



19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

11 Número de publicación: **2 362 645**

51 Int. Cl.:
G06F 9/46 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Número de solicitud europea: **02003606 .7**

96 Fecha de presentación : **15.02.2002**

97 Número de publicación de la solicitud: **1233631**

97 Fecha de publicación de la solicitud: **21.08.2002**

54 Título: **Sistema y procedimiento de gestión de un puerto serie.**

30 Prioridad: **16.02.2001 US 788125**

45 Fecha de publicación de la mención BOPI:
08.07.2011

45 Fecha de la publicación del folleto de la patente:
08.07.2011

73 Titular/es: **Microsoft Corporation**
One Microsoft Way
Redmond, Washington 98052, US

72 Inventor/es: **Sherman, Roman y**
Shell, Scott R.

74 Agente: **Carpintero López, Mario**

ES 2 362 645 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

Sistema y procedimiento de gestión de un puerto serie

Campo de la invención

5 La presente invención se refiere, en general, a un software informático y, más concretamente, a la gestión de un puerto serie.

Antecedentes

10 Las comunicaciones de puerto serie se utilizan habitualmente en los dispositivos informáticos o en los sistemas informáticos. La conexión física con el puerto serie en un dispositivo o sistema informático se establece mediante hardware, como por ejemplo mediante cableado directo, módems, dispositivos de interfaz de red, y otros dispositivos de transmisión. Aunque la conexión física es gestionada por hardware, la conexión entre el puerto serie y una aplicación sobre el dispositivo informático a menudo es manipulada por software.

15 Los programas de aplicación software requieren un identificador procedente del hardware con el fin de establecer una conexión. Una interfaz hardware comunica con el hardware para solicitar el identificador. El hardware devuelve el identificador a la interfaz hardware. La interfaz hardware pasa el identificador al programa de aplicación. Una vez que el programa de aplicación recibe el identificador, el programa de aplicación gestiona el control de la manipulación hasta que la conexión se termina. El programa de aplicación utiliza el identificador para transferir y recibir datos.

El documento EP 0506278 divulga un controlador de dispositivo de almacenamiento en memoria intermedia desde una aplicación hacia una interfaz de puerto.

20 **Sumario**

La presente invención tiene por objeto la provisión de un sistema y un procedimiento para la gestión de un puerto serie.

25 De acuerdo con un aspecto de la invención, el procedimiento de gestión de un puerto serie, proporciona un puerto serie virtual que establece unas conexiones de comunicación de datos entre el puerto serie físico y un programa de aplicación. El puerto serie virtual suspende de forma periódica la conexión física entre la sesión de comunicación de datos para que los mensajes ordenados puedan ser enviados hacia y procesados por el puerto serie físico.

30 De acuerdo con otro aspecto de la invención, el procedimiento de gestión de otro puerto serie incluye un puerto serie virtual que suspende de forma periódica una sesión de comunicación de datos. La sesión de comunicación de datos suspendida es reanudada dentro de un periodo de tiempo determinado mediante un cálculo heurístico. Aunque la sesión de comunicación de datos se suspenda, los mensajes de comando son comunicados al puerto serie.

35 En otro aspecto adicional de la invención, una sesión de comunicación de datos es suspendida durante un periodo de tiempo determinado debido a varios criterios que incluyen la tasa de transmisión de bits del canal de comunicación, la tasa de transmisión de la comunicación del puerto serie, y el cuanto de tiempo. Dado que la tasa de transmisión de comunicación del puerto serie es generalmente mayor que la tasa de transmisión de bits del canal, las sesiones de comunicación de datos pueden ser suspendidas de forma periódica sin suspender la conexión. Mediante la multiplexión entre la sesión de comunicación de datos y el modo comando, pueden llevarse a cabo al mismo tiempo múltiples sesiones.

40 De acuerdo con otro aspecto de la invención, un gestor del puerto serie virtual proporciona la gestión del puerto serie entre los programas de aplicación y el puerto serie físico dentro de un dispositivo electrónico portátil. En un ejemplo, el dispositivo portátil es un dispositivo telefónico inalámbrico, como por ejemplo un teléfono celular. En otro ejemplo, el dispositivo electrónico portátil es un tipo de teléfono celular GSM. En otro ejemplo adicional, el dispositivo electrónico portátil es un tipo de dispositivo de gestor de información personal (PIM). Otros dispositivos electrónicos que operan en un entorno de comandos AT pueden, así mismo, utilizar la presente invención para la gestión del puerto serie.

