una función de grabación que opera para permitir que las entradas de usuario gráficas y ciertos comandos introducidos a través de la pizarra blanca sean almacenados en un archivo de sesión de pizarra blanca, que es un registro de toda la actividad sobre la pizarra blanca desde que se creó el archivo. El usuario pueda realizar el registro de la actividad de la pizarra blanca agrupando ciertas actividades en el mismo o en diferentes archivos o, alternativamente, almacenando todas las actividades en un único archivo.

20 Se puede crear una vista instantánea o bien por el usuario o bien automáticamente por el software de aplicación para permitir la creación de una imagen que sea la composición de todas las entradas del usuario gráficas desde un primer marcador insertado en el archivo de sesión hasta un segundo, posterior, marcador insertado en el archivo de sesión. El software de aplicación implementa una función de visor para visualizar todas las instantáneas en un archivo de sesión y para permitir la selección, visión y manipulación de una instantánea particular. Por ejemplo, se puede seleccionar una instantánea, verse y moverse o copiarse de un archivo de sesión a otro. Además, se puede seleccionar una instantánea y exportarla en un formato de archivo de datos seleccionado para manipulación mediante otro programa de aplicación. Ventajosamente, la secuencia en la que se introdujeron las entradas gráficas del usuario, junto con la información de tiempo indicativa de la relación temporal en la que se introdujeron las entradas, se almacenan en orden para permitir una visión posterior de la forma en la que las entradas se introdujeron y también para permitir la edición de la escritura individual o borrado de trazados. Como se ha usado en el presente documento, el término “trazado” se define para indicar puntos de datos generados desde el momento en que el marcador u otro instrumento tal como un dedo humano o un borrador se presionan sobre la superficie de escritura de la pantalla blanca con suficiente presión para producir la transmisión de datos desde la pantalla blanca al PC hasta el momento en que la presión anteriormente mencionada aplicada a la superficie de escritura se finaliza levantando el marcador, dedo o borrador o mediante el contacto con cualquier borde de la pizarra blanca. Además, nada que el usuario pueda hacer sobre la pantalla blanca hará que los datos sean eliminados del archivo de sesión. Naturalmente los archivos se pueden borrar en la forma normal desde el PC.

30 El software de aplicación recibe preferentemente la información transmitida desde la pantalla blanca por medio de un controlador que recibe una información de flujos de bits, formateados como paquetes, desde la pantalla blanca y realiza un número de funciones sobre los bits recibidos antes de almacenar la información en una memoria intermedia para su recuperación por parte del software de aplicación. Por ejemplo, el controlador realiza ciertas funciones de comprobación de error en el flujo de bits recibido y descarta paquetes que se determina estén corrompidos. El software de aplicación inserta ventajosamente un comando de levantar rotulador, que indica que el marcador, o borrador, usado por usuario se ha levantado de la superficie de escritura de la pizarra blanca, para evitar un fenómeno conocido como trazado, en el que el paquete corrupto, descartado por el controlador, puede tener incluido un comando de levantar rotulador.

40 Estas y otras características y ventajas de la invención se comprenderán mejor teniendo en cuenta la descripción detallada a continuación.

Breve descripción de los dibujos

55 La Figura 1 es un diagrama de bloques esquemático de una realización preferida;
 la Figura 2 es una representación esquemática del flujo de información en el sistema de la Figura 1;
 la Figura 3 es un diagrama de bloques de los componentes principales de un equipo de control contenido en la pizarra blanca de la Figura 1;
 la Figura 4 es una representación esquemática de los componentes funcionales principales de la aplicación de software ejecutada por el ordenador la Figura 1;
 la Figura 5 es un diagrama de bloques del archivo de sesión de la Figura 4;

la Figura 6 es un diagrama de flujo que muestra el funcionamiento de la parte de la aplicación de software de la Figura 3 y

la Figura 7 es un diagrama de bloques esquemático de una implementación alternativa del software de aplicación de pizarra blanca de la Figura 1.

5 **Descripción detallada**

I. Visión general

La Figura 1 de los dibujos muestra una realización preferida que incluye una pizarra blanca 10 conectada a un ordenador de sobremesa de finalidad general 12. El ordenador de sobremesa tiene la forma preferentemente de un ordenador personal (PC) que funciona con el sistema operativo Windows95™ o Windows 3.1™, ambos disponibles en Microsoft Corporation, Redmon, Washington. La pizarra blanca 10 emplea preferentemente tecnología de membrana resistiva convencional sensible a la presión aplicada sobre la superficie de escritura de la pizarra blanca 14. En respuesta a la entrada aplicada presionando sobre la superficie de escritura de la pizarra blanca 14, se generan dos tensiones, indicadores de unas coordenadas x e y de la entrada para su uso por un equipo de control 16 (mostrado en la Figura 2) que convierte las dos tensiones analógicas en coordenadas digitales para la transmisión al PC 12 que incluye una pantalla 13 y está conectado a una impresora 15. El PC está conectado también a un dispositivo de impresión y a otros dispositivos de entrada tal como un ratón 17 y un teclado 19 para recibir entradas para el control e inicio de diferentes funciones en el PC. El equipo de control reconoce preferentemente las entradas del usuario aplicadas en una zona de escritura 18 de la pizarra blanca como datos de entrada gráficos y las entradas del usuario aplicadas en una zona de botones 20 de la pizarra blanca como entradas de comandos correspondientes a uno de la pluralidad de comandos, para la transmisión de información de control que es o bien almacenada en el PC o bien interpretada por el PC para ejecutar una función predefinida. La Figura 1 muestra en 23 una imagen que se puede crear por medio del marcador 24 sobre la zona de escritura 18. Cuando se desea crear nuevas imágenes, se puede usar el borrador 25 para borrar la imagen 23 de la zona de escritura 18. El borrador 25 tiene una forma ventajosamente circular para permitir una pluralidad de funciones de borrado como se describe con mayor detalle en el presente documento.

La pizarra blanca 10 incluye también un altavoz 48 para la transmisión de información audible para alertar al usuario sobre acciones particulares que se están realizando en la pizarra blanca y para otras informaciones de estado. Se proporciona también una indicación visual en la forma de un Diodo Emisor de Luz (LED) 46 para indicar al usuario que se está suministrando alimentación a la pizarra blanca través del cable de alimentación 47 y que la pizarra blanca está operativa.

La zona de botones o control 20 incluye preferentemente los siguientes botones que se pueden tocar por usuario o bien con un dedo o mediante el uso de un marcador 24 para producir la transmisión de una señal por el equipo de control 16 al PC que es indicativa del comando deseado: (a) cuatro botones de colores incluyendo rojo, verde, azul y negro, (b) un botón de borrador estrecho, (c) un botón de borrador ancho, (d) un botón de instantánea/borrar todo, (e) un botón de instantánea, (f) un botón de impresión y (g) un botón de mostrar tablero. Se proporciona a continuación una explicación más detallada de las funciones realizadas por cada uno de estos botones.

El PC 12 tiene preferentemente almacenado en él un software de aplicación de pizarra blanca anteriormente mencionado que implementa una función de grabación para registrar datos de entrada del usuario introducidos por medio de la superficie de la pizarra blanca 14 y una función de visor para la visualización y para permitir la manipulación de los datos registrados. Se muestra una vista ampliada de la pantalla 13 en la Figura 1 en 55. El software de aplicación, proporciona y acepta información por medio del fácil uso de una Interfaz de Usuario Gráfica (GUI).

Como se ve en 55, la pantalla 13 visualiza la vista dentro de una ventana que preferentemente contiene tres zonas: una zona de selección 27, una zona del tablero actual 28 y una zona de instantánea 29. La zona de selección 27 visualiza, como se muestra en 30, una imagen que el usuario ha seleccionado para visualizar. La imagen en la zona 27 puede ser o bien una imagen que se ha introducido y almacenado previamente o puede ser una imagen representativa de las entradas que se están realizando actualmente sobre la pizarra blanca. La zona del tablero actual 28 visualiza una imagen representativa de las entradas que se están introduciendo actualmente sobre la pizarra blanca. Por ello, si el usuario ha seleccionado la imagen actual para visualización en la zona de selección entonces la zona de selección 27 y la zona de la imagen actual 28 mostrarán la misma imagen. La zona de instantáneas 27 contiene una versión en tamaño reducido de cada instantánea contenida en el archivo de sesión abierto actualmente. Una barra de desplazamiento con botones tal como se puede ver en 51 permite al usuario desplazarse arriba y abajo a través de las imágenes para seleccionar una imagen. Una vez seleccionada, la imagen se muestra en la zona de selección 27.

La Figura 2 de los dibujos muestra el flujo lógico de la información desde la pizarra blanca 10 al PC 12. Las entradas de usuario gráficas generadas por medio de la zona de escritura 18 y la zona de control 20 se transmiten al equipo de control que genera un flujo de paquetes de datos, uno de los cuales se ve en 60, para la transmisión al PC 12. Un controlador 62 ejecutado por el PC recibe los paquetes de datos 60, realiza cierta comprobación de errores y otras funciones sobre los datos y un programa de aplicación 64 ejecutado por el PC recibe los datos desde el controlador,

elimina los datos redundantes y almacena los datos en un archivo de sesión 66. El paquete de datos se formatea en una forma explicada con mayor detalle a continuación en conexión con el comando *Formateo de Tableta* implementado por el equipo de control. El archivo de sesión se formatea en una forma explicada más completamente en conexión con la Figura 5.

5 **II. Pizarra blanca**

La pizarra blanca 10 puede tomar una variedad de tamaños dependiendo de la localización en la que se va a usar. Por ejemplo, en una realización preferida tiene las dimensiones de 60,96 x 91,44 cm (24 x 36 pulgadas) para permitir su uso en una oficina o pequeño despacho por un usuario individual. Las pizarras blancas de mayores tamaños pueden encontrar uso en salas de conferencia o áreas de reunión para su uso durante discusiones en grupo. Como se ha observado anteriormente, la pizarra blanca emplea preferentemente un sensor que usa una tecnología de membrana resistiva para detectar la posición de las entradas del usuario sobre la superficie de escritura 14. Mientras que se puede emplear cualquiera de entre un número de tecnologías para detectar las entradas del usuario, la tecnología de membrana resistiva se ha encontrado que tiene un cierto número de ventajas incluyendo la durabilidad, fiabilidad, coste relativamente bajo. Además, la tecnología de membrana resistiva permite una simplicidad de fabricación.

La superficie de escritura 14 permite ventajosamente el uso de marcadores de punta suave borrables convencionales tales como se ven en 24. La zona de escritura 18 de la pizarra blanca es blanca y por ello proporciona un elevado grado de contraste con la mayoría de los colores de los marcadores. La superficie de escritura 14 puede ser borrada y limpiada con un borrador 25 que facilita el borrado de la pizarra blanca en una forma que se va a explicar o mediante la mayor parte de otros paños, tejidos desechables convencionales, etc.

El sensor se sitúa bajo la superficie de escritura de la pizarra blanca. Consiste en dos hojas de material que están separadas físicamente y en tensión. Cada hoja se recubre con una película conductora y se realizan contactos eléctricos a los bordes superior e inferior de la hoja del fondo (Y) y a los bordes izquierdo y derecho de la hoja de encima (X). Estos cuatro contactos se conectarán por medio de cables al equipo de control (mostrado en la Figura 2 y descrito con mayor detalle a continuación). La escritura sobre la superficie se detecta por la aplicación de una fuerza suficiente para hacer que los dos recubrimientos resistivos se pongan en contacto entre sí. Las hojas actúan como divisores de resistencia y se crea un gradiente de tensión mediante la aplicación de tensiones diferentes a los bordes de una hoja. La tensión en el punto de contacto se induce sobre la hoja que no está alimentada. Esta tensión es directamente proporcional a la localización del contacto sobre la hoja alimentada. Si, por ejemplo, el punto de contacto está 1/4 de camino desde el borde izquierdo del sensor y a 1/2 camino desde arriba desde el borde inferior, las tensiones que describen la localización exacta del contacto se podrían expresar en forma de coordenadas como sigue:

$$\text{la coordenada es } (1/4 \times \text{gradX}, 1/2 \times \text{gradY})$$

en la que,

35 gradX = tensión de control del borde derecho – tensión de control del borde izquierdo
gradY = tensión de control del borde superior – tensión de control del borde inferior

Con las tensiones de control derecha y superior fijadas en 5 V y las tensiones izquierda e inferior fijadas a 0 V, la coordenada se reduce a (1,25, 2,5).

Teclas de función de la pizarra blanca

40 La pizarra blanca se diseña con teclas de función o botones 20 alrededor del perímetro exterior para permitir que el usuario maneje la mayor parte de las actividades de dibujo directamente desde el tablero. La aplicación soporta la decodificación y procesamiento de estas teclas de función según sea necesario como se describe en el presente documento:

45	<p>rotulador rojo rotulador negro rotulador verde rotulador azul borrador estrecho borrador ancho instantánea/borrar todo</p>	<p>Cambiar el color actual de dibujo a rojo Cambiar el color actual de dibujo a negro Cambiar el color actual de dibujo a verde Cambiar el color actual de dibujo a azul Borrar una zona estrecha alrededor de la posición de dibujo actual Borrar una zona ancha alrededor de la posición de dibujo actual Borrar el tablero completo e insertar un marcador para guardar la imagen actual como una instantánea.</p>
50	<p>imprimir instantánea mostrar tablero</p>	<p>Imprimir una copia de la imagen actual Insertar un marcador para guardar la imagen actual como una instantánea Restaura la ventana del visor al foco de primer plano</p>

55

III. Equipo de Control

La Figura 3 de los dibujos muestra un diagrama lógico de bloques de los componentes en el equipo de control 16. El equipo de control emplea un microprocesador convencional 28 para implementar las funciones realizadas por el equipo de control por medio de un programa almacenado en la ROM 34. Un convertidor analógico a digital (A/D) 36 convierte las señales analógicas recibidas desde el sensor de la pizarra blanca 30 en información digital para su procesamiento por el microprocesador 28. Una memoria de acceso aleatorio no volátil (NOVRAM) 38 almacena información de configuración. Un módulo de control del sensor 40 responde a las señales de control desde el microprocesador 28 para proporcionar las señales analógicas apropiadas al sensor 30 y para recibir las señales eléctricas en la forma de señales analógicas desde el sensor 30. Un circuito de detección 54 informa al microprocesador 28 cada vez que se realiza un nuevo toque sobre el sensor 30, comprobando el contacto inicial y verificando que el contacto se sigue manteniendo. El equipo de control incluye también un Diodo Emisor de Luz (LED) 46 y un altavoz 48 para la generación de información visual y audible. Una interfaz RS-232 50 proporciona un enlace serie entre el equipo de control y el PC. El microcontrolador 28 es el responsable de la coordinación de toda la actividad del equipo de control. Toma los datos del sensor desde el convertidor A/D, calcula las coordenadas de contacto, filtra los datos de las coordenadas, envía los datos al mundo exterior y decodificar mensajes desde, y envía mensajes a, el mundo exterior.

Como se ha observado previamente, el equipo de control recibe información posicional en dos dimensiones por medio de cuatro cables conectados al conjunto de sensor 30 de la membrana resistiva de la pizarra blanca. Como se ve, el sensor 30 incluye una hoja de encima 31 y una hoja de debajo 32. La superficie exterior de la hoja superior 31 corresponde a la superficie de escritura 14. Los cables conectados a las hojas 31 y 32 se usan alternativamente para proporcionar energía eléctrica a una hoja y para detectar una entrada del usuario con la hoja opuesta mediante la detección de la tensión en la localización en la que se presiona por el usuario.

El equipo de control implementa un conjunto de funciones que se ejecutan tras la solicitud apropiada por el software controlador ejecutado por el PC 12. Cada una de las funciones implementadas por el equipo de control se describe individualmente a continuación.