45 Brevemente descrita, la presente invención proporciona un puerto serie virtual (VSP) que está situado entre un puerto serie de un dispositivo electrónico móvil, unas aplicaciones que requieren una manipulación de conexión del puerto serie, y otras aplicaciones que requieren un acceso en modo comando hacia el puerto serie. Las aplicaciones de comunicación de datos (por ejemplo, la exploración en la Web, e-mail, etc.) conectan el puerto serie a través del VSP. El VSP crea un identificador de conexión virtual que es devuelta a la aplicación. La solicitudes en modo comando (por ejemplo, solicitudes de mensajería cortas) son recibidas por la capa de abstracción de hardware, traducidas en mensajes en modo comando (por ejemplo, comandos de AT) y situadas en la fila de espera. Una memoria intermedia almacena continuamente los datos entrantes mientras la sesión de comunicación de datos está suspendida. El VSP utiliza un cálculo heurístico para determinar el intervalo de tiempo máximo durante el cual puede ser interrumpida la sesión de comunicación de datos. El cálculo heurístico de un cuanto de tiempo (por ejemplo, 0,5

segundos) en base a la relación de una tasa media de transmisión de bits para la comunicación en aire (por ejemplo, 7,000 bits / seg) y una tasa de transmisión de bits para el puerto serie (por ejemplo, 152,000 bits / seg). Esto contribuye a asegurar que las comunicaciones de datos no son interrumpidas durante un intervalo de tiempo lo suficientemente largo como para provocar una pérdida de paquetes de datos. El parámetro del cuanto de tiempo y los otros parámetros pueden ser ajustados de forma estadística y dinámica para responder a un tamaño de memoria intermedia finita de un hardware.

De acuerdo con un aspecto de la presente invención, un sistema de gestión de un puerto incluye una interfaz que establece una conexión de datos con el puerto. Otra interfaz recibe las solicitudes de comando y almacena los mensajes de comando en una memoria intermedia. La interfaz suspenda de forma periódica la conexión de datos establecida con el puerto si los mensajes de comando están situados en la memoria intermedia. La interfaz envía tantos mensajes de comando desde la memoria intermedia hacia el puerto en cuanto puedan ser ejecutados dentro de un periodo de tiempo prescrito y, a continuación, reanuda la conexión con el puerto. El intervalo de tiempo prescrito puede ser calculado mediante un cálculo heurístico.

De acuerdo con otro aspecto de la invención, se dispone un puerto gestionado. El gestor de puerto genera un identificador virtual cuando se establece una conexión de datos. Un cálculo heurístico es utilizado para determinar cuándo establecer la conmutación entre un modo datos y un modo comando. Los datos son encaminados entre el puerto y el identificador virtual cuando el modo operativo es el modo datos. Los mensajes de comando son situados en una memoria intermedia. Diversos mensajes de comando (posiblemente ninguno) existentes en la memoria intermedia son seleccionados. Los mensajes de comando seleccionados procedentes de la memoria intermedia son enviados hacia el puerto cuando el modo operativo es el modo comando.

De acuerdo con otro aspecto adicional de la presente invención se proporciona un medio legible por computadora que incorpora unas instrucciones ejecutables por computadora para gestionar una conexión de puerto. Un modo llamada es activado cuando una conexión de datos se establece. El modo llamada es desactivado cuando se determina la conexión de datos. El cálculo heurístico es utilizado para determinar un intervalo de tiempo. Los datos son encaminados entre una aplicación y el puerto durante el intervalo de tiempo cuando el modo operativo es el modo llamada. La conexión de datos se suspende durante la duración del intervalo de tiempo cuando el modo operativo es el modo llamada. El modo comando es activado, o bien cuando el modo llamada es activado, o cuando la conexión de datos es suspendida. Los mensajes de comando están esperando el procesamiento cuando el modo operativo es el modo comando son, a continuación, procesados. La conexión de datos suspendida se reanuda después de que se ha completado el procesamiento de los mensajes de comando cuando el modo llamada está activo.

Estas y otras características distintivas y ventajas, que caracterizan la presente invención, se pondrán de manifiesto mediante la lectura de la descripción detallada posterior en combinación con los dibujos asociados.

Breve descripción de los dibujos

La FIGURA 1 es un diagrama de bloques funcional de un dispositivo informático adaptado para implementar una forma de realización de la invención;

la FIGURA 2 ilustra un entorno operativo de una forma de realización de la invención;

la FIGURA 3 es un diagrama de bloques funcional de un dispositivo informático móvil ejemplar;

la FIGURA 4 es un diagrama de bloques que ilustra diversos componentes funcionales de un sistema de puerto serie ejemplar;

la FIGURA 5 ilustra el funcionamiento de una aplicación ejemplar del sistema mostrado en la FIGURA 4;

la FIGURA 6 es un diagrama que ilustra diversos componentes de un gestor de dispositivos;

la FIGURA 7 es un diagrama de bloques que ilustra el funcionamiento de un sistema ejemplar que incluye un puerto serie virtual;

la FIGURA 8 es un diagrama de bloques que ilustra diversos componentes funcionales de otro sistema de puerto serie ejemplar;

la FIGURA 9 es un diagrama de bloques funcional de una capa de abstracción de hardware ejemplar;

la FIGURA 10 es un diagrama de flujo que ilustra el funcionamiento de un puerto serie virtual ejemplar;

la FIGURA 11 es un diagrama de flujo que ilustra el procesamiento de unos mensajes de comando de un puerto serie virtual;

la FIGURA 12 es otro diagrama de flujo que ilustra el funcionamiento de otro puerto serie virtual ejemplar, de acuerdo con la presente invención.

Descripción detallada

La presente invención tiene por objeto la provisión de un procedimiento y un sistema para su puesta en interfaz, de forma dinámica, con un puerto serie. Una interfaz de puerto "virtual", o artificial, modera las sesiones de comunicación de datos y / o voz entre diversos programas y dispositivos de aplicación que comunican con y / o a través del puerto serie. En una forma de realización de la invención, el puerto serie virtual es implementado en un dispositivo móvil.

Con referencia a la FIGURA 1, un sistema ejemplar para la implementación de la presente invención incluye un dispositivo informático, como por ejemplo el dispositivo informático 100. En una configuración básica, el dispositivo informático 100 típicamente incluye al menos una unidad de procesamiento 102 y una memoria 104 del sistema. Dependiendo de la exacta configuración y del tipo de dispositivo informático, la memoria 104 del sistema puede ser volátil (como por ejemplo una RAM), no volátil (como por ejemplo una ROM), una memoria instantánea, o una combinación de las dos. La memoria 104 del sistema típicamente incluye un sistema operativo 105, uno o más módulos de programa 106, y puede incluir unos datos de programas 107. Esta configuración básica se ilustra en la FIGURA 1 por los componentes situados dentro de la línea de puntos 108.

El dispositivo informático 100 puede, así mismo, ofrecer algunas características distintivas o una funcionalidad adicionales. Por ejemplo, el dispositivo informático 100 puede, así mismo, incluir unos dispositivos adicionales de almacenamiento de datos (extraíbles y / o no extraíbles) como por ejemplo, discos magnéticos, discos ópticos o una cinta. Dicho almacenamiento adicional se ilustra en la FIGURA 1 mediante el almacenamiento extraíble 109 y el almacenamiento no extraíble 110. Los medios de almacenamiento informáticos pueden incluir medios volátiles y no volátiles, extraíbles y no extraíbles, implementados mediante cualquier procedimiento o sistema técnico para el almacenamiento de información, como por ejemplo, unas instrucciones legibles por computadora, estructuras de datos, módulos de programas u otros datos. La memoria 104 del sistema, el almacenamiento extraíble 109 y el almacenamiento no extraíble 110 son ejemplos de medios de almacenamiento informáticos. Los medios de almacenamiento informáticos incluyen, pero no se limitan a, una RAM, una ROM, una EEPROM, la memoria instantánea u otros medios técnicos de memorias, el CD-ROM, los discos versátiles digitales (DVD) u otro medio técnico que pueda ser utilizado para almacenar la información deseada y al que se pueda acceder mediante el dispositivo informático 100. El dispositivo informático 100 puede, así mismo, incorporar un (unos) dispositivo(s) de entrada 112. Como por ejemplo un teclado, un ratón, un lápiz, un dispositivo de entrada de voz, un dispositivo de entrada táctil. Pueden incluirse, así mismo, un (unos) dispositivo(s) de salida 114, como por ejemplo una pantalla, unos altavoces, una impresora, etc. Todos estos dispositivos son conocidos en la técnica y no necesitan ser analizados *in extenso* en la presente memoria.

El dispositivo informático 100 contiene, así mismo, una(s) conexión(es) entre comunicaciones 116 que permite(n) que el dispositivo comunique con otros dispositivos informáticos 118 como por ejemplo a través de una red. La(s) conexión(es) de comunicaciones 116 es (son) un ejemplo de medios de comunicación. Los medios de comunicación típicamente incorporan unas instrucciones legibles por computadora, unas estructuras de datos, unos módulos de programa u otros datos en una señal de datos modulada, como por ejemplo una onda portadora u otro mecanismo de transporte que incluya cualquier medio de suministro de información. El término "señal de datos modulada" significa una señal que presenta una o más de sus características fijadas o cambiadas de tal manera que codifiquen información en la señal. A modo de ejemplo, y no de limitación, los medios de comunicación incluyen medios cableados, como por ejemplo una red cableada o una conexión de punto a punto, y medios inalámbricos como por ejemplo medios acústicos, de RF, infrarrojos y otros medios inalámbricos. El término medios legibles por computadora, tal y como se utiliza en la presente memoria, incluye tanto medios de almacenamiento como medios de comunicación.

La FIGURA 2 ilustra un ejemplo de un entorno operativo 200 en el cual la invención puede ser implementada en un dispositivo móvil 210. El dispositivo móvil 210 es capaz de comunicar a través de un canal o de unos canales de comunicación 220 con otros dispositivos de comunicación 230.

El entorno operativo 200 mostrado en la FIGURA 2 es solo un ejemplo de un entorno operativo apropiado y no pretende sugerir limitación alguna en cuanto al alcance del uso, o de la funcionalidad de la dirección. Otros sistemas, entornos y / o configuraciones informáticas sobradamente conocidas que pueden ser apropiadas para su uso en la invención incluyen, pero no se limitan a, las computadoras personales, las computadoras de servidor, los dispositivos de mano o portátiles, los sistemas multiprocesadores, los sistemas basados en microprocesadores, los sistemas electrónicos de consumidor programables, los PCs de red, las minicomputadoras, las computadoras de gran sistema, los entornos informáticos distribuidos que incluyen cualquiera de los sistemas o dispositivos expuestos, y similares.