Solicitud de tono de audio

El equipo de control proporciona advertencias audibles por medio del altavoz 48 al usuario. Esta función se selecciona por un comando del ordenador central que toma el siguiente formato:

<SOH>A Tx<CR>

En la que, SOH y CR son calificadores. SOH (Comienzo De la Cabecera) es un calificador que indica que los datos siguientes se deben interpretar como un comando. CR es un terminador para un comando. Cuando al equipo de control recibe CR sabe entonces que el comando que se está recibiendo ya ha sido recibido. X es una variable que especifica un sonido único. Los sonidos preferentemente incluyen un click, un pitido, un sonido de fanfarrias, un golpe y un zumbido. Un click es un único tono de muy corta duración. Un pitido es un único tono de aproximadamente 0,125 s de duración y de media a alta frecuencia. El sonido de fanfarrias es una serie de tres pitidos, cada uno incrementando en frecuencia respecto al anterior. Un golpe es una serie de tres pitidos, cada uno decreciente en frecuencia respecto al anterior. Un zumbido es un tono de baja frecuencia que es continuo durante el toque. Cualquiera de los sonidos precedentes se puede producir en respuesta a eventos que incluyen toque, borrado, toque de un botón y falta en el toque del botón.

Calibrado extendido

Sintaxis: <SOH>CX<CR>

Descripción: Inicia una calibración interactiva, en dos puntos de la pantalla táctil para permitir la calibración del tamaño del área activa de la pantalla. Como se usa en el presente documento, la expresión "área activa" se refiere a un área rectangular sobre la pantalla táctil en la que se puede hacer contacto mediante un dispositivo de toque para producir la transmisión de los valores digitales de posición indicativos de la posición del contacto del dispositivo de toque. El contacto de un dispositivo de toque en el exterior del área activa es ignorado por el equipo de control.

Los puntos de calibración (objetivos) se fijan en el interior desde la esquina de la imagen de video proporcionada por la pantalla táctil para una operación precisa y fácil. Durante la calibración, el área activa de la pantalla táctil se define mediante el mapeado de localizaciones a un sistema de coordenadas absolutas X, Y. Las dos áreas objetivo en la pantalla se tocan lo que envía las coordenadas X, Y para estos puntos de toque al equipo de control. El equipo de control calcula los otros puntos de toque en base a estos dos valores.

Solamente Dedo

Sintaxis: <SOH>FO<CR>

Descripción: Hace que el equipo de control acepte solamente una entrada de un dedo (es decir marcador o dedo) y es el modo por defecto cuando se usa una pizarra blanca de membrana resistiva. Si se usa una pizarra blanca que emplee una tecnología de sensor que permita el uso de un rotulador especial conectado eléctricamente al sensor de la pizarra blanca, entonces el equipo de control puede funcionar para aceptar entradas por medio de dispositivos diferentes que puedan tener diferentes interacciones eléctricas con el sensor de la pizarra blanca. Como se describe en la Patente de Estados Unidos US-A-5956020 titulada "A Touchscreen Controller with Pen and/or Finger Inputs", tal característica permite ventajosamente que el equipo de control determine qué tipo de dispositivo de entrada está siendo usado. Sin embargo, para una pizarra blanca de membrana resistiva, no hay interacción eléctrica directa entre el sensor y el dispositivo de entrada debido a que la superficie de escritura 18 aísla eléctricamente el dispositivo de entrada del sensor. La siguiente descripción es aplicable sin embargo si la pizarra blanca tiene la forma de sensor capacitivo tal como se describe en la patente mencionada anteriormente. Con tales dispositivos, el equipo de control puede ignorar la entrada del rotulador y puede ofrecer tres entradas de dispositivos táctiles:

- (1) el modo de Rotulador o Dedo detecta el contacto de un rotulador entero, dando prioridad al contacto del rotulador cuando se detectan ambos.
- (2) el modo Solamente de Dedo detecta al contacto del dedo solamente y procesa los datos de coordenadas del dedo.
- (3) el modo Solamente Rotulador detecta el contacto del rotulador solamente y procesa los datos de coordenadas del rotulador

Estos ajustes cambian de vuelta al ajuste por defecto con el inicio de la alimentación, o si se envía un comando de Restaurar por Defecto. Si se ha cambiado el ajuste por defecto en el bloque 1 de datos del equipo de control, este ajuste solamente cambia de nuevo al ajuste por defecto con un comando de Restaurar por Defecto.

Respuesta: <SOH>0<CR> Respuesta positiva

Formato en Bruto

Sintaxis: <SOH>FR<CR>

Descripción: Este comando se usa principalmente para diagnósticos. Hace que el equipo de control devuelva el nivel de la señal (cantidad de toque) de cada una de las cuatro esquinas de la pizarra blanca en formato digital. Los valores devueltos no se corrigen respecto a valores de desplazamiento y desvío. Sin embargo, los valores de desplazamiento y desvío se pueden obtener usando el comando de Obtener Bloque de Parámetros. Los datos de formato en bruto son un paquete de 7 bytes que incluyen 1 byte de estado y 6 bytes de datos binarios de las esquinas. El formato de datos para el paquete se fija ventajosamente para proporcionar la transferencia de datos más efectiva. El primer byte de cada paquete siempre tiene su bit alto (Bit 7) fijado para proporcionar una sincronización con el sistema central. El dato se devuelve en 10 bits, que se proporcionan en dos bytes.

Para usar el Formato en Bruto, el equipo de control y el PC central deben estar en un modo de comunicación de datos de 8 bits. Se debe enviar un comando de Reposición para finalizar el Formato en Bruto. El equipo de control puede devolver varios bytes de datos entre el tiempo en el que el comando de Reposición se envía y cuando equipo de control recibe el comando de Reposición. Se puede realizar un barrido continuo para el acuse de recibo de la Reposición o se puede enviar una segunda Reposición después de que hayan pasado aproximadamente 10 segundos.

Respuesta: <SOH>0<CR> Respuesta positiva

Después de que el equipo de control esté en el modo de Formato en Bruto, el equipo de control produce datos continuamente en el siguiente formato:

<paquete de 7 bits> <paquete de 7 bits> ... <paquete de 7 bits> ...

Tabla 1

Byte	Bits 0 - 7
1	b0 - b3: Nivel del controlador (cantidad de señal enviada desde el equipo de control) b4: Reservado b5: Reservado b6: Reservado b7: Bit de sincronización (Siempre 1)
2	b0 - b2: 3 bits más significativos de la esquina superior izquierda(UL) b3: Siempre 0 b4 - b6: 3 bits más significativos de la esquina inferior izquierda(LL) b7: Siempre 0
3	b0 - b2: 3 bits más significativos de la esquina inferior derecha (LR) b3: Siempre 0 b4 - b6: 3 bits más significativos de la esquina superior derecha(UR) b7: Siempre 0

(continuación)

Byte	Bits 0 - 7
4	b0 - b6: 7 bits menos significativos de la esquina inferior izquierda (LL) b7: Siempre 0
5	b0 - b6: 7 bits menos significativos de la esquina superior izquierda (UL) b7: Siempre 0
6	b0 - b6: 7 bits menos significativos de la esquina superior derecha (UR) b7: Siempre 0
7	b0 - b6: 7 bits menos significativos de la esquina inferior derecha (LR) b7: Siempre 0

Formato de Tableta

Sintaxis: <SOH>FT<CR>

5 Descripción: Este es el formato por defecto y es el único formato en el que se transmiten datos. Este comando hace que el equipo de control produzca la salida de datos de coordenadas del toque X, Y en un paquete de 5 bytes. El paquete incluye 1 byte de estado y 4 bytes de datos de las coordenadas X, Y binarias. El protocolo establece también que la salida de la coordenada X e Y como 14 bits binarios que proporcionan un intervalo de 0 a 16.383. Los bits de orden más bajo (X3-X0 e Y3-Y0) en una pantalla táctil de 1024 por 1024 no son significativos debido a que los datos pueden fluctuar con cada toque y por lo tanto no pueden ser completamente precisos. Para usar el Formato de Tableta, el equipo de control y el sistema central deben estar en un modo de comunicación de datos de 8 bits.

Respuesta: <SOH><CR> Respuesta positiva

15 Después de que el equipo de control esté en el modo de Formato de Tableta, el toque de la pantalla hace que el equipo de control devuelva una respuesta en el formato siguiente:

SxxYy

S = byte de estado; primer byte de datos. Refiérase a la Tabla 2 a continuación que define los bits de estado (Byte 1) para los datos en Formato de Tableta.

Xx = datos de la coordenada X (horizontal); segundo y tercer byte de datos.

20 Yy = datos de la coordenada Y (vertical; cuarto y quinto byte de datos)

	MSB*		Bits			LSB*		
Secuencia de datos	7	6	5	4	3	2	1	0
S - Byte 1	1	S6	S5	S4	S3	S2	S1	S0
X - Byte 2	0	X6	X5	X4	X3	X2	X1	X0
x - Byte 3	0	X13	X12	X11	X10	X9	X8	X7
Y - Byte 4	0	Y6	Y5	Y4	Y3	Y2	Y1	Y0
y - Byte 5	0	Y13	Y12	Y11	Y10	Y9	Y8	Y7
* MSB = Bit Más Significativo, LSB = Bit Menos Significativo								

Tabla 2. Bis de estado del Formato de tableta

Bit	Descripción	Valores
S0	Estado conmutador 1	No usado
S1	Estado conmutador 2	0 = Conmutador está desconectado.
S2 - S4	Reservado	-----
S5	Dedo o rotulador	No usado

(continuación)

Bit	Descripción	Valores
S6	Proximidad	1 = Se está tocando la pizarra blanca (un toque o un toque continuado). 0 = La pizarra blanca no está siendo tocada (se ha levantado un toque o inactiva). Cuando el bit en la proximidad cambia de 1 a 0 (No toque alzado), se obtiene un conjunto final de datos de coordenadas X, Y con el byte de proximidad igual a 0 y los datos de las coordenadas X, Y iguales al último punto de toque.
S7	Sincronización del paquete	Siempre 1.

Obtener Bloque de Parámetros

5 Descripción: Permite el acceso a todos los parámetros iniciales o de ejecución usados por el equipo de control. El comando de Obtener Bloque de Parámetros (GP) trabaja en conjunto con el comando de Establecer Bloque de Parámetros (SP). Este par de comandos se usan para finalidades de configuración y diagnóstico. Los parámetros se recuperan usando el comando GP y se modifican en bloques de datos usando del comando SP. Los bloques incluyen los datos de calibración e inicialización, datos de linealización y variables en tiempo de ejecución.

Modo de Flujo

10 Sintaxis: <SOH>MS<CR>

Descripción: Hace que el equipo de control envíe un flujo continuo de datos de coordenadas (X, Y) cuando se toca la superficie de la pizarra blanca. El equipo de control continúa enviando datos siempre que se siga tocando la pantalla. El equipo de control envía los datos incluso si el toque es fijo y sin cambios. El formato de los datos de la coordenada depende del último comando de formato recibido por el equipo de control.

15 Respuesta: <SOH>0<CR> Respuesta positiva

Comando Nulo

Sintaxis: <SOH>Z<CR>

20 Descripción: Consulta al equipo de control y espera una respuesta. Este comando se usa para determinar si la comunicación con el equipo de control se ha establecido. El uso de este comando no afecta a los parámetros de operación actuales del equipo de control.

Respuesta: <SOH>0<CR> Respuesta positiva

Producir Salida de Identidad

Sintaxis: <SOH>Z<CR>

25 Descripción: Hace que el equipo de control notifique una cadena de identidad del firmware, que incluye el tipo de equipo de control y el número de versión de firmware.

Respuesta: <SOH>CcXxxx<CR>

en la que:

Cc = dos caracteres ASCII que describen el tipo de equipo de control. Q1 = equipo de control Serie/SMT3 que está disponible en Micro Touch Systems, Inc., Methuen, Massachusetts.

30 Xxxx = cuatro caracteres ASCII que indican el número de versión del firmware en formato decimal. Los dos primeros caracteres representan el número de versión; los dos últimos caracteres representan el nivel de la revisión. Por ejemplo, 0100 significa la Versión 1, Revisión 0 (esto es 1.0) o 0510 significa Versión 5, Revisión 1 (5.1).

Reposición

35 Sintaxis: <SOH>R<CR>

Descripción: Inicializa el hardware y el firmware, hace que el equipo de control detenga al envío de datos y recalcula las condiciones del entorno (por ejemplo, valores de desvío y desplazamiento). El comando de Reposición también cancela los comandos de Formato en Bruto y Calibrar en Bruto y devuelve el equipo de control táctil a la operación

normal. El ordenador debería enviar preferentemente un comando de Reposición siempre que se alimente y se intente establecer comunicación con el equipo de control táctil. El comando de Reposición puede llevar hasta 240 milisegundos (0,25 segundos) para ejecución. Por lo tanto, el programa de aplicación debería esperar al menos 250 milisegundos (y recibir la respuesta al comando) antes de enviar otro comando al equipo de control táctil a continuación de una reposición.

5

Respuesta: <SOH>0<CR> Respuesta positiva

Restaurar por Defecto

Sintaxis: <SOH>RD<CR>

Descripción: Hace que el equipo de control táctil asuma los parámetros de operación por defecto de fábrica. El comando de Restaurar por Defecto copia los parámetros por defecto de fábrica desde la ROM a la memoria no volátil (NOVRAM) y a continuación ejecuta un comando de Reposición. La Tabla 3 lista los valores por defecto de fábrica. El comando de Restaurar por Defecto es útil en situaciones en las que unos comandos inadvertidos al equipo de control han convertido a la pizarra blanca en inoperativa.

10

Tabla 3. Valores por defecto de fábrica

Parámetro	
Formato de datos	Formato de Tableta
Modo operativo	Modo de Flujo
Ajuste serie	N, 8, 1
Tasa de bits	9600
Tasa de bits automática	N/D
Modo de dedos	Modo de dedo
Vuelta a calibración de fábrica	Sí

15

El comando de Restaurar por defecto Requiere aproximadamente 75 a 100 milisegundos, más el tiempo de ejecución del comando de Reposición. En consecuencia, el programa de aplicación central debería esperar un mínimo de 350 milisegundos (y recibir la respuesta al comando) antes de enviar otro comando al equipo de control táctil.

20

Respuesta: <SOH>0<CR> Respuesta positiva

Establecer Bloque de Parámetros

Descripción: Permite el acceso a todos los parámetros de inicio y tiempo de ejecución usados por el equipo de control. El comando de Establecer Bloque de Parámetros (SP) trabaja en conjunto con el comando de Obtener Bloque de Parámetros (GP). Este par de comandos se usan para finalidades de configuración y diagnóstico. Los parámetros se recuperan (usando GP) y se modifican (usando SP) en bloques de datos. Los bloques incluyen los datos de calibración e inicialización, datos de linealización y variables en tiempo de ejecución.

25

Tipo de Unidad

Sintaxis: <SOH>UT<CR>

Descripción: Hace que el equipo de control notifique una cadena de identidad del equipo de control. Esta cadena identifica el tipo de equipo de control actualmente adjunto al sistema, lista las características soportadas por el equipo de control y produce la salida del estado del hardware del equipo de control táctil.

30

Respuesta: Devuelve el código de identificación de hasta 8 caracteres ASCII en el siguiente formato:

<SOH>TTFFFsS<CR>

en la que:

35

TT = Dos caracteres ASCII que indican el tipo de equipo de control, que preferentemente indica un equipo de control Serie/SMT3 disponible en MicroTouch Systems, Inc.

QM = Equipo de control Serie/SMT3

Ffff = Cuatro caracteres ASCII que indican las características soportadas por el equipo de control. Unos asteriscos (****) indican que no se configuran características adicionales.

Ss = Dos caracteres ASCII que proporcionan información de estado sobre el hardware del equipo de control táctil. Los dos caracteres representan 1 byte. Cada carácter está en el intervalo de 0 a 9 y de A a F.

5 La Tabla 4 define el significado de cada bit en el byte de estado. Cada tipo se puede fijar a 1 ó 0, en los que:

- 1 = Error
- 0 = No error
- 00 = No errores de diagnóstico (respuesta normal)

Tabla 4. Definición de bits para el comando del Tipo de Unidad

Bit	Estado Serie/SMT#	Estado del rotulador táctil
0	Reservado	Error de RAM. Mal funcionamiento del hardware.
1	Error de la ROM. Error de verificación de la suma de comprobación del firmware.	Igual.
2	Error PWM. Incapaz de establecer un intervalo de operación PWM en el inicio. Error no recuperable.	Error Analógico a Digital (A/D). El convertidor A/D funciona incorrectamente.
3	Error de la NOVRAM. Los parámetros operativos en el equipo de control de la NOVRAM no son válidos. Usando valores por defecto.	Igual.
4	Error HDW. El hardware del equipo de control ha fallado (incapaz de inicializar o configurar la matriz de puertas). Error no recuperable.	Error del ASIC. El Circuito Integrado de Aplicación Específica (ASIC) ha fallado.
5	Reservado.	Reponer marcador. 1= No se ha enviado un comando de Tipo de Unidad desde la última reposición. 0 = Se ha enviado un comando de Tipo de Unidad desde la última reposición.
6	Error de cable. Los datos de linealización en el cable de la NOVRAM no son válidos.	Reservado.
7	Error NOVRAM2. Los datos de linealización en el equipo de control de la NOVRAM no son válidos.	Igual.

IV. Software del ordenador

10 La Figura 4 de los dibujos muestra un diagrama de bloques de la interacción entre el software de aplicación, mostrado en 64 y el controlador 62. El controlador proporciona el soporte de comunicaciones necesario para una interfaz con la pizarra blanca a través del puerto serie 70 de la capa de hardware del PC y una interfaz API apropiada en la capa de aplicación del PC como se describe con más detalle a continuación. El software de aplicación proporciona toda la funcionalidad requerida para una Interfaz de Usuario Gráfica (GUI) y el sistema de archivos. En una realización preferida, el software de aplicación funciona bajo el sistema operativo Windows95 o Windows 3.1 en conjunto con las extensiones Win32S, cada una de las cuales está disponible en Microsoft Corporation, Redmond, Washington. Varios aspectos del desarrollo de software para operar bajo los sistemas operativos precedentes y para utilizar los servicios proporcionados por tales sistemas se describen en la literatura disponible en Microsoft Corporation. A modo de ejemplo, el entorno de desarrollo Visual C disponible en Microsoft Corporation se puede emplear para desarrollar sistemas que empleen las características descritas en el presente documento. En la descripción a continuación, se utilizan numerosos acrónimos en conexión con los tipos de archivo, servicios o protocolos empleados en los sistemas operativos precedentes. Los expertos en la materia naturalmente comprenderán el significado de tales términos en el contexto de los sistemas operativos Windows precedentes.

Tanto el controlador como la aplicación se pueden componer que varios componentes individuales incluyendo archivos DLL, archivos EXE, archivos INI o similares. El software de instalación proporciona toda la funcionalidad

necesaria para instalar el controlador y la aplicación en el sistema del usuario incluyendo la detección de una pizarra blanca conectada y la configuración de los ajustes del puerto de comunicación.

El controlador toma la forma de un controlador de dispositivo virtual (VxD) que recibe datos de las comunicaciones serie desde la pizarra blanca y pasa los mensajes reunidos resultantes hacia la aplicación. El controlador se carga en el arranque de Windows con el resto del sistema operativo. En este punto, el controlador está en la memoria, sin embargo no se comunica con el puerto serie o con la aplicación. La aplicación establece la comunicación con el controlador a través de una API en modo protegido. La aplicación puede entonces enviar los comandos del controlador para su ejecución. La aplicación comienza pidiendo al controlador la apertura del puerto serie conectado a la pizarra blanca. El controlador abre el puerto serie usando un VCOMM en Windows 3.1 (o Windows 3.11) y Windows95. En Windows 3.1 ó 3.11, el controlador no abre el puerto serie. El controlador ajusta a continuación el puerto serie para que comunique con la pizarra blanca mediante el cambio de la tasa de bits, etc. e inhabilitando las interrupciones del puerto serie. A continuación el controlador instala un gestor de interrupciones para el Reloj en Tiempo Real (IRQ 8).

El reloj en tiempo real, o RTC, genera una interrupción aproximadamente 1.000 veces por segundo por defecto. El RTC se puede ajustar por medio de la CMOS del ordenador para que se ejecute a velocidades diferentes. El controlador modifica la CMOS de modo que la interrupción RTC queda activada a una velocidad de aproximadamente 2.000 veces por segundo. Usando esta interrupción como un latido, el controlador consulta al puerto serie para leer datos enviados desde la pizarra blanca.

Los datos de la pizarra blanca se manejan de una variedad de formas, dependiendo de cómo la aplicación ha instruido al controlador para procesarlos. Hay varios comandos que determinan cómo serán procesados los datos. Por ejemplo, si la aplicación envía el comando "comenzar registro", los datos serán devueltos a la aplicación por medio de un procedimiento de retrollamada. La dirección del procedimiento de retrollamada se pasa al controlador cuando la aplicación envía el comando "comenzar registro". La aplicación podría indicar también al controlador el uso de los datos que llegan desde la pizarra blanca como una entrada para el ratón mediante el envío de un comando "establecer modo ratón".

Ventajosamente, el controlador emplea la consulta en lugar de las interrupciones para determinar si los datos están disponibles en el puerto serie. La norma actual para PC compatibles con IBM es compartir las IRQ 3 e IRQ 4 entre los puertos COMM 1 a 4. Los puertos COMM 1 y 3 comparten típicamente la IRQ 4 y los puertos COMM 2 y 4 comparten típicamente la IRQ 3. Las IRQ se usan para señalar a un puerto serie que hay datos entrantes desde un dispositivo externo. El problema con esto es que la compartición de las IRQ funciona solamente si todo el sistema acuerda compartirlas. Si una aplicación desea tomar el control de la IRQ 3 y no compartirla, el sistema lo permitirá, y no será posible compartir esa IRQ. Adicionalmente, no solamente las aplicaciones pueden compartir las IRQ de modo no amigable, sino que las placas madre, las tarjetas COMM, módems y cualquier otro dispositivo serie puede establecer una IRQ como no compartible. Estos problemas no se pueden resolver con el software. Por lo tanto, la capacidad de un aplicación de comunicación serie para funcionar apropiadamente estará altamente ligada al computador en el que está instalada. Incluso si la máquina en la que la aplicación se instala está compartiendo amigablemente las IRQ, hay otros problemas que complican esta cuestión. Por ejemplo, las IRQ pueden no señalar dos puertos a la vez. Si el puerto COMM 2 está siendo usado a la vez que se está usando el puerto COMM 4, y los puertos comparten la IRQ 3, los puertos obtendrán notificación a una velocidad mucho más lenta. Esto puede conducir a pérdida de datos desde la UART y producir otros problemas de tiempos que podrían interferir con ambas operaciones de comunicación.

El método de consulta es ventajoso debido a que el controlador no depende de una interrupción. Mediante el no uso de la interrupción como el procedimiento para determinar si están disponibles los datos, e inhabilitando las interrupciones en el puerto serie que se está usando por la pizarra blanca, el controlador evita todos los problemas con la compartición de las IRQ. El controlador puede trabajar en cualquier PC independientemente de otros dispositivos fijados al sistema. Otro beneficio es el de no interferir con otros dispositivos serie. Naturalmente, el PC aún tiene que tener al menos un puerto serie disponible para la pizarra blanca.

El controlador también soporta una API apropiada para recibir mensajes (como se describe a continuación) desde la aplicación y transmitir los datos en forma serie de vuelta a la pizarra blanca. El controlador soporta compartir las interrupciones hasta el grado en que otro controlador de dispositivo serie tal como un controlador de ratón o de un módem pueda recibir también datos de comunicaciones usando la misma interrupción asignada.

A intervalos predeterminados, que en una realización preferida es aproximadamente de 500 microsegundos, el controlador 62 consulta al puerto serie para ver si hay información en el puerto serie del PC que está conectado a la pizarra blanca y realiza las funciones mostradas en la Figura 6. Como se ve en la Figura 6, en 80, el controlador recibe eventos de la pizarra blanca en la forma de entradas del usuario gráficas introducidas en la zona de escritura 18 y eventos especiales introducidos a través de la zona de botones 20 por medio del puerto serie 81. En 82, el controlador comprueba el paquete recibido para determinar la presencia de errores en los cinco bytes del paquete. Como se ha observado previamente en conexión con la descripción del comando de Formato de Tableta, el paquete de cinco bytes se codifica de modo que el primer bit del primer byte se establece siempre a un valor uno y el primer bit de los cuatro bytes restantes se establece en un valor cero. Si se recibe un byte en el que el primer bit está fijado

a uno, los siguientes cuatro bytes recibidos deberían cada uno tener el primer bit fijado en cero para que el paquete sea considerado un paquete válido. Si se recibe un byte en el que el primer bit está fijado a uno y los siguientes cuatro bytes no comienzan con un cero, entonces aquél con el primer bit fijado a uno y los bytes posteriores son ignorados (etapa 83) hasta el siguiente byte recibido que tenga el primer bit fijado a uno. Por ello, el procedimiento anterior reduce ventajosamente la posibilidad de recibir información falsa por parte del controlador en el caso de ruido y que provoque que se pierdan datos a través del enlace serie que conecta la pizarra blanca al ordenador. Si se ignora la información debido a la detección de un error, el controlador inserta un comando de levantar rotulador en la etapa 84 para reducir la posibilidad de trazados producidos por la posible pérdida de un paquete que contenga un comando de rotulador arriba. Si el byte perdido no contiene un comando de levantar rotulador entonces la etapa 84 hará que aparezca una pequeña discontinuidad en la imagen almacenada por el controlador. Sin embargo, se cree que éste es un compromiso aceptable respecto a la alternativa de perder un comando de levantar rotulador que podría hacer que la imagen almacenada en el PC tenga una línea que conecta el punto en el que tuvo lugar la elevación y no fue registrada hasta el punto en el que se registró el siguiente paquete válido.

En 85, se comprueban los eventos de la PB para determinar si hay un evento especial y por lo tanto requiere una acción inmediata y, si es así, el controlador en 88 genera una retollamada a la aplicación 64 para hacer que la acción se realice. Preferiblemente todos los eventos introducidos en la zona de control de la pizarra blanca son eventos especiales. En caso contrario, en 87, el controlador almacena el evento en un tipo de memoria intermedia de anillo (véase en 68 en la Fig. 3) que se almacena en una memoria compartida. En una realización preferida, el controlador simplemente distingue entre los datos generados en la pizarra blanca en un grado suficiente para determinar si los datos se introdujeron en la zona de escritura o si los datos se introdujeron en la zona de control y, si es así, si los datos introducidos en la zona de control requieren una acción inmediata. El reconocimiento real de una acción en el evento se realiza por la aplicación.

Periódicamente, si la memoria intermedia de anillo contiene una cantidad determinada de datos, el controlador generará una retollamada a la aplicación para recuperar la información almacenada en la memoria intermedia de anillo 68 para visualización y almacenamiento en un archivo de sesión. Los eventos especiales que requieren una acción inmediata son el evento de Impresión y el evento de Mostrar Tablero. La aplicación y el controlador se comunican entre sí pasándose mensajes por medio de colas de mensaje (véanse en la Figura 4 en 65 y en 63) asociados respectivamente con la aplicación y el controlador.

La aplicación de pizarra blanca interactúa con el equipo de control por medio de una librería estándar de rutinas de interfaz implementadas por el controlador que proporcionan un conjunto de comandos a ser usados por el controlador para hacer uso de las capacidades de entrada de la pizarra blanca 10. De acuerdo con la invención, se pueden escribir por lo tanto una amplia variedad de programas de aplicación además del programa de aplicación descrito en el presente documento para hacer un uso efectivo de la pantalla táctil usando un conjunto estándar de llamadas a una función de la Interfaz del Programa de Aplicación (API) fácilmente entendidas que pueden estar convenientemente expresadas en el mismo lenguaje usado por el programador de la aplicación para una programación convencional. La forma en que se pueden realizar las llamadas a la función API así como las funciones realizadas y los valores devueltos para cada llamada a una función se explican en detalle a continuación:

Comunicación de Aplicación a Controlador

La aplicación envía solicitudes al controlador por medio de una rutina conocida como rutina de API en Modo Protegido. La aplicación debe ejecutarse en un entorno de 16 bits para acceder a la rutina. Esto significa que una aplicación de 32 bits necesitará un truncado a una DLL de 16 bits para comunicar con el VxD.

El código siguiente se puede usar para recuperar la dirección de la rutina de la API en Modo Protegido del controlador. Este código debe estar en un segmento de código de 16 bits. La siguiente nota se aplica al código contenido en el presente documento: Copyright 1966, Micro Touch Systems, Inc.

```

45 static BOOL GetVxDAPIHandler(void)
    {
        char      *szError = NULL;
        WORD      wError = 0;
        /* obtener la entrada de la API a Ibid - si está instalada */
50  _asm
    {
        mov     ax, 1600h           ; ¿modo mejorado?
        int    2Fh                 ; llamada a la api
        test   al, 7Fh             ; ¿modo mejorado en ejecución?
55  jz      not_running_enhanced   ; no
        mov     ax, 1684h          ; obtener llamada a la API del dispositivo
        mov     bx, MTWBS_DEV_ID   ; para el controlador
        int    2Fh                 ; obtener el punto de entrada a la api
        mov     word ptr MTWBS_API, di ; guardar la dirección de la llamada
60  mov     word ptr MTWBS_API + 2, es
        mov     ax, es             ; ¿está instalada la aplicación?
        or     ax, ax
    }

```

ES 2 394 015 T3

```

    jnz    vxd_installed

    mov    ax, di
    or     ax, ax
    jz     vxd_not_installed      ; si no, dividir

5     mov    wError, 3            ; si solamente es es 0 entonces error de memoria
    jmp    get_out

vxd_installed:
    mov    wError, 0            ; mostrar el éxito (PHERR_NOERROR)
    jmp    get_out

10    not_running_enhanced:
    mov    wError, 1            ; ;no windows enh! (PHERR_NO386ENH)
    jmp    get_out              ; devolver nuestro código de error

vxd_not_installed:
    mov    wError, 2            ; ;Ibid? (PHERR_NOVPOSTD)

15    get_out:
    }
    /* comprobación de error y notificación */
    if ( wError == 1 )
        szError = ";EJECUTANDO WINDOWS MEJORADO!";
20    else if ( wError == 2 )
        szError = ";INSTALE PRIMERO Ibid.VxD!";
    else if ( wError == 3 )
        szError = ";ERROR DE MEMORIA! no se pudo cargar Ibid.VxD"
25    if (wError)
    {
        MessageBox( NULL, szError, "VxD TEST.EXE", MB_OK );
        return ( FALSE );
    }
    return ( TRUE );
30 }

```

Llamando al controlador

El controlador proporciona una interfaz similar a SendDriverMessage para especificar un comando del controlador y los parámetros para el comando. La API para esta función es:

```
static LRESULT CallVxDAPIService(WORD Api_Selector, LPARAM lParam1, LPARAM lParam2);
```

35 Esta rutina pasa los datos por medio de registros. Se usa el siguiente código para llamar al controlador desde la aplicación.

```
static LRESULT CallVxDAPIService(WORD Api_Selector, LPARAM lParam1,
LPARAM lParam2)
{
40     LRESULT      dwReturn = 0;          /* se supone lo peor          */
    /* comprobación sanitaria */
    if ( MTWBS_API )
    {
45         _asm
        {
            push    ax
            push    bx
            push    cx
            push    dx
50         push    di

            mov    ax, Api_Selector      ; función a llamar
            mov    box, word ptr lParam1  ; cargar lParam1 en los registros
            mov    cx, word ptr lParam1 + 2

            mov    dx, word ptr lParam2   ; cargar lParam1 en los registros
            mov    di, word ptr lParam2 + 2

55         call    dword ptr [MTWBS_API]  ; llamar el servicio de la rutina API VxD

            mov    word ptr dwReturn, ax  ; rellenar el valor de devolución
            mov    word ptr dwReturn + 2, dx ; rellenar el valor de devolución

60         pop     di
            pop     dx

```

```

        pop    cx
        pop    bx
        pop    ax
    }
5      }
      /* devolver el resultado */
      return ( dwReturn );
} /* */

```

Las API en Modo Protegido del Controlador

10 Los comandos a los que el controlador responde se listan a continuación con una descripción de cada comando, los parámetros del comando y los valores devueltos esperados:

MTWBCMD_NOP

Este comando es la ID usada en el comando **MTWBCMD_GetVxdVersion** para obtener el número de versión del controlador sin una especificación de etiqueta de comando y no se debería pasar al controlador

15 **MTWBCMD_GetVersion**

Obtiene el número de versión del firmware (que es el código ejecutado por el equipo de control).