La FIGURA 3 ilustra un ejemplo de un dispositivo informático móvil 300. Dispositivos informáticos móviles ejemplares incluyen, pero no se limitan a: un teléfono móvil, un transceptor de radio inalámbrico, un gestor de información personal (PIM), un organizador electrónico o cualquier otro tipo de dispositivo informático de mano. El dispositivo informático móvil 300 incluye un procesador 360, una memoria 362, una pantalla 328, y un teclado 332. La pantalla 328 puede incluir una pantalla sensible al tacto (pantalla táctil). La memoria 362 incluye en general tanto una memoria volátil, (por ejemplo una RAM) y una memoria no volátil (por ejemplo una ROM), tarjetas de PCMCIA, etc.

Un sistema operativo 364 reside en la memoria 362 y ejecuta en el procesador 360. El dispositivo informático móvil 300 incluye un sistema operativo, como por ejemplo el sistema operativo de Windows® CE de Microsoft Corporation, u otro sistema operativo.

5 Uno o más programas de aplicación 366 son cargados en la memoria 362 y ejecutados por el sistema operativo 364. Ejemplos de programas de aplicación incluyen programas de correo electrónico, programas de planificación, programas de PIM (gestión de información personal), programas de procesamiento de programas, programas de hojas de cálculo, programas de exploración en Internet, y otros. El dispositivo informático móvil 300, incorpora, así mismo, un gestor de notificación 368 cargado en la memoria 362, el cual ejecuta en el procesador 360. El gestor de notificación 368 gestiona las solicitudes de notificación procedentes de las aplicaciones 366.

10 El dispositivo informático móvil 300 incorpora un suministro de potencia 370 que se organiza en una o más baterías. El suministro de potencia 370 puede, así mismo, incluir una fuente de potencia externa que anule o recargue las baterías, como por ejemplo un adaptador de CA o un puerto o cuna de acoplamiento energizado.

15 El dispositivo informático 300 se muestra, así mismo, con tres tipos de mecanismos de notificación externos: un LED 340, un dispositivo vibratorio 372, y un generador de audio 374. Estos dispositivos están directamente acoplados al suministro de potencia 370 de forma que, cuando son activados, permanezca en funcionamiento durante el periodo dictado por el mecanismo de notificación aun cuando el procesador del dispositivo informático móvil y otros componentes puedan cerrarse para cerrar la potencia de la batería. El LED 340, de modo preferente, permanece de modo indefinido hasta que el usuario adopta una medida. Las versiones actuales del dispositivo vibratorio 372 y del generador de audio 374 utilizan demasiada potencia para las baterías del dispositivo informático móvil de hoy en día y, por tanto, están configurados para apagarse cuando el resto del sistema lo hace o en un periodo determinado después de la activación. Una ilustración ejemplar de un sistema telefónico se representa en la FIGURA 4. Los programas de aplicación 410, 420 comunican por medio del HAL 460 con el hardware 470 del puerto serie. Un primer tipo de programa de aplicación 410, aplicación 1, comunica directamente con el HAL 460. Un segundo tipo de programa de aplicación 420 Aplicación 2, comunica con el HAL 460 a través de una pila TCP/IP. En una forma de realización, una interfaz de programa de aplicación (API), como por ejemplo el sistema operativo de Windows® CE de Microsoft Corporation (WINSOCK) es utilizada para comunicar entre el programa de aplicación 420 y el HAL 460. Unos tomacorrientes se disponen a través de una llamada WINSOCK mediante el programa de aplicación. La Aplicación 2 420 comunica con y envía y recibe unas memorias intermedias que son asignadas por la llamada de tomacorrientes. La capa WINSOCK encapsula los mensajes salientes dentro de los mensajes TCP/IP a través de la capa TCP/IP 440, la capa TCP/IP 440 encapsula los mensajes para la comunicación en serie a través de un protocolo de punto a punto (PPP). La capa PPP 450 transfiere los datagramas salientes al HAL 460 para su comunicación a través de hardware 470 del puerto serie.

25 El HAL 460 comunica con el hardware 470 del puerto serie para establecer unas conexiones y encaminar los paquetes de datos entrantes y salientes. El hardware 470 de puerto serie transmite y recibe paquetes de datos desde un canal de comunicación 480. EL canal de comunicación 480 puede ser cualquier interfaz de comunicación apropiada, como por ejemplo, un canal de comunicaciones inalámbricas utilizado en comunicaciones celulares.

El HAL 460 incluye un nivel de interfaz de radio 462, RIL, y un puerto serie virtual 464, VSP. En una forma de realización de la invención, el RIL 462 y el VSP 464 están integrados en un programa de interfaz común. En otra forma de realización de la invención, el RIL 462 y el VSP 464 son programas de interfaz separados.

40 El funcionamiento de un programa de aplicación ejemplar que establece una conexión y comunica por medio del HAL se muestra en la FIGURA 5. El procesamiento empieza en el bloque 510 donde el programa de aplicación envía una solicitud para una conexión de puerto serie a través de una interfaz de programa de aplicación, como por ejemplo un comando abierto de WINSOCK. El procesamiento avanza hasta el bloque 520 donde el programa de aplicación espera la confirmación de la conexión abierta (espera una manipulación virtual).