Este comando sólo se puede ejecutar después de que se haya abierto el puerto con o bien el comando **MTWBCMD_OpenCommPort** o bien el comando **MTWBCMD_OpenCommPortEx**.

20 IPParam1 ignorado
IPParam2 ignorado

Devuelve:

El número versión del controlador en caso de éxito

0 si el número de versión no se ha devuelto desde el firmware. El controlador notificará la aplicación cuando los datos han recibido por medio de la Retrollamada de Mensaje.
25 1 en caso de fallo.

30 **NOTA:** Los comandos **MTWBCMD_GetVersion**, **MTWBCMD_GetWhiteBoardInfo**, **MTWBCMD_GetTimeout** y **MTWBCMD_SetTimeout** se envían al firmware cuando el controlador recibe o bien el comando **MTWBCMD_OpenCommPort** o bien el comando **MTWBCMD_OpenCommPortEx**. Debido a que estos comandos requieren una realimentación inmediata desde el controlador, el controlador debe intentar obtener los datos tan pronto como sea posible. Si es posible para cualquiera de estos comandos, se enviarán antes de que el controlador haya recibido los datos desde la pizarra blanca debido al retraso en la recepción de datos desde la pizarra blanca. En este caso, el controlador devolverá un 0 para indicar a la aplicación que debería esperar a una notificación desde el controlador de que los datos están listos. Una vez que la aplicación recibe la notificación, puede llamar al controlador para obtener la información. Si el controlador detecta que la pizarra blanca está desconectada del puerto serie, entonces estos comandos devolverán -1.
35

MTWBCMD_SetLoggingMode

Este comando dice al controlador que comience el registro de datos recibidos desde la pantalla blanca adjunta al sensor identificado por IPParam1 llamando a la retrollamada proporcionada en IPParam2. El puerto serie no necesita estar abierto para usar este comando, pero los datos no serán registrados hasta que el puerto esté abierto.

40 IPParam1 contiene el número de sensor, 0 a 4 (en el que 0 se interpreta como 1)
IPParam2 contiene un puntero lejano a la rutina de retrollamada en la aplicación

El prototipo de función de retrollamada es:

```
long PASCAL WhiteboardCallbackEx (long logHandle, TouchPacket FAR *lpPacket);
```

El parámetro TouchPacket FAR* apunta a la siguiente estructura de:

```

45 struct TouchPacket
    {
        WORD    wTabletX;        // localización X
        WORD    wTabletY;        // localización Y
        WORD    wState;          // cabecera del paquete de Tablet
50        WORD    wRepCount;      // repetir contador = paquetes descargados + 1
    };

```

La retrollamada se debe localizar en un segmento de código bloqueado. No puede realizar ninguna llamada a las API de Windows excepto para la rutina PostMessage. No pueda acceder a ningún dato salvo a los bloqueados.

5 La primera vez que el controlador llama a la retrollamada, pasa NULL para el logHandle. Esto dice a la aplicación que debería retornar el logHandle que identifica a qué pizarra blanca pertenecen los paquetes. No se realiza ningún procesamiento del paquete esta primera vez. Todas las demás veces que el controlador llama a la retrollamada, se pasará el logHandle para identificar de qué tablero viene el paquete y el valor de retorno será ignorado.

Devuelve:

número de sensor - en caso de éxito
0 - en caso de fallo

MTWBCMD_SetMouseMode

10 Este comando se usa para o bien consultar al controlador para que retorne el estado del modo ratón o para cambiar el modo ratón a activo o a inactivo. Cuando el modo ratón está activo, el controlador usa los paquetes táctiles que se están enviando desde la pizarra blanca adjunta al número de sensor pasado a través de IParam1 como entrada para el ratón.

IParam1 el número de sensor del que tomar la entrada, 0 a 4 (en el que 0 se interpreta como 1)
IParam2 0 = detener el modo ratón, 1 = iniciar el modo ratón, 2 = obtener el modo ratón

15 Retornos:

Para IParam2 = 1 o IParam2 = 0

VERDADERO - en caso de éxito
FALSO - en caso de fallo

Para IParam2 = 2

1 - para el modo ratón ACTIVO
0 - para el modo ratón INACTIVO

MTWBCMD_StopLoggingMode

20 Este comando indica al controlador que detenga el registro de datos enviados por la pizarra blanca identificada en IParam1.

IParam1 Manipulador de la Pizarra Blanca (devuelto desde **MTWBCMD_SetLoggingMode**)
IParam2 ignorado

Devuelve:

VERDADERO - en caso de éxito
FALSO - en caso de fallo

MTWBCMD_GetWhiteBoardInfo

25 Este comando se puede usar para obtener varias piezas de información acerca del ajuste del controlador y de la o las pizarras blancas. Este comando sólo se puede ejecutar después de que se haya abierto el puerto con o bien el comando **MTWBCMD_OpenCommPort** o bien el comando **MTWBCMD_OpenCommPortEx**.

30 IParam1 la palabra baja del IParam1 especifica el tipo de información requerida, la palabra alta depende de la información solicitada.
IParam2 depende del tipo de información solicitada.

Este comando permite el usuario solicitar los siguientes datos pasando la ID como la palabra baja del IParam1.

ID	dato a ser devuelto
1	el tamaño de la estructura e MTWB_BoardInfo
2	el número de sensores fijados al equipo de control. El controlador obtiene este número del archivo <i>system.ini</i> bajo la entrada "NumberOfBoards".
3	la posición relativa del sensor 2. El controlador obtiene este número del archivo <i>system.ini</i> bajo la entrada "LocationBoard2".
4	la estructura MTWB_BoardInfo para la pizarra blanca adjunta al sensor 1.
5	la estructura MTWB_BoardInfo para la pizarra blanca adjunta al sensor 2.

Estas lds se representan por el siguiente enum:

```
enum MTWBS_INFO {
    MTWBINFO_BoardinfoSize = 1      ,
    MTWBINFO_MultiboaidGetnum      ,
5   MTWBINFO_MultiboardPos        ,
    MTWBINFO_BoardinfoGet1        ,
    MTWBINFO_BoardinfoGet2        ,
} ; MTWBINFO_Last
```

10 Cuando se pasa el comando al controlador con la palabra baja del IParam1 igual a 4 ó 5, se usan las palabras altas del IParam1 y del IParama2. En ambos casos, la palabra alta del IParam1 necesita contener el tamaño de la memoria intermedia referenciada por IParam2. IParam2 contiene un puntero al comienzo de la estructura de datos MTWB_BoardInfo. La palabra alta del IParam1 contiene el tamaño de la memoria intermedia referenciada por el IParam2. La estructura del MTWB_BoardInfo es:

```
15 struct tagBoardInfo {
    WORD m_Size; // la idea de los controladores sobre el tamaño de esta
    estructura
    DWORD m_iLeftDrawStart;
    DWORD m_iRightDrawStart;
    DWORD m_iTopDrawStart;
20  DWORD m_iBottomDrawStart;
    DWORD m_iBoardWidth;
    DWORD m_iBoardHeight;
    DWORD m_ixResolution;
    DWORD m_iyResolution;
25  DWORD m_iTBD1; // futuros
    DWORD m_iTBD2;
};
typedef struct tagBoardInfo MTWB_BoardInfo;
```

Devuelve:

30 Cuando la palabra baja de IParam1 = 1
El tamaño de la estructura MTWB_BoardInfo

Cuando la palabra baja de IParam1 = 2
El número de tableros adjuntos al firmware (por defecto es 1)

35 Cuando la palabra baja de IParam1 = 3
La posición relativa del sensor 2 que puede ser uno de los siguientes valores:

```
enum MTWBS_POSITION {
    MTWBPOS_Error      , // Error
    MTWBPOS_Board2IsBelow  , // Tablero 2 debajo del tablero 1
40  MTWBPOS_Board2IsLeft   , // Tablero 2 a la izquierda del tablero 1
    MTWBPOS_Board2IsAbove , // Tablero 2 por encima del tablero 1
    MTWBPOS_Board2IsRight , // Tablero 2 a la derecha del tablero 1
    MTWBPOS_Last       , // El primer valor ilegal
} ;
```

El valor por defecto es MTWBPOS_Error.

45 Cuando la palabra baja de IParam1 = 4 o 5

El tamaño de la estructura MTWB_BoardInfo en caso de éxito

0 si la información de la pizarra blanca no se ha devuelto desde el firmware. El controlador notificará a la aplicación cuando los datos se han recibido a través de la Retrollamada de Mensaje.

-1 si los parámetros son incorrectos o si la pizarra blanca está desconectada.

50 NOTA: Los comandos **MTWBCMD_GetVersion**, **MTWBCMD_GetWhiteBoardInfo**, **MTWBCMD_GetTimeout** y **MTWBCMD_SetTimeout** se envían al firmware cuando el controlador recibe o bien el comando **MTWBCMD_OpenCommPort** o bien el comando **MTWBCMD_OpenCommPortEx**. Debido a que estos comandos requieren una realimentación inmediata desde el controlador, el controlador debe intentar obtener los datos tan pronto como sea posible. Si es posible para cualquiera de estos comandos, se enviarán antes de que el controlador haya recibido los datos desde la pizarra blanca debido al retraso en la recepción de datos desde la pizarra blanca. En este caso, el controlador devolverá un 0 para indicar a la aplicación que debería esperar a una notificación desde el controlador de que los datos están listos. Una vez que la aplicación recibe la notificación, puede llamar al controlador para obtener la información. Si el controlador detecta que la pizarra blanca está desconectada del puerto serie, entonces estos comandos devolverán -1.

MTWBCMD_SetSound

Este comando se usa para decir al controlador que envíe un comando de sonido al firmware. Este comando sólo se puede ejecutar después de que se haya abierto el puerto o bien con el comando **MTWBCMD_OpenCommPort** o bien con el comando **MTWBCMD_OpenCommPortEx**.

- 5 IParam1 la ID del comando de sonido a enviar al firmware.
IParam2 ignorado.

La ID a pasar como el LParam1 viene del siguiente enum:

```
10 enum MTWBS_SOUND {
    MTWBSOUND_DoNothing,           // no enviar un comando de sonido al equipo de control = 0
    MTWBSOUND_LoudClick,
    MTWBSOUND_LoudHum,
    MTWBSOUND_LoudBeep,
    MTWBSOUND_LoudFlop,
    15 MTWBSOUND_LoudFanFare,
    MTWBSOUND_SoftClic,
    MTWBSOUND_SoftHum,
    MTWBSOUND_SoftBeep,
    MTWBSOUND_SoftFlop,
    20 MTWBSOUND_SoftFanFare,
    MTWBSOUND_Quiet,           // no hacer sonido cuando se toque = B
    MTWBSOUND_Last           // El primer valor ilegal
};
```

Devuelve:

VERDADERO - en caso de éxito
FALSO - en caso de fallo

- 25 **NOTA:** un éxito cuando se envía un comando al firmware de la pizarra blanca sólo significa que el controlador fue capaz de enviar el comando. No significa que el firmware haya recibido el comando ni que actúe con él apropiadamente.

MTWBCMD_SetEraseMode

- 30 Este comando se usa para indicar al controlador que o bien active o bien desactive el marcador de modo borrador en el comando sonido. El firmware usa in bit en la cabecera del comando de sonido para determinar si el tablero está en el "modo borrador".

IParam1 contiene el número de sensor, 0 a 4 (en el que 0 se interpreta como 1)
IParam2 1 = fijar modo borrador, 0 = detener modo borrador

Devuelve:

VERDADERO - en caso de éxito
FALSO - en caso de fallo

- 35 **NOTA:** un éxito cuando se envía un comando al firmware de la pizarra blanca sólo significa que el controlador fue capaz de enviar el comando. No significa que el firmware haya recibido el comando ni que actúe con él apropiadamente.

MTWBCMD_SetTimeout

- 40 La aplicación puede usar este comando para escribir un nuevo valor de tiempo de expiración en el Bloque 1 del firmware de la pizarra blanca. Este comando se debería usar con precaución ya que sobrescribe los datos del Bloque 1. Este comando sólo se puede ejecutar después de que el puerto se haya abierto o bien con el comando **MTWBCMD_OpenCommPort** o bien con el comando **MTWBCMD_OpenCommPortEx**.

- IParam1 ignorado
45 IParam2 la palabra baja de IParam2 contiene el tiempo de expiración del firmware, la palabra alta del IParam2 contiene el intervalo de confirmación del controlador.

- La palabra baja del IParam2 contiene el tiempo de expiración del firmware en segundos. El valor debe estar entre 0 y 255. La palabra alta del IParam2 es el intervalo, en segundos, que el controlador espera entre solicitudes de confirmación al equipo de control. El valor del tiempo de expiración del equipo de control se escribirá en la EEPROM del equipo de control haciéndole persistente. El intervalo de confirmación del controlador se repone al valor por defecto, 60 segundos, o al valor fijado por la entrada "VerifyTime" en el archivo *system.ini* cada vez que se carga el controlador. Una vez que se envía o bien el comando **MTWBCMD_OpenCommPort** o bien el comando

MTWBCMD_OpenCommPortEx, el controlador intenta leer el Bloque de datos 1 desde el firmware, si la lectura del bloque tiene éxito, entonces se fija el intervalo de confirmación en el valor VerifyTime contenido en la estructura de datos del Bloque 1.

Devuelve:

VERDADERO - en caso de éxito
FALSO - en caso de fallo

- 5 NOTA: Los comandos **MTWBCMD_GetVersion**, **MTWBCMD_GetWhiteBoardInfo**, **MTWBCMD_GetTimeout** y **MTWBCMD_SetTimeout** se envían al firmware cuando el controlador recibe o bien el comando **MTWBCMD_OpenCommPort** o bien el comando **MTWBCMD_OpenCommPortEx**. Debido a que estos comandos requieren una realimentación inmediata desde el controlador, el controlador debe intentar obtener los datos tan pronto como sea posible. Si es posible para cualquiera de estos comandos, se enviarán antes de que el controlador haya recibido los datos desde la pizarra blanca debido al retraso en la recepción de datos desde la pizarra blanca. En este caso, el controlador devolverá un 0 para indicar a la aplicación que debería esperar a una notificación desde el controlador de que los datos están listos. Una vez que la aplicación recibe la notificación, puede llamar al controlador para obtener la información. Si el controlador detecta que la pizarra blanca está desconectada del puerto serie, entonces estos comandos devolverán -1.

15 **MTWBCMD_GetTimeout**

Este comando hace que el controlador lea los datos del Bloque 1 desde el firmware para obtener el valor del tiempo de expiración del firmware. Este comando sólo se puede ejecutar después de que se haya abierto el puerto o bien con el comando **MTWBCMD_OpenCommPort** o bien con el comando **MTWBCMD_OpenCommPortEx**.

IParam1 ignorado
IParam2 ignorado

Devuelve:

- 20 El valor del tiempo de expiración del firmware en caso de éxito
- 0 si el tiempo de expiración no se ha devuelto desde el firmware. El controlador notificará a la aplicación cuando los datos se han recibido a través de la Retrollamada de Mensaje.
- 1 en caso de falla.

- 25 NOTA: Los comandos **MTWBCMD_GetVersion**, **MTWBCMD_GetWhiteBoardInfo**, **MTWBCMD_GetTimeout** y **MTWBCMD_SetTimeout** se envían al firmware cuando el controlador recibe o bien el comando **MTWBCMD_OpenCommPort** o bien el comando **MTWBCMD_OpenCommPortEx**. Debido a que estos comandos requieren una realimentación inmediata desde el controlador, el controlador debe intentar obtener los datos tan pronto como sea posible. Si es posible para cualquiera de estos comandos, se enviarán antes de que el controlador haya recibido los datos desde la pizarra blanca debido al retraso en la recepción de datos desde la pizarra blanca. En este caso, el controlador devolverá un 0 para indicar a la aplicación que debería esperar a una notificación desde el controlador de que los datos están listos. Una vez que la aplicación recibe la notificación, puede llamar al controlador para obtener la información. Si el controlador detecta que la pizarra blanca está desconectada del puerto serie, entonces estos comandos devolverán -1.

MTWBCMD_OpenCommPort

- 35 Este comando indica al controlador establecer el puerto COM y tomar control sobre él (en Windows 3.11 y superior). El controlador también instala un gestor de interrupciones del Reloj en Tiempo Real (RBC). Este comando se debe enviar para que el controlador se comunique con la pizarra blanca.

IParam1 ignorado
IParam2 ignorado

Devuelve:

VERDADERO - en caso de éxito
FALSO - en caso de fallo

MTWBCMD_CloseCommPort

- 40 Este comando indica al controlador liberar el puerto COM. El controlador también desinstala el gestor de interrupciones del RBC.

IParam1 ignorado
IParam2 ignorado

Devuelve:

VERDADERO - en caso de éxito
 FALSO - en caso de fallo

MTWBCMD_SetLED

La aplicación puede usar este comando para escribir un nuevo valor en el byte de estado del LED. Este comando sólo se puede ejecutar después de que se haya abierto el puerto o bien con el comando **MTWBCMD_OpenCommPort** o bien con el comando **MTWBCMD_OpenCommPortEx**.

5

NOTA: Los LED no están actualmente soportados por el hardware.

IParam1 el número de sensor de la pizarra blanca a cambiar, 0 a 4 (en el que 0 se interpreta como 1)
 IParam2 palabra baja = ID del rotulador o un nuevo valor para el byte completo, palabra alta = subtipo de comando

10 El subtipo de comando puede ser uno de los valores siguientes:

- 0 Poner a cero el bit especificado en la palabra baja
- 1 Poner a uno el bit especificado en la palabra baja
- 2 establecer el byte completo al valor de la palabra baja

Devuelve:

Actualmente FALSO
 Cuando simplemente VERDADERO - en caso de éxito
 FALSO - en caso de fallo

MTWBCMD_OpenCommPortEx

Este comando indica el controlador establecer el puerto COM especificado por el IParam1 y tomar su control (en Windows 3. 11 y superior). El controlador instala también el gestor de interrupciones del Reloj en Tiempo Real (RBC). Este comando se debe enviar para que el controlador comunique con la pizarra blanca.

15

IParam1 número del puerto COM
 IParam2 número de la IRQ

Devuelve:

VERDADERO - en caso de éxito
 FALSO - en caso de fallo

MTWBCMD_SetMessageCallback

Este comando se usa para establecer una Retrollamada de Mensajes en la aplicación. El controlador usa esta retrollamada para notificar los errores y para enviar los mensajes de la aplicación.

20

IParam1 contiene el número de sensor, 0 a 4 (en el que 0 se interpreta como 1)
 IParam2 contiene el puntero lejano a la rutina de retrollamada en la aplicación

El prototipo de función de retrollamada es:

void PASCAL WhiteboardMessageCallback (long SensorNumber, long MsgCode);

El código del mensaje puede ser uno de los siguientes valores:

25

```
enum MTWB_MESSAGE {
    MTWBMSG_NONE ,
    MTWBMSG_ERR_VERSION ,
    MTWBMSG_ERR_VIRTUALIZE_RTC ,
    MTWBMSG_TIMEOUT ,
    30 MTWBMSG_DISCONNECT ,
    MTWBMSG_RECONNECT ,
    MTWBMSG_ERR_PORT_READ ,
    MTWBMSG_ERR_PORT_WRITE ,
    MTWBMSG_ERR_QUEUE_PUT ,
    35 MTWBMSG_ERR_QUEUE_GET ,
    MTWBMSG_ERR_INPUT_DATA ,
    MTWBMSG_WBINFO_READY ,
    MTWBMSG_LAST
};
```

Devuelve:

Número del sensor - en caso de éxito
0 - en caso de fallo

MTWBCMD_GetVxdVersion

Este comando se usa para obtener el número de versión del Controlador y si un comando en particular está soportado por el controlador.

IParam1 es el número del comando a ser comprobado
IParam2 es el puntero a la etiqueta del comando

5 IParam1 puede ser uno de los siguientes valores:

```
enum MTWBS_COMMAND {
    MTWBCMD_NOP = -1,
    MTWBCMD_GetInfo,
    MTWBCMD_SetInfo,
    10 MTWBCMD_GetName,
    MTWBCMD_GetVersion,
    MTWBCMD_DisplayInfo,
    MTWBCMD_SetLoggingMode,
    MTWBCMD_SetMouseMode,
    15 MTWBCMD_StopLoggingMode,
    MTWBCMD_GetWhiteBoardInfo,
    MTWBCMD_SetSound,
    MTWBCMD_SetEraseMode,
    20 MTWBCMD_SetTimeout,
    MTWBCMD_GetTimeout,
    MTWBCMD_OpenCommPort,
    MTWBCMD_CloseCommPort,
    MTWBCMD_SetLED,
    25 MTWBCMD_OpenCommPortEx,
    MTWBCMD_SetMessageCallback,
    MTWBCMD_GetVxdVersion,
    MTWBCMD_RemoveMessageCallback,
    MTWBCMD_Last
};
```

30 Úsese MTWBCMD_NOP para obtener el número de versión del controlador sin necesidad de especificar una etiqueta de comando.

IParam 2 puede ser uno de los siguientes valores

```
#define MTWBCMD_TAG_GETINFO (VOID FAR*)"GetInfo"
#define MTWBCMD_TAG_SETINFO (VOID FAR*)"SetInfo"
35 #define MTWBCMD_TAG_GETNAME (VOID FAR*)"GetName"
#define MTWBCMD_TAG_GETWVERSION (VOID FAR*)"GetWbVersion"
#define MTWBCMD_TAG_DISPLAYINFO (VOID FAR*)"DisplayInfo"
#define MTWBCMD_TAG_SETLOGGINMODE (VOID FAR*)"SetLoggingMode"
#define MTWBCMD_TAG_SETMOUSEMODE (VOID FAR*)"SetMouseMode"
40 #define MTWBCMD_TAG_STOPLOGGINMODE (VOID FAR*)"StopLoggingMode"
#define MTWBCMD_TAG_GETWHITEBOARDINFO (VOID FAR*)"GetWhiteBoardInfo"
#define MTWBCMD_TAG_SETSOUND (VOID FAR*)"SetSound"
#define MTWBCMD_TAG SETERASEMODE (VOID FAR*)"SetEraseMode"
#define MTWBCMD_TAG SETTIMEOUT (VOID FAR*)"SetTimeout"
45 #define MTWBCMD_TAG GETTIMEOUT (VOID FAR*)"GetTimeout"
#define MTWBCMD_TAG OPENCOMMPORT (VOID FAR*)"OpenCommPort"
#define MTWBCMD_TAG CLOSECOMMPORT (VOID FAR*)"CloseCommPort"
#define MTWBCMD_TAG SETLED (VOID FAR*)"SetLED"
50 #define MTWBCMD_TAG OPENCOMMPORTEX (VOID FAR*)"OpenCommPortEx"
#define MTWBCMD_TAG SETMESSAGECALLBACK (VOID FAR*)"SetMessageCallback"
#define MTWBCMD_TAG GETVXDVERSION (VOID FAR*)"GetVxdVersion"
#define MTWBCMD_TAG REMOVEMESSAGECALLBACK (VOID FAR*)"RemoveMessageCallback"
```

Devuelve:

Número de versión si el comando está soportado (la etiqueta de comando interna para el número especificado es la misma)
0 si el comando no está soportado

MTWBCMD_RemoveMessageCallback

Este comando indica al controlador detener el envío de mensajes generados desde la pizarra blanca identificada en IParam1.

IParam1 gestor de la pizarra blanca (devuelto desde el **MTWBCMD_SetMessageCallback**)
 IParam2 ignorado

Devuelve:

VERDADERO - en caso de éxito
 FALSO - en caso de fallo

5 El Archivo de la Sesión

La Figura 5 de los dibujos muestra un formato del archivo de sesión. El archivo de la sesión consiste en tres elementos principales: la cabecera de archivo 76 que describe el archivo; los bloques de índices 77 que mantienen las entradas que definen las imágenes y los bloques de datos 78 que contienen todos los datos usados para generar las imágenes, incluyendo los cambios en el color y la herramienta de escritura. Estos bloques de datos 78 y los toques de índices 77 se crean según sea necesario. El bloque de cabecera 76 proporciona una información general que se aplica al archivo global. Incluyendo las mediciones de la pizarra blanca, los tamaños de borrador usados en el archivo, el tamaño de los mapas de bits instantáneos almacenados en el archivo y la identificación del archivo. El tamaño de la cabecera es de 1.024 bytes. Hay sólo un bloque de cabecera por archivo. Los bloques de índices contienen entradas de índices que pertenecen a cada imagen del archivo y un identificador de los eventos especiales. Cada bloque de índices contiene una cabecera que describe la localización de los bloques de índice siguiente y previo. Una entrada de índice determina cómo se creó la imagen, o bien por vía de un botón de instantánea o del botón de borrar todo. La entrada del índice contiene el mapa de bits de la instantánea en formato comprimido, un nombre de longitud variable de la imagen, la localización de comienzo y final de los datos en el archivo, la fecha y hora en la que se creó, junto con otros atributos. Una entrada de índice puede variar de tamaño, pero los bloques de índice se establecen en 65.536 bytes. El tamaño es mayor debido a que los mapas de bits de las instantáneas se almacenan en el bloque de índices. Los bloques de datos contienen todos los datos usados para generar la imagen. Los datos se componen de puntos de datos que se almacenan en un formato comprimido. Hay un punto absoluto seguido por puntos relativos. Esta serie de puntos componen un vector. Hay apartados de datos de eventos especiales que significan un cambio de color o de modo. Un bloque de datos tiene 4.096 bytes.

25 Ventajosamente, el archivo de sesión contiene la información necesaria para recrear las relaciones secuenciales y temporales generadas con la entrada de información por medio de la pizarra blanca. Por ejemplo, cada trazado introducido sobre la zona de escritura, incluyendo los trazados del borrador se guarda en el archivo de sesión, junto con las relaciones temporales necesarias para recrear cada trazado. Por ello, el archivo de sesión permite la visión y edición posterior, trazado por trazado, de una imagen. Los trazados con el borrador se pueden seleccionar, ver y eliminar, tal como puede hacerse con los trazados con el marcador o con el dedo del usuario. Además, el almacenamiento de la información temporal permite el uso de un software de reconocimiento de escritura manual. La visualización de una imagen creada en la pizarra blanca en una pantalla de ordenador se puede realizar mediante la visualización de cada trazado almacenado en el archivo de sesión. Por ello, un trazado con un marcador se visualiza de acuerdo con el color designado para el marcador en el archivo de sesión y un trazado con un borrador se visualiza en blanco de acuerdo con el ancho del borrador elegido (estrecho o ancho).

Componente de la aplicación

El componente de la aplicación proporciona toda la funcionalidad (no contenida dentro del controlador) para soportar la operación de la pizarra blanca. Esto incluye una GUI, la gestión del sistema de archivos, las comunicaciones API, el control de proceso, etc.

40 En el inicio, la aplicación se ejecuta minimizada (condición por defecto) o como la ventana más al fondo en la pantalla 13 (opción seleccionable por el usuario). Un botón predefinido (Mostrar Tablero) se proporciona en la pizarra blanca para traer la ventana minimizada a primer plano y mostrar el estado actual del proceso de grabación. La aplicación responde a este botón restaurándose a sí misma en el foco principal y mostrando la imagen del tablero actual como se ve en la Fig. 1.

45 Una función de registro dentro de la aplicación recibe flujos de eventos de la pizarra blanca desde el controlador (la estructura exacta de los datos se ha descrito anteriormente), representando las coordenadas de la actividad de escritura actual del usuario. Los puntos redundantes y puntos superfluos (es decir alineados y muy cerca en su separación) se eliminan del flujo de datos por el controlador previamente al almacenamiento de los datos en el archivo de sesión, proporcionando así un mecanismo de compresión de datos de pérdida cero limitado. Por ejemplo, si el usuario presiona y mantiene un dedo o marcador en la superficie de la pizarra blanca en un punto particular de modo que produzca la generación repetida del mismo dato para indicar la acción, la función de registro recibirá solamente el primer caso de los datos y grabará un valor indicativo del número de veces que el punto se repitió. Además, si el controlador recibe una primera coordenada y una segunda coordenada, dentro de un intervalo

ajustable de la primera coordenada, entonces la segunda coordenada se descartará y solamente se enviará la primera al registrador. Los expertos en la materia a la vista de la presente divulgación apreciarán que se pueden emplear otros procedimientos para eliminar los puntos redundantes y superfluos.

- 5 En momentos o eventos predeterminados (definidos por el usuario) la aplicación inserta un marcador en el flujo de datos, marcando de modo efectivo el estado del tablero actual para revisión. La aplicación de visualización proporciona una vista instantánea de cada estado del tablero marcado. El usuario puede colocar explícitamente una marca a través de un botón programable. La aplicación también proporciona marcadores automáticos para eventos tales como "borrar todo" como se describe adicionalmente en el presente documento.

Interfaz Gráfica del Usuario (GUI)

- 10 La aplicación proporciona una Interfaz de Documento Única (SDI) como interfaz pero también permite capacidades de ventanas divididas como se describe adicionalmente en el presente documento. La disposición de ventanas es similar a la mostrada en la figura 1. La aplicación proporciona tres elementos visibles principales en la ventana de la SDI: vista de selección, control de listas e imagen reducida del tablero actual.

- 15 La vista de selección es una zona subdividida de la SDI de aplicación y proporciona el medio principal para ver cualquier instantánea dentro del archivo de sesión actual. La zona de visión permite que la magnificación de la imagen sea ajustable desde el menú de visión o botones de la barra de tareas y su tamaño es ajustable moviendo las barras de división. La relación de aspecto de la imagen presentada se preserva (por defecto) a menos que el usuario elija otra opción por medio de los ajustes de preferencias. El color de fondo por defecto de la vista de selección es blanco con trazados de rotulador coloreados. Los anchos de los trazados del rotulador visualizados son ajustables mediante una preferencia del usuario. El ancho mostrado y almacenado de los trazados del borrador es ajustable por separado mediante una preferencia del usuario. La vista de selección muestra o bien una instantánea seleccionada del control de listas o la imagen del tablero actual de la instantánea del tablero actual.

- 20 La vista del tablero actual es también una zona subdividida de la SDI de la aplicación y proporciona medios secundarios para ver el estado actual de la pizarra blanca. El tamaño de la vista es ajustable moviendo las barras de división y la relación de aspecto de la vista se preserva (por defecto) a menos que el usuario escoja otra cosa a través de los ajustes de preferencias. El color de fondo por defecto de la vista del tablero actual es también blanco con trazados de rotulador coloreados. El borde de la vista es una representación gráfica ajustable de la pizarra blanca real. La vista del tablero actual se actualiza frecuentemente para mostrar la actividad de dibujo en el tablero sustancialmente en tiempo real.