45 En el bloque 530, el procesamiento determina si se ha recibido un identificador. El procedimiento continuará desde el bloque 530 hasta el bloque 520 hasta que se determine que ha sido recibido un identificador. Una vez que se ha recibido un identificador, el procesamiento avanza desde el bloque 530 hasta el bloque 540 donde la conexión es supervisada.

50 Mientras la conexión es supervisada, el procesamiento avanza hasta el bloque 550 donde se determina si la conexión está activa. Si la conexión no está activa, el procesamiento avanza hasta el bloque 560, en otro caso, el procedimiento avanza hasta el bloque 570. En el bloque 560, se determina si el proceso ha finalizado. Si el procedimiento no ha finalizado, entonces el procesamiento retrocede hasta el bloque 540. En otro caso, el procesamiento avanza hasta el bloque 570.

55 En el bloque 570, se determina si ha terminado la comunicación de los datos. La conexión del puerto puede terminar mediante una solicitud efectuada por el programa de aplicación. Así mismo, la conexión del puerto puede terminar en base a otros criterios, como por ejemplo un estado de fin de temporización detectado en el bloque 560. Si las comunicaciones de datos no han terminado, entonces el procesamiento continúa hasta el bloque 580 donde los datos son enviados y / o recibidos por el programa de aplicación según las necesidades.

De acuerdo con lo descrito con anterioridad, las comunicaciones de datos se establecen, en general, mediante la solicitud de un puerto serie de comunicaciones mediante la apertura de una nueva llamada de datos y la obtención de un identificador del puerto serie en respuesta a la solicitud de puerto. Después de que se ha recibido el identificador del puerto serie todos los datos son transmitidos hacia y recibidos del identificador del puerto serie. Al final de la comunicación de los datos, la llamada se termina.

Las llamadas telefónicas de voz se establecen utilizando un modo comando para iniciar la llamada (por ejemplo, un comando ATD). Después de que se ha establecido la conexión, la llamada telefónica de voz comunica directamente con la conexión física sin utilizar ningún recurso de puerto serie. Sin embargo, otros sistemas telefónicos de voz pueden comunicar utilizando un protocolo basado en Internet (por ejemplo, un teléfono de web). Las aplicaciones telefónicas basadas en Internet (o en la web) se conectarán con el puerto serie de manera similar a las conexiones de comunicación de datos descritas con anterioridad.

La FIGURA 6 ilustra un sistema gestor de dispositivos 600 que contiene un conjunto de controladores y servicios de dispositivos. En general, los controladores de dispositivos consisten en unas rutinas que son ejecutadas en varias etapas en una solicitud de entrada / salida (I / O). Los servicios pueden ser solicitados mediante un programa de aplicación a través de una llamada de sistema operativo (OS). Una llamada de interfaz de programa de aplicación (API) es redirigida al gestor de dispositivos donde el gestor de dispositivos lleva a cabo las diversas operaciones del dispositivo. Para el hardware de puerto serie descrito con anterioridad, el gestor de dispositivos lleva a cabo las operaciones requeridas para establecer las conexiones, cerrar las conexiones y encaminar la información hacia y desde el puerto serie.

Comenzando en el gestor de dispositivos 605, se proporciona una rutina INITIALIZE 610. La rutina inicializar inicializa el controlador del VSP. La rutina INITIALIZE 610 registra las rutinas asociadas con el controlador de dispositivos con el sistema de I / O del sistema operativo. Una rutina DE - INITIALIZE 620 es invocada cuando el controlador de dispositivos es descargado por el sistema operativo.

El gestor de dispositivos inicia una rutina OPEN 630 en respuesta a una solicitud de API para abrir una nueva conexión de puerto serie. En un ejemplo de la presente invención, la rutina OPEN 630 devolverá un identificador al gestor de dispositivos cuando una llamada ya esté en curso. En este ejemplo, la rutina OPEN falla y no devolverá un identificador a menos que una llamada esté en curso, dado que el VSP no puede ser abierto sin un canal de datos válido. Programa de aplicación inicia una solicitud de VSP para cerrar la conexión cuando ya no se requiera la conexión actualmente abierta. El gestor de dispositivos responde a la solicitud de API iniciando la rutina CLOSE 640, lo cual libera el identificador que estaba anteriormente en curso mediante el programa de aplicación.

El gestor de dispositivos inicia una rutina WRITE 650 y una rutina READ 660 en respuesta a las solicitudes de API para leer y escribir hacia un identificador específico. En un ejemplo de la presente invención, las solicitudes de leer y escribir son encaminadas hacia el puerto serie físico a menos que el puerto serie físico no esté disponible. Cuando el puerto serie físico no esté disponible, las solicitudes de leer y escribir son bloqueadas. El puerto serie físico no está disponible cuando se está llevando a cabo una solicitud de modo comando. Ejemplos de solicitudes de modo comando incluyen, pero no se limitan a , una solicitud para: verificar la intensidad de la señal de una conexión, verificar el estado de la red, llevar a cabo una operación de red, y otras.