- 25 El control de la lista de instantáneas comprende una tercera parte subdividida de la SDI de la aplicación y proporciona los medios principales para clasificar y ver el conjunto de instantáneas almacenadas en el archivo de sesión actual. El tamaño del control de la lista de instantáneas es ajustable moviendo las barras de división apropiadas. La resolución del mapa de bits de cada instantánea permanece fija permitiendo que más o menos instantáneas sean visibles cuando el tamaño del control de la lista se ajusta. El color de fondo por defecto del control de la lista es gris oscuro y el color de fondo por defecto de cada instantánea es blanco con trazados del rotulador coloreados. Las instantáneas se pueden seleccionar usando secuencias de ratón y teclado de Windows estándar. Las opciones de selección incluyen una, varias contiguas, varias individuales o todas. El significado de este proceso de selección cambia para las diferentes acciones deseadas de acuerdo con las convenciones impuestas por el sistema operativo Windows particular. Por ejemplo, si una imagen se arrastra a otra aplicación, sólo la última imagen seleccionada será arrastrada a la otra aplicación. Durante un movimiento o adición, todas las instantáneas seleccionadas se mueven o añaden en el orden seleccionado y si se copian al portapapeles y se pegan, todas las instantáneas seleccionadas reaparecen cuando se pegan en el orden seleccionado.

- 30 Se proporcionan barras de desplazamiento según sea apropiado para la navegación a través de la lista disponible de instantáneas. Se muestra un campo de nombre editable junto con cada instantánea. Por defecto, este campo se inicializa con la hora de la creación y la fecha de la creación de la instantánea. Esta información de hora y fecha se muestra usando una pequeña ventana de herramienta de notas flotante cuando el cursor del ratón descansa sobre una instantánea. Esta característica permite que el usuario renombre las instantáneas sin perder el acceso a la información de la fecha de creación. La barra del título del control de la lista de instantáneas muestra del texto "Imágenes guardadas: N" en el que "N" es el número de instantáneas.

- 35 La GUI proporciona la funcionalidad de Incrustación y Enlazado de Objetos (OLE) para permitir que se seleccione una instantánea y se arrastre mediante el ratón a otra aplicación compatible con OLE abierta. Esta acción da como resultado que el contenido de esa instantánea sea pegado en la aplicación objetivo. La GUI también proporciona la funcionalidad necesaria para permitir que una instantánea sea seleccionada y arrastrada mediante el ratón sobre un botón o icono de programa del escritorio (según sea apropiado) para lanzar al programa objetivo con una copia del contenido de esa instantánea.

La GUI proporciona la funcionalidad necesaria para permitir que las instantáneas sean reordenadas y borradas dentro del control de la lista. Este usuario realiza esta tarea seleccionando la instantánea objetivo y a continuación o bien arrastrándola a la localización deseada dentro del control de la lista o bien presionando la tecla "Suprimir" para

eliminarla. El borrado de múltiples instantáneas seleccionadas está también soportado. Por defecto, se usa una ventana de diálogo para confirmar la acción de borrado. Las instantáneas borradas no son recuperables.

5 La GUI proporciona la funcionalidad necesaria para permitir que los usuarios oculten temporalmente instantáneas seleccionadas. Las instantáneas ocultas permanecerán con los marcadores en el archivo de la sesión pero no se mostrarán en el control de la lista a menos que se habilite la opción del menú de ver para mostrar las instantáneas ocultas. En este caso, las instantáneas ocultas se muestran con un borde especial para indicar su estado oculto.

La GUI proporciona preferentemente, los siguientes apartados de menú:

Menú Archivo

10 El menú de archivo proporcionará soluciones para *nuevo, abrir, guardar, guardar como, cerrar, exportar imagen, imprimir, Vista previa, Ajustes de impresión*, lista de los archivos abiertos previamente más recientes, *salir*. La función de *cerrar* hace que el archivo de pizarra blanca actual sea cerrado y se cree un nuevo archivo sin título.

Menú Edición

15 El menú de edición proporciona opciones para *copia, añadir imagen(es) a..., mover imagen(es) a..., borrar, seleccionar todo, ocultar, mostrar, ocultar todo, instantáneas, opciones*. Cuando el tablero actual tiene el foco: *exportar, copiar, añadir imagen(es) a* están habilitados y *mover imagen(es) a, borrar* y *cortar* están atenuados. La función de añadir imágenes toma la imagen actualmente seleccionada y añade una copia de ella con el contexto del trazado al archivo de sesión objetivo seleccionado. Esta operación (si tiene éxito) se confirma a continuación mediante un cuadro de diálogo emergente que informa al usuario de que la adición tuvo éxito. Cuando la selección actual es la vista del tablero actual, se convierte en la vista del tablero actual en el archivo de sesión objetivo. El tablero actual en el archivo de sesión objetivo se preserva como una instantánea. Cuando la selección actual es una instantánea o instantáneas, la imagen o imágenes se copian en el archivo objetivo y aparecen como las últimas instantáneas en el archivo objetivo. El tablero actual permanece sin cambios en el archivo objetivo.

20 La función de *Mover imágenes a* toma la imagen actualmente seleccionada y añade una copia de ella con el contexto del trazado al archivo de sesión objetivo seleccionado. La imagen seleccionada actualmente se borra entonces del archivo de sesión actual. Esta operación (si tiene éxito) se confirma a continuación mediante un cuadro de diálogo emergente que informa al usuario de que el movimiento tuvo éxito. Cuando la selección actual es la vista del tablero actual, esta función está inhabilitada. Cuando la selección actual es una instantánea o instantáneas, la imagen o imágenes se copian en el archivo objetivo y aparecen como las últimas instantáneas en el archivo objetivo. El tablero actual permanece sin cambios en el archivo objetivo.

25 La función *cortar* permite que el usuario borre una imagen reducida seleccionada y la coloque en el portapapeles al mismo tiempo. La función *cortar* está inhabilitada (atenuada) si se seleccionan múltiples imágenes reducidas. La función de cortar no funciona en la imagen del tablero actual.

Menú Vista

30 El menú vista proporciona opciones para *herramientas, lista de imágenes reducidas, barra de estado, tablero actual, disposición de pantalla..., imágenes reducidas ocultas, acercar zoom, alejar zoom*.

Menú Ayuda

El menú ayuda proporciona opciones para *índice* y *acerca de*.

Botones de la barra de herramientas

35 La GUI proporciona una barra de herramientas móvil. Las funciones soportadas por los botones incluyen *nuevo, abrir, guardar, imprimir, imagen reducida, cortar, copiar, tablero actual, ver transparencia, vista normal, plena pantalla, acercar zoom, alejar zoom, tamaño a ventana y ayuda sensible al contexto*.

Barra de estado

La GUI Incluye una barra de estado que proporciona ayuda al usuario indicando el estado actual de la aplicación y la fecha y hora de creación de la imagen reducida seleccionada.

40 La aplicación soporta la impresión de una única y múltiples imágenes (una por página) en cualquier impresora gráfica de color o blanco y negro apropiadamente configurada para Windows. Las opciones de impresión incluyen la imagen del tablero actual, un grupo de imágenes instantáneas a partir de una selección múltiple o todas las imágenes en el archivo actual incluyendo el tablero actual.

45 Un cuadro de diálogo de impresión incluye opciones para especificar la cabecera y el pie de página por medio de menús desplegables y cuadros de texto editables. El formateo de la cabecera y el pie pueden realizarse por medio de caracteres de control embebidos que permiten varios formatos de fecha y hora representativos de o bien la fecha

y/o la hora actual o bien la fecha y/o la hora de creación. Otros caracteres de control permiten la especificación de tipos de letra, atributos de los tipos tales como negrita, cursiva y tamaño de letra. La aplicación proporciona una pantalla de vista previa de la impresión con controles de zoom para permitir que el usuario vea previamente la acción de la impresora objetivo. La GUI proporciona dos niveles de ayuda al usuario en línea: general en el que puede buscarse y sensible al contexto. La GUI proporciona tres botones de selección de vista en la barra de herramientas. Estos botones proporcionan los medios para conmutar la GUI entre una vista de disposición normal, una vista de plena pantalla de clasificación de transparencias y una vista a plena pantalla página a página (o transparencia a transparencia).

Sistema de gestión de archivos

La aplicación almacena toda la actividad de dibujo y borrado del tablero secuencialmente en un archivo de datos denominado un archivo de "sesión". Solamente está abierto un archivo de sesión a la vez y toda la actividad del tablero se añade al final lógico (no necesariamente físico) de ese archivo. La gestión de este archivo es consistente con el producto SDI excepto en lo observado en el presente documento. Este archivo contiene toda la información necesaria para recuperar y ver instantáneas y para navegar entre marcadores trazado a trazado. El sistema de archivos soporta borrado, ocultación y reordenación de las imágenes reducidas. El sistema de archivo se diseña ventajosamente para minimizar la pérdida de datos en el caso de finalización anormal del proceso. Se usan archivos temporales e indicadores para restaurar el estado previo de la aplicación del sistema de archivos cuando la aplicación se reinicia después de una finalización anormal del proceso. El archivo por defecto para la entrada de datos desde el tablero es el último archivo de sesión activo antes de que la aplicación se cerrase. El formato del archivo incluye lo siguiente: (1) lista de vectores, que contienen los trazados individuales escritos en la pizarra blanca; (2) marca de tiempo de los datos escritos; (3) resolución de al menos 4000 puntos tanto en las dimensiones X como Y y (4) imágenes instantáneas almacenadas como mapas de bits.

Acciones de control del archivo de sesión

La aplicación soporta preferentemente, las siguientes acciones de control de archivo iniciadas por el usuario. **NUEVO** – Se crea un nuevo archivo de sesión vacío siempre que el usuario seleccione "nuevo" desde el menú archivo. El archivo de sesión actual se cierra con un marcador por defecto añadido al final y todas las acciones del tablero posteriores irán dentro de este nuevo archivo sin título. Se da la opción al usuario de mantener o descartar los cambios no guardados en el archivo actual, que va a ser cerrado. Bajo condiciones por defecto, la imagen del tablero actual no se traslada al nuevo archivo. Sin embargo, se puede proporcionar una opción para permitir que el usuario cambie este comportamiento por defecto usando los ajustes de las opciones.

ABRIR – Un archivo de sesión creado previamente que actualmente está cerrado se vuelve a abrir para ver e introducir datos del tablero. El archivo de sesión actual se cierra con un marcador por defecto añadido al final y todas las acciones del tablero posteriores irán a este archivo vuelto a abrir. Se da la opción al usuario de mantener o descartar los cambios no guardados en el archivo de sesión actual, que va a ser cerrado.

GUARDAR – Esta función pone cualquier cambio no guardado en el archivo de sesión actual. Los cambios incluyen imágenes reducidas añadidas, reordenadas, ocultas, borradas y cualquier dato de dibujo del tablero desde que se guardó por última vez.

GUARDAR COMO – Esta función crea una copia completa del archivo de sesión actual y comienza a usar la nueva copia para todas las actividades del tablero adicional y de edición. Cualquier cambio sin guardar en el archivo de sesión actual irá en la copia creada de nuevo, no en el archivo original.

AÑADIR IMAGEN A – Esta función permite que el usuario añada una copia de la imagen seleccionada actualmente al final de otro archivo de sesión.

MOVER IMAGEN A – Esta función permite que el usuario añada una copia de la imagen seleccionada actualmente al final de otro archivo de sesión y borre inmediatamente a continuación la imagen seleccionada actualmente del archivo de sesión actual.

Preservación de datos

Una prioridad en el diseño del sistema de archivos es la preservación de los datos de la pizarra blanca del usuario. El diseño incorpora técnicas de redundancia apropiadas para asegurar que se pierden los datos mínimos en el caso de interrupción del sistema, etc. Cualquier archivo de sesión sin guardar, sin título es seguido por un archivo temporal oculto para asegurar la recuperación de datos hasta que el usuario decida activamente guardar o eliminar ese archivo de sesión nuevo.

Compresión de datos

El software se diseña para usar varias técnicas de compresión de datos que incluyen la codificación de longitud de repetición (run length) para minimizar el espacio de disco usado por los archivos de sesión.

Exportar datos

La aplicación soporta tres métodos para exportar datos: portapapeles, archivo, arrastrar y soltar de OLE. Éstos incluyen la capacidad de copiar la imagen seleccionada actualmente al portapapeles y exportar la imagen seleccionada actualmente a un archivo gráfico. Los formatos gráficos para los tres modos incluyen al menos el de mapa de bits (.BMP) y metarchivo de Windows (.WMF).

5 Características avanzadas

La aplicación soporta un conjunto de características avanzadas que se inhabilitan por defecto en una instalación normal. Estas características se activan mediante alguna clase de selección dentro del diálogo de opciones. Una vez habilitadas, estas características avanzadas proporcionan soporte para las siguientes funciones como mínimo:

- 10 a) Edición de marcadores – La capacidad de editar las propiedades del marcador a través de un diálogo.
- b) Navegar por trazados – La capacidad de navegar a través de un archivo trazado a trazado y página a página usando las teclas de flechas, teclas de página e iconos de la barra de herramientas.
- c) Eventos del ratón – Cuando se habilitan a través de un cuadro de comprobación separado, esta característica produce la salida de los movimientos del cursor del ratón y mensajes seleccionados por el ratón desde el controlador de la pizarra blanca al sistema operativo Windows.

15 Ajustes opcionales

La aplicación proporciona un apartado de selección de “Opciones” desde el menú edición que producirán a continuación una serie de diálogos en el estilo de un cuaderno de notas con pestañas. Se pueden seleccionar, como mínimo, los siguientes parámetros:

- 20 a) Preservar la relación de aspecto del dibujo del tablero.
- b) Anular salvapantallas durante la actividad de dibujo. – El valor por defecto para esta característica es habilitado.
- c) Guardar los ajustes y disposición de pantalla al salir.
- d) Seleccionar anchos de marcador y borrador.
- 25 e) Anular anchos del rotulador.
- f) Conmutar al modo de dibujo por defecto. – Esta característica define el número de minutos y segundos que deberían transcurrir antes que el estado del dibujo del tablero cambie por defecto a rotulador negro. El valor por defecto debería ser 2:00 minutos.
- g) Marcador después de n minutos de inactividad. – Esta característica está inhabilitada por defecto.
- h) Sonidos.
- 30 i) Mostrar imágenes ocultas.
- j) Usar archivo previo al inicio. – Esta característica está inhabilitada por defecto.

Ajustes del ancho del rotulador y borrador

35 Como se ha observado previamente, la aplicación soporta la selección del usuario de los anchos del rotulador y del borrador. Se soporta un ancho para todos los cuatro colores de rotulador. Existe un ancho definido por separado para el borrador ancho y para el borrador estrecho. Los anchos del borrador se almacenan en el archivo de sesión y se muestran siempre tal como fueron registrados. Los anchos del rotulador se almacenan también en el archivo de sesión y se muestran siempre como fueron registrados a menos que el usuario elija una selección para anular los valores de los anchos de rotulador mostrados. La anulación no afecta a los valores Guardados previamente.

40 Cuando el usuario cambia cualquier valor de ancho, el cambio se aplica desde ese momento en adelante en el proceso de grabación. No se requiere editar los anchos guardados previamente.

Se ha de entender que los mecanismos y técnicas específicos que se han descrito son meramente ilustrativos de una aplicación de los principios de la invención. Se pueden realizar numerosas modificaciones a los procedimientos y aparatos descritos sin separarse del verdadero espíritu y alcance de la invención. Por ejemplo, como se muestra en la Figura 7, la GUI puede tomar otras formas. En la Figura 7, la ventana 90 muestra ventajosamente la imagen 91 introducida por el usuario sobre la pizarra blanca 10. Una pluralidad de imágenes reducidas, que son versiones reducidas de las imágenes introducidas por medio de la pizarra blanca se muestran en 92 y 93. La imagen reducida mostrada en 92 es representativa de la imagen que se está visualizando (la imagen activa) por el visor. Por ello, en el ejemplo mostrado en la Fig. 7, la imagen reducida 92 es una imagen reducida de la imagen 91. Cualquiera de las imágenes reducidas tales como la 93 se puede convertir en la imagen activa para sustituir a la imagen 92 mediante la selección de la imagen reducida por medio de tecnologías bien conocidas empleadas por los usuarios del sistema operativo Windows para seleccionar elementos particulares. Se pueden mostrar imágenes reducidas adicionales para su uso en las imágenes de los botones mostrados en 94. Estos botones permiten que el usuario haga que las imágenes reducidas mostradas en 92, 93 cambien en respuesta al botón particular seleccionado. Por ejemplo, el usuario puede moverse hacia atrás a través de las imágenes reducidas para ver imágenes reducidas creadas previamente a la imagen reducida actualmente visualizada. El usuario puede moverse también hacia adelante a través de las imágenes reducidas para ver solamente la imagen reducida creada más recientemente. Por ello, las imágenes reducidas y los botones 24 permiten que el usuario recupere y vea rápidamente una pluralidad de imágenes, para seleccionar una imagen para visión, impresión, manipulación, etc. adicional.

Los botones 95 permiten la visión de secuencias con una imagen activa particular. Por ejemplo, cada trazado del marcador 24 se puede sustituir secuencialmente, para permitir la visión de la secuencia de trazados que produjeron la generación de una imagen tal como se muestra en 91. Por ello mediante el uso de los botones 95, se puede añadir o eliminar el estado correspondiente a la imagen activa permitiendo de este modo la edición de la imagen.

5 Por ejemplo, ciertas partes de la imagen 91 se pueden eliminar y la imagen resultante mostrada en el visor se puede entonces guardar, imprimir o transmitir.

Además, la estructura particular del software descrito en el presente documento se puede modificar también dentro de los principios de la presente invención. Por ejemplo, las funciones realizadas por la aplicación se puedan realizar mediante una pluralidad de programas, tal como un programa grabador que se ejecuta en el PC para grabar todas las acciones sobre la pizarra blanca y un programa visor que se puede invocar de vez en cuando para ver y manipular imágenes grabadas por el grabador. Las funciones realizadas por el controlador de la aplicación se pueden reasignar también en otras formas.

10

Serán también evidentes modificaciones adicionales para los expertos en la materia a la vista de la divulgación contenida en el presente documento.

15

REIVINDICACIONES

1. Una pizarra blanca (10) dispuesta para conectarse a un ordenador (12) que responde a las entradas desde la pizarra blanca, resultantes las entradas al ordenador de los toques sobre una superficie de escritura (14) de la pizarra blanca, comprendiendo la pizarra blanca:

5 un área activa sobre la superficie de escritura dispuesta para que el usuario proporcione escrituras, dibujos u otras anotaciones en color que puedan borrarse, siendo el área activa sensible a la presión, en la que la presión sobre el área activa da como resultado la transmisión de datos al ordenador y una función de grabador para la creación de un archivo de sesión (66) de la actividad realizada por dicho usuario sobre dicha superficie de escritura, estando la pizarra blanca **caracterizada porque** los puntos redundantes de la actividad realizada sobre la superficie de escritura no se almacenan en el archivo de sesión.

2. Una pizarra blanca según se reivindica en la reivindicación 1 **caracterizada porque** comprende:

una zona de escritura (18) sobre la superficie de escritura;
medios, sensibles a la actividad de dibujo por parte de dicho usuario en dicha zona de escritura para producir la transmisión de información que representa dicha actividad de dibujo al ordenador;
15 una zona de control (20) sobre la superficie de escritura, que incluye:

un primer botón que, cuando se selecciona, indica un color y un segundo botón, medios, sensibles a la selección de dicho primer botón para producir la transmisión de información del color del rotulador a dicho ordenador que es indicativa de un color de rotulador seleccionado por dicho usuario para dicha actividad de dibujo en dicha información posterior a dicha selección y
20 medios, sensibles al comando de ver pantalla introducido por medio de dicho segundo botón, para producir la transmisión de un comando a dicho ordenador que hace que dicho ordenador presente para visualización, un programa de aplicación que contiene una imagen que corresponde a la actividad de dibujo por parte de dicho usuario.

3. Una pizarra blanca según se reivindica en la reivindicación 2 **caracterizada porque**

dicho usuario puede borrar un dibujo en la zona de escritura usando un borrador (25) y la pizarra blanca comprende además:

un tercer botón de la zona de control y medios, sensibles a un primer comando de borrar introducido por medio de dicho tercer botón, para producir la transmisión de un comando a dicho ordenador para interpretar dicha actividad de dibujo posterior como borrado de un área en dicha zona de escritura y para responder al mismo borrando un área correspondiente en dicha imagen actual.

4. Una pizarra blanca según se reivindica en la reivindicación 3 **caracterizada porque**

dicho tercer botón indica un modo de borrado estrecho cuando se selecciona y dicha pizarra blanca comprende además:

un cuarto botón que, cuando se selecciona, indica un modo de borrado ancho; y medios, sensibles a la selección de dicho cuarto botón, para producir la transmisión de un comando a dicho ordenador para interpretar dicha actividad de dibujo posterior como el borrado de un área en dicha zona de escritura, interpretando dicho ordenador dicha actividad de dibujo posterior en dicha información como productora del borrado de un área mayor que si dicha actividad de dibujo posterior hubiese sido posterior a la selección de dicho tercer botón.

5. Una pizarra blanca según se reivindica en la reivindicación 2 **caracterizada porque** adicionalmente comprende:

un quinto botón que, cuando se selecciona, indica una instantánea, medios, sensibles a la selección de dicho quinto botón, para producir la transmisión de un comando de instantánea a dicho ordenador para hacer que dicho ordenador almacene una información de marcador que haga que dicho ordenador interprete una parte de dicha información designada por dicha información de marcador como que comprende una imagen instantánea.

6. Una pizarra blanca según se reivindica en la reivindicación 5 **caracterizada porque** adicionalmente comprende:

un sexto botón y medios, sensibles a un comando de borrar todo introducido por medio de dicho sexto botón, para producir la transmisión a dicho ordenador de dicho comando de instantánea y para producir la transmisión a dicho ordenador de un comando que haga que la imagen actual sea borrada completamente en dicha pantalla.

7. Una pizarra blanca según se reivindica en la reivindicación 2 **caracterizada porque** adicionalmente comprende:

un séptimo botón que, cuando se selecciona, indica que dicha actividad de dibujo que está siendo visualizada en dicha pantalla se ha de imprimir en una impresora accesible desde dicho ordenador y medios, sensibles a la selección de dicho séptimo botón, para producir la transmisión a dicho ordenador de un comando que haga que dicho ordenador produzca la impresión de una imagen correspondiente a dicha imagen actual.

8. Una pizarra blanca según se reivindica en cualquier reivindicación precedente, **caracterizada porque** la pizarra blanca comprende dos hojas conductoras separadas físicamente pero que hacen contacto entre sí cuando se tocan y producen un cambio detectable en la tensión o resistencia.

9. Una pizarra blanca como se reivindica en la reivindicación 1 **caracterizada porque** el archivo de sesión permite la visión posterior de trazados individuales realizados sobre dicha superficie de escritura y/o el archivo de sesión almacena trazados y relaciones temporales entre los trazados realizados sobre la superficie de escritura, en el que un trazado son los puntos de datos generados desde el momento en que un marcador u otro instrumento tal como un dedo humano o un borrador se presiona sobre la superficie de escritura de la pizarra blanca con suficiente presión para producir la transmisión de datos desde la pizarra blanca al ordenador hasta el momento en que la presión anteriormente mencionada aplicada a la superficie de escritura se finaliza mediante la elevación del marcador, dedo o borrador, o mediante el contacto con cualquier borde de la pizarra blanca.

10. Una pizarra blanca como se reivindica en la reivindicación 1 **caracterizada porque** comprende una función de visor para la visualización de una imagen de los trazados registrados en el archivo de sesión según se realizaron por dicho usuario sobre dicha superficie de escritura, en el que un trazado son los puntos de datos generados desde el momento en que un marcador u otro instrumento tal como un dedo humano o un borrador se presiona sobre la superficie de escritura de la pizarra blanca con suficiente presión para producir la transmisión de datos desde la pizarra blanca al ordenador hasta el momento en que la presión anteriormente mencionada aplicada a la superficie de escritura se finaliza mediante la elevación del marcador, dedo o borrador o mediante el contacto con cualquier borde de la pizarra blanca.

11. Una pizarra blanca como se reivindica en la reivindicación 10 **caracterizada porque** la función de visor permite la edición de trazados en el archivo de sesión.

12. Un sistema de pizarra blanca que comprende una pizarra blanca (10) cómo se define en cualquier reivindicación precedente conectada a un ordenador (12) que responde a las entradas desde la pizarra blanca.

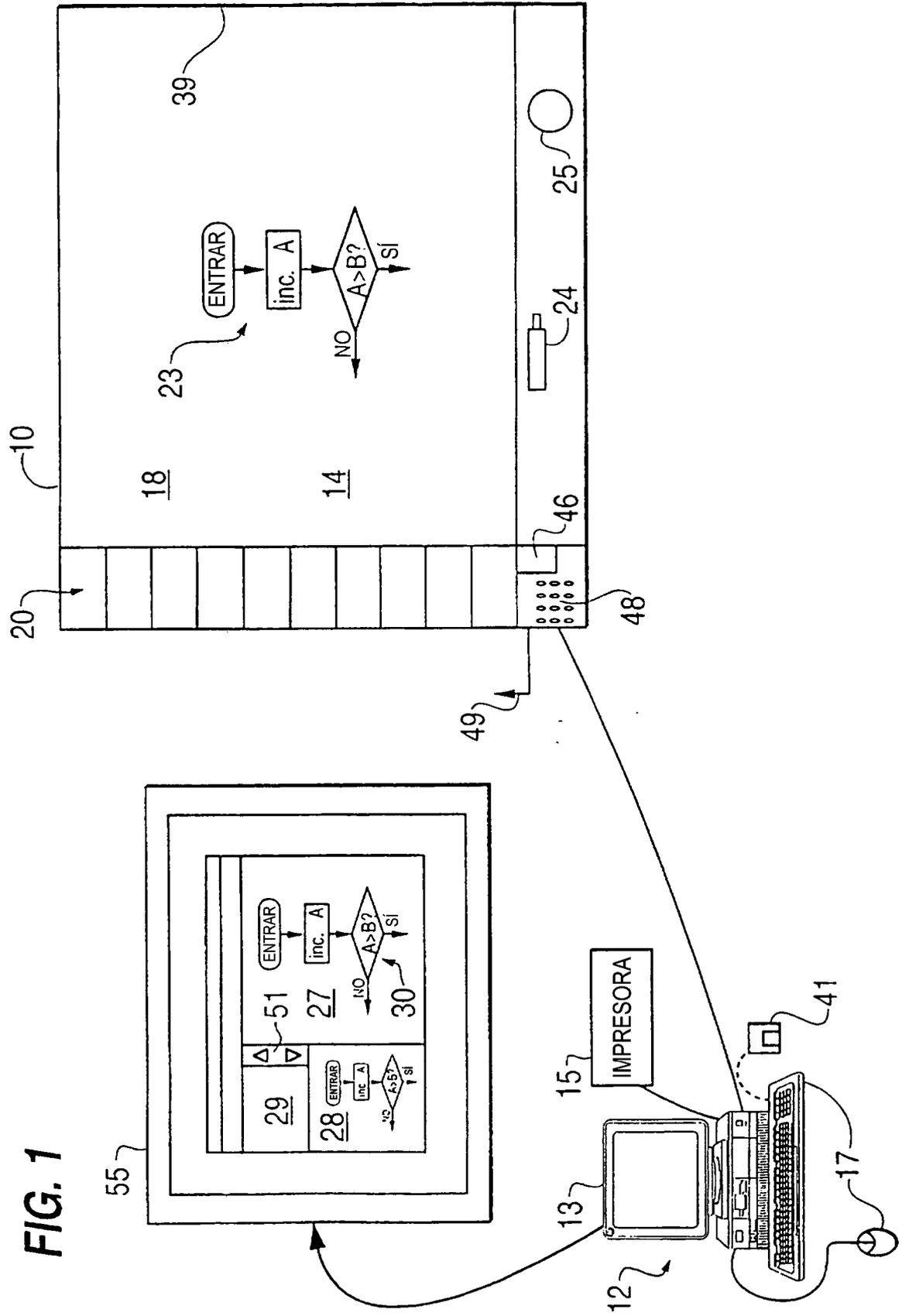
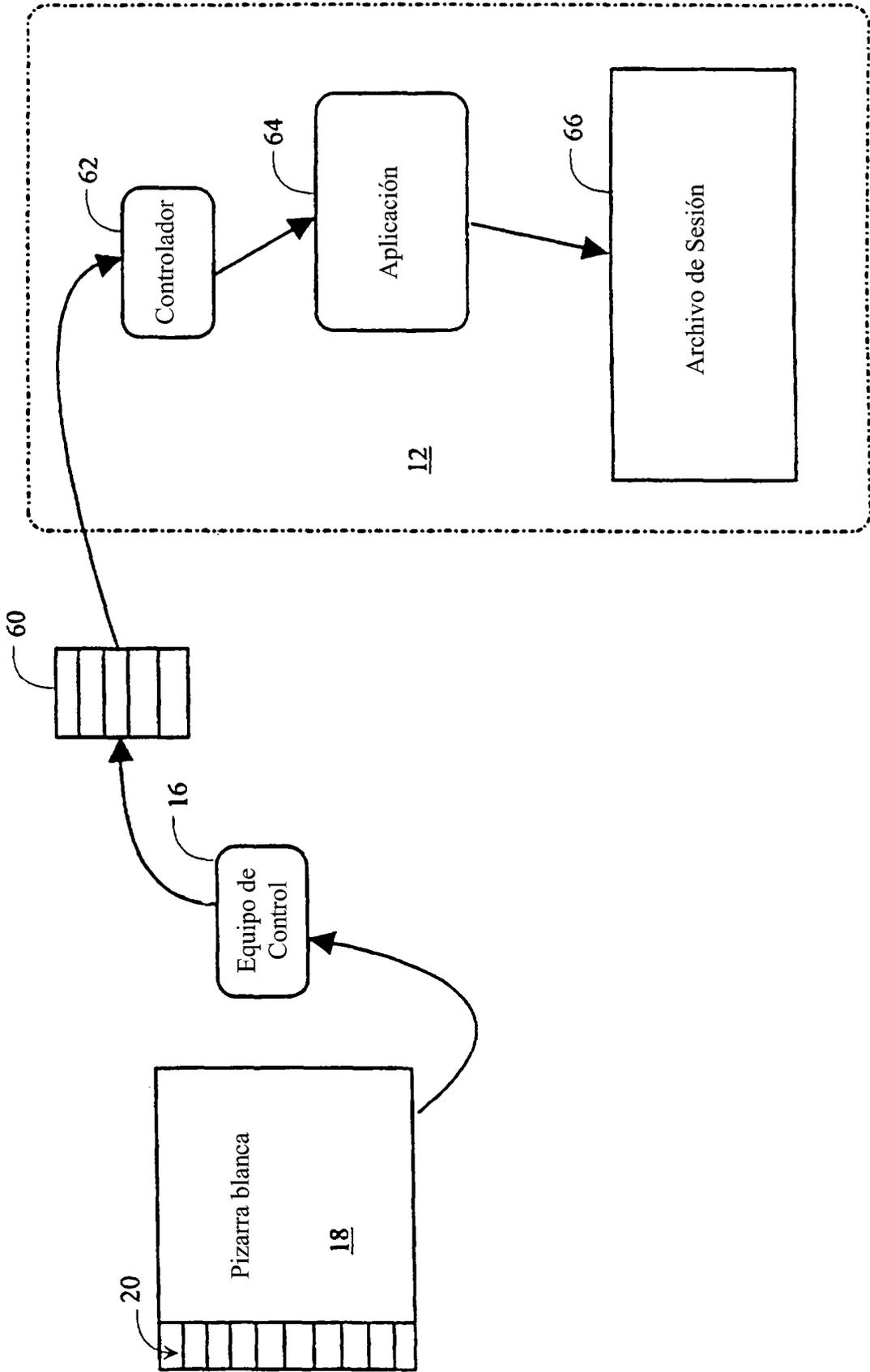