El gestor de dispositivos inicia una rutina IO CONTROL 670 en respuesta a una solicitud de la API para el control de I / O que es dirigida a un identificador del VSP. Cuando el puerto serie está en un modo datos, la rutina IO CONTROL 670 dirige la solicitud de control de I / O al puerto serie físico. Sin embargo, cuando el puerto serie es conmutado al modo comando, la solicitud de control de I / O es bloqueada hasta que el puerto serie es conmutado de nuevo al modo datos. Tal y como se expondrá más adelante, el ajuste del terminal de datos listo (DTRi) al nivel débil en el puerto serie físico, conmutará una conexión de datos actual con el puerto serie del modo datos al modo comando. Las operaciones de control de I / O que intentan modificar el estado de la línea DTR del puerto serie son falladas por el VSP para que se impida que los módulos externos conmuten el puerto externo físico fuera del modo datos.

El gestor de dispositivos establece una correspondencia de la llamada API con las rutinas INITIALIZE, DE - INITIALIZE, OPEN, CLOSE, READ, WRITE e IO CONTROL. En un ejemplo de la invención, el VSP solo es accesible mediante programas de aplicación cuando una aplicación de datos está en curso. En este ejemplo, las rutinas OPEN, CLOSE, READ, WRITE e IO CONTROL fallarán hasta que se establezca una llamada de datos. Después de que se ha establecido una llamada de datos, las rutinas (y las correspondientes llamadas API del gestor de datos) comenzarán a tener un resultado positivo. El VSP actúa como un multiplexor que conmuta una conexión entre el puerto serie físico del dispositivo electrónico móvil y las diversas aplicaciones que operan en al menos dos modos. Las aplicaciones que están enviando flujos de comunicación de datos, como por ejemplo la exploración en Internet, comunicación de voz, y tipos de aplicaciones de correo electrónico, están operando en un modo datos. Las aplicaciones que solicitan información del sistema hardware o de la red, como por ejemplo la intensidad de la señal de un teléfono móvil, el estado de la red, y otras solicitudes relacionadas con el hardware, están operando en un modo comando. Tal y como se expuso con anterioridad, un dispositivo móvil inalámbrico incluye un HAL que incluye tanto un RIL como un VSP. Una panorámica de las operaciones de procesamiento ejemplares para los mensajes de modo comando se muestra en la FIGURA 7.

Tal y como se muestra en la FIGURA 7, los mensajes comando son enviados al HAL mediante un programa de aplicación que envía una solicitud de modo comando 710 a través de una llamada API (REQUEST). El RIL del HAL recibe la solicitud de modo comando 720 y traduce la solicitud 722 a otro formato (por ejemplo, el formato Comando de AT) que es apropiado para el hardware de puerto serie (o para el controlador de hardware). El RIL envía el mensaje de comando al VSP, ya sea como un mensaje directo o mediante otros medios, como por ejemplo una memoria intermedia una fila de espera u otro área de memoria almacenada. En una forma de realización, el RIL y el VSP, comparten una fila de espera a la que se accede en base a al menos un criterio, como por ejemplo la prioridad, la solicitud más reciente, así como a otros criterios. El VSP recibe el mensaje de comando 730 y procesa el mensaje en base a la actividad actual del puerto serie.

5
10
15

Cuando actualmente no hay una llamada activa (CALL INACTIVE), el VSP envía el mensaje de comando formateado directamente al puerto serie 736 y espera una contestación (cuando se requiera) del puerto serie. Sin embargo, el VSP no envía inmediatamente el mensaje de comando al puerto serie cuando una llamada está activa (CALL ACTIVE). Por el contrario, el VSP utiliza un cálculo heurístico (732 CALCULATE HERUSTIC) para determinar si la llamada actual puede o no ser interrumpida el tiempo suficiente para gestionar el mensaje de comando. Si es así, la llamada activa actual es suspendida (SUSPEND CALL) y el mensaje de comando es enviado al puerto serie 736.

En caso contrario, entonces el VSP espera 734 hasta que esté disponible un tiempo de suspensión suficiente para procesar el mensaje de comando antes de suspender la llamada activa actual.

20

El VSP espera un mensaje de contestación del puerto serie si es apropiado. El VSP recibe un mensaje de contestación 738 después de que el puerto serie envíe el mensaje de contestación (SEND REPLY). A continuación el VSP envía el mensaje de contestación al RIL (SEND REPLY) y reanuda cualquier llamada anteriormente suspendida (RESUME CALL). El RIL recibe el mensaje de contestación 724 (RIL RECEIVES REPLY) del VSP y traduce la contestación a otro formato 726 (RIL TRANSLATES REPLY). El RIL envía de nuevo el mensaje traducido al programa de aplicación (API REPLY) donde es recibido 712 (APLICATTION RECEIVES REPLY).

25
30

Como se expuso con anterioridad, una llamada es suspendida por el VSP en base a un cálculo heurístico. El RIL solicita al VSP que interrumpa temporalmente la comunicación de datos (por ejemplo, suspenda la llamada para que el RIL o el VSP puedan llevar a cabo diversas operaciones con el puerto serie (por ejemplo, operaciones de redes celulares). En un ejemplo, la interrupción de la comunicación de datos se consigue mediante la conmutación del puerto serie físico en un modo comando mediante el ajuste en un nivel débil de la línea DTR. Una vez en el modo comando, las solicitudes, como por ejemplo los comandos AT pueden ser enviados al puerto serie. Las contestaciones procedentes del puerto serie son entonces recibidas desde el puerto serie en un formato, como por ejemplo respuestas y / o notificaciones AT. Después de que se ha completado la solicitud en modo comando del puerto serie, el puerto serie físico es devuelto al modo comunicación de datos mediante la ejecución del mensaje de comando como por ejemplo un ATO.

35
40

Dado que el puerto serie físico es conmutado al modo comando mediante el ajuste débil de la línea DTR, el gestor del IO CONTROL no puede ser autorizado a manipular la línea DTR. Mediante la inhabilitación del acceso a la línea DTR, el VSP queda libre para operar adecuadamente. Las comunicaciones de datos a través del VSP no pueden ser interrumpidas durante largos intervalos de tiempo, dado que las memorias intermedias de recibir y enviar son de tamaño limitado. En un tipo de dispositivo de teléfono celular, las memorias intermedias de enviar y recibir son parte del hardware del teléfono celular. Con el fin de impedir la pérdida de datos, la conexión de comunicación de datos puede ser interrumpida durante cortos periodos de tiempo de acuerdo con lo determinado mediante un cálculo heurístico. El VSP utiliza la heurística siguiente para determinar el tiempo máximo durante el cual el enlace de datos puede ser interrumpido:

$$\text{MaxInterruptTime} = \text{TimeQuantum} \times \left[1 - \frac{\text{EffectiveAirBitRate}}{\text{SoftwareToModuleBitTRate}} \right]$$

45

en la que

MaxInterruptTime es el tiempo máximo durante el cual las comunicaciones de datos pueden ser interrumpidas;

TimeQuantum es el cuanto de tiempo tomado como base del cálculo;

50

EffectiveAirBitRate es la tasa de transmisión de bits media a la cual los datos son enviados a través del aire (sobre el último *TimeQuantum*); y

SoftwareToModuleBitRate es la tasa de transmisión de bits a la cual los datos son comunicados al módulo celular.

La clave de la heurística es el hecho de que la *SoftwareToModuleBitRate* es generalmente mucho más alta que la *EffectiveAirBitRate*. Por ejemplo, las conexiones habituales de puerto serie pueden transmitir y recibir datos a 152,000 bits / segundo. Sin embargo, el flujo de datos habitual de las comunicaciones por ondas aéreas es de aproximadamente 7,000 bits / segundo. De esta manera, la *SoftwareToModuleBitRate* de 152,000 bits / segundo es sustancialmente más alta que la *EffectiveAirRate* de 7,000 bits / segundo. El puerto serie físico está predominantemente inactivo a través de cada *TimeQuantum*, dado que aproximadamente un 5% del tiempo es utilizado para la comunicación de datos. De esta manera, en este ejemplo, aproximadamente el 95% del *TimeQuantum* está disponible para mensajes de comando para interrumpir la comunicación de datos. En un ejemplo de la presente invención, se utiliza un *TimeQuantum* de 0,5 segundos.

Otro ejemplo de un sistema que utiliza un puerto serie virtual se muestra en la FIGURA 8. Tal y como se muestra en la figura, los programas de aplicación comunican a través de una API al RIL. Más concretamente, el primer programa de aplicación 810 (Aplicación 1) comunica con el RIL a través de diversas APIs, como por ejemplo una API telefónica (TAPI), una API telefónica extendida (ExTAPI), y otras. Las conexiones TAPI son comunicadas a través de un controlador ExTAPI 820, el cual, a su vez, comunica con un controlador TAPI 832 al RIL 840. Otros tipos de mensajes son comunicados mediante controladores específicos (por ejemplo, un controlador de SMS, un controlador de SIM y otros) al RIL 840. Otra aplicación 821 (Aplicación 2) comunica por medio de una pila TCP / IP 822, la cual, a su vez, comunica con el controlador TAPI 832 hasta el RIL 840.

De acuerdo con lo expuesto con anterioridad, el RIL 840 comunica con el VSP 850 a través de unos mensajes de comando, como por ejemplo comandos AT. El VSP 850 proporciona o bien un mensaje de comando (por ejemplo comandos AT) hacia el puerto serie 860 (COM1) del sistema o bien transmite datos hacia y desde una aplicación hasta el puerto serie 860 del sistema.

La primera aplicación 810 llama a una API para recuperar información del puerto serie. Un módulo intermedio (por ejemplo una TAPI) traduce dicha información en otro formato que sea apropiado para el RIL 840. El RIL 840 traduce la solicitud en un mensaje de comando apropiado para el puerto serie 860, y envía el mensaje de comando al VSP 850. Cuando una llamada no está actualmente en curso, el mensaje de comando es enviado directamente desde el VSP 850 al puerto serie 860. El Puerto Serie 860 del Sistema ejecuta el comando y emite una respuesta que es propagada de nuevo a la aplicación de solicitud 810 el orden inverso.

Por ejemplo, la Aplicación 1, llama al `lineGetCurrentOperator()` de la API ExTAPI para recuperar el nombre del operador inalámbrico de un teléfono celular. La interfaz ExTAPI traduce esta llamada a una `lineDevSpecific()` de la API TAPI. La interfaz TAPI traduce esto en una llamada al `RIL_GetOperatorName()` de la API del RIL. El RIL traduce esto en un `AT + COPS?` de Comando AT, el cual es enviado a través del VSP al Puerto Serie del Sistema.