FIG. 1

FIG. 2



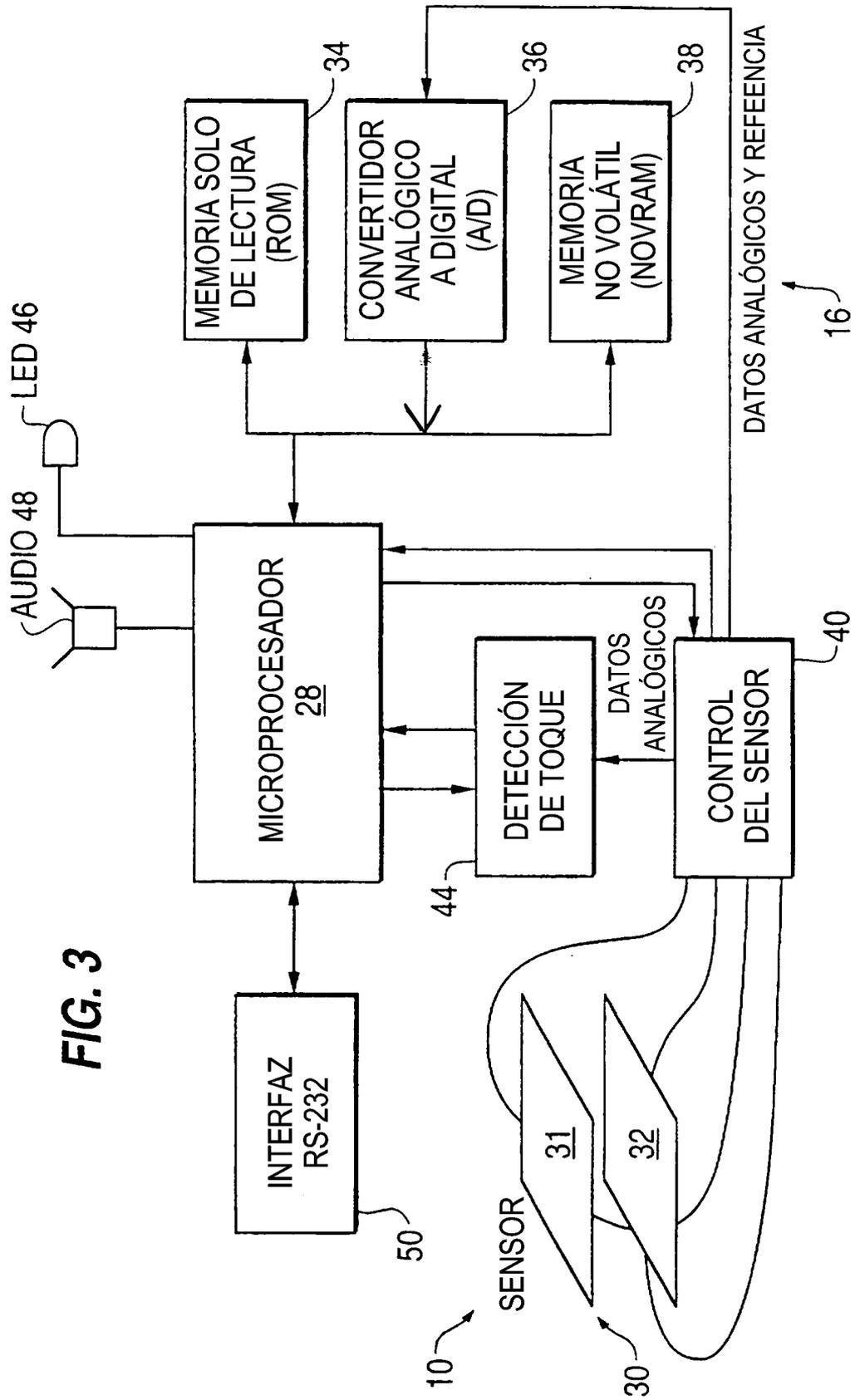


FIG. 3

FIG. 4

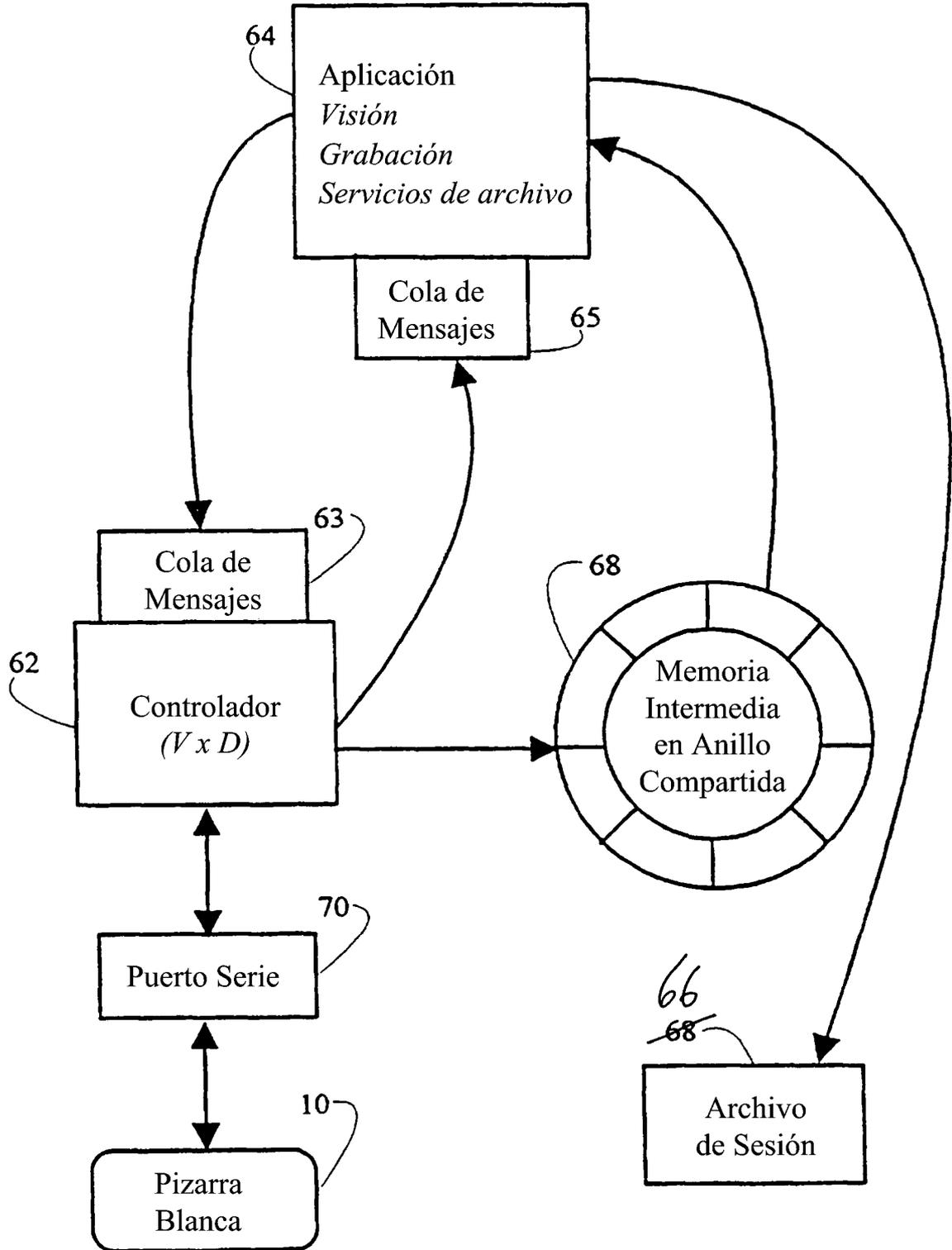


FIG. 5

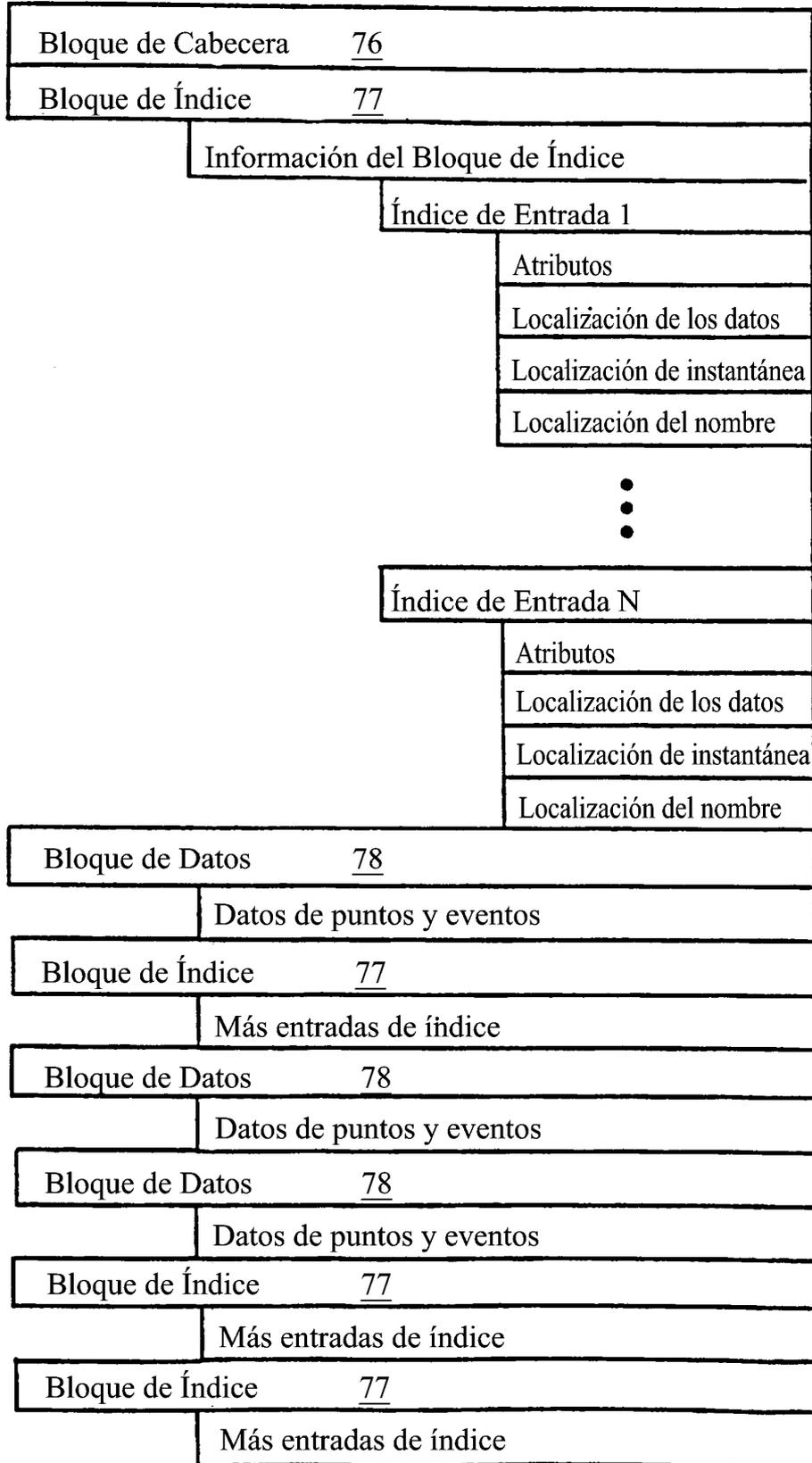


FIG. 6

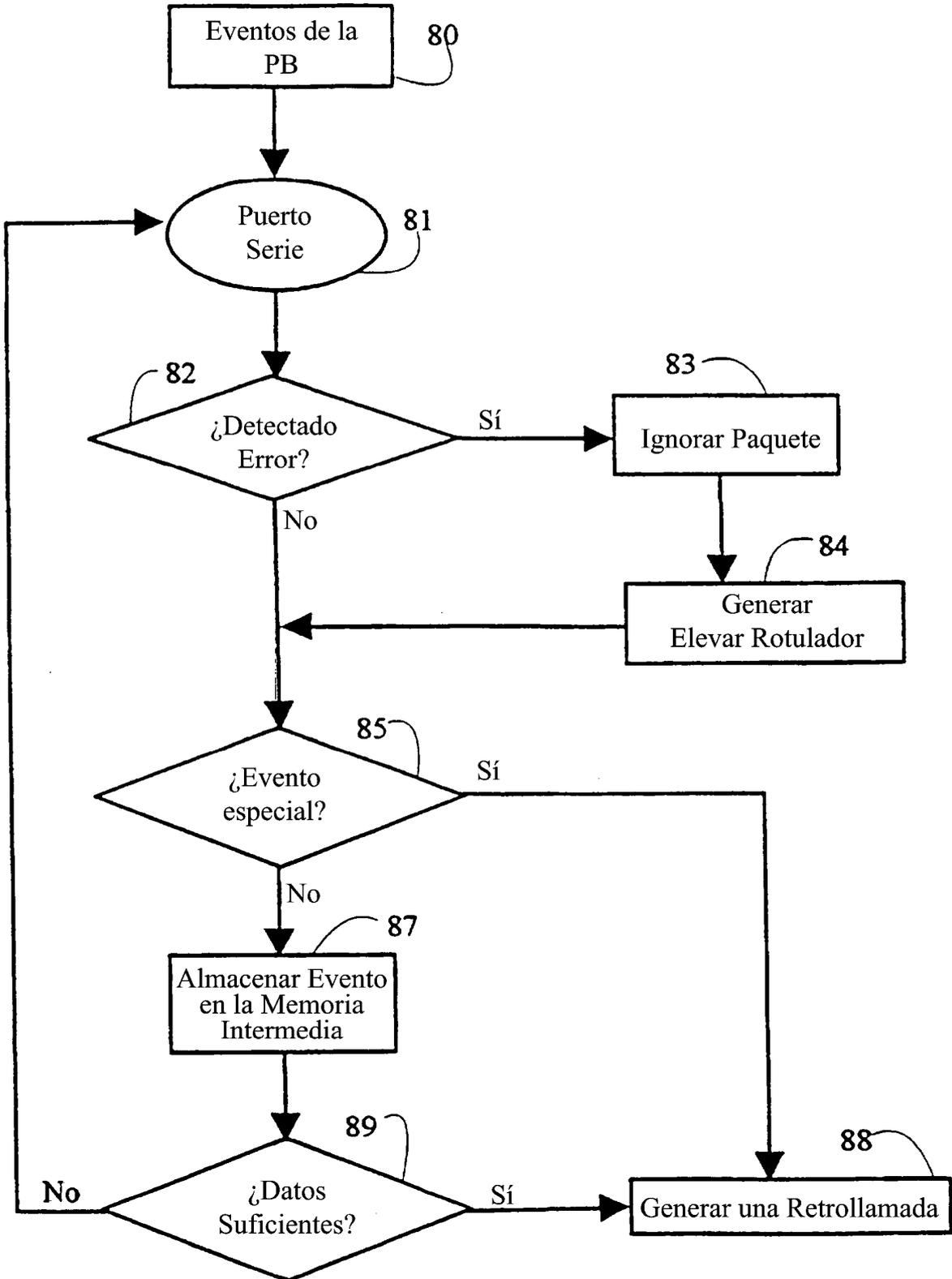


FIG. 7