Cuando la Aplicación 2 812 llama a una API para establecer una comunicación de datos, una llamada API para abrir una conexión TCP / IP es propagada hasta el RIL a través de la pila TCP / IP 822 y la interfaz TAPI 832. El RIL envía un comando al VSP solicitando una conexión, a la cual, el VSP devuelve un identificador virtual que es propagado de nuevo hasta la aplicación 812. Una vez que el identificador virtual es recibido por la aplicación 812 se establece un enlace de datos entre el VSP 850 y la aplicación 812. La aplicación 812 puede entonces empezar a leer y escribir datos hacia el identificador virtual sin interrupción por parte del RIL. Si la Aplicación 1 810 solicita información del puerto serie mientras el enlace de datos está activo, la solicitud de información fluye desde la aplicación hasta el VSP 850, de acuerdo con lo descrito con anterioridad. Sin embargo, después de que el VSP recibe la solicitud, lleva a cabo el cálculo heurístico descrito con anterioridad para determinar cuándo enviar la solicitud al puerto serie. En el intervalo de tiempo apropiado la línea DTR se ajusta débil para inhabilitar temporalmente la conexión del puerto serie, y el mensaje de comando (por ejemplo `AT + COPS?`) es procesado. Después de que el VSP envía la contestación (completando de esta forma el comando) desde el puerto serie hasta el RIL, el puerto serie se vuelve a conectar al enlace de datos mediante el envío de un comando apropiado hasta el puerto serie (por ejemplo `ATO`).

De acuerdo con lo expuesto con anterioridad, una fila de espera puede ser utilizada para comunicar entre el RIL y el VSP. La FIGURA 9 ilustra una implementación ejemplar del HAL que incluye filas de espera de mensajería. Son, así mismo, posibles otras filas de espera de mensajes y se incluyen en el ámbito del experto en la materia.

Tal y como se muestra en la FIGURA 9, el HAL900 incluye un RIL 910, una primera fila de espera Q1, 920, una segunda fila de espera Q2, 930 y un VSP 940. El RIL 910 envía unos mensajes de comando a la primera fila de espera 920. El VSP periódicamente recupera un mensaje de comando de la primera fila de espera 920 para su procesamiento. El mensaje de comando seleccionado es enviado al puerto serie (no mostrado). El puerto serie envía un mensaje de contestación al VSP 940 que es posteriormente enviado a la segunda fila de espera 930. El RIL 910 recupera el mensaje de contestación de la segunda fila de espera 930.

Un ejemplo del procesamiento en el VSP 1000 se muestra en la FIGURA 10. En el bloque 1010, el VSP calcula una heurística en base a una información, de acuerdo con lo expuesto con anterioridad. La heurística indica un tiempo de interrupción máximo para la conexión de comunicación de datos. Como alternativa, la heurística puede ser calculada una vez cuando la conexión se establezca en primer término, o en otros momentos en los que pueda ser necesario en base a la carga de la red. A continuación, el VSP determina si hay cualquier dato que deba ser enviado

al puerto serie en el bloque 1020. Cuando los datos que se corresponden con un identificador virtual deben ser encaminados, el procesamiento avanza hasta el bloque 1025 donde el VSP encamina un bloque (o múltiples bloques) de los datos hacia el puerto serie. Como alternativa, el procesamiento avanza hasta el bloque 1030.

5 En el bloque 1030, el VSP determina si hay cualquier dato que debe ser recibido desde el puerto serie. Si los datos están esperando recepción en el puerto serie, entonces el procesamiento avanza hasta el bloque 1035 donde los datos existentes en el puerto serie son encaminados hasta el identificador virtual. Como alternativa, el procesamiento avanza hasta el bloque 1040. El procesamiento, así mismo, avanza hasta el bloque 1040 desde los bloques 1025 y 1035.

10 En el bloque 1040, el VSP examina la primera fila de espera (Q1) para determinar si un mensaje de comando está esperando su procesamiento. Si los mensajes de comando están en la primera fila de espera, entonces el procesamiento avanza hasta el bloque 1050, donde al menos un mensaje de comando es procesado. Como alternativa, el procesamiento continúa hasta el bloque 1010. El procesamiento, así mismo, continúa hasta el bloque 1010 desde el bloque 1050.

15 La FIGURA 11 ilustra un ejemplo de procesamiento de mensajes de comando 1100 para el bloque 1050 de la FIGURA 10. Tal y como se muestra en la FIGURA 11, la conexión de comunicación de datos con el puerto serie es suspendida en el bloque 1110. Cuando no hay ninguna conexión activa, este bloque 1110 es innecesario. Avanzando hasta el bloque 1120, un mensaje de comando es recuperado de la primera fila de espera (Q1). Cuando la fila de espera tiene múltiples mensajes contenidos en ella, puede ser utilizado cualquier criterio razonable para seleccionar uno o más de los mensajes. Por ejemplo, en una forma de realización de la invención, una fila de espera en base a la prioridad es utilizada cuando la prioridad más alta es procesada en primer término.

20 Avanzando hasta el bloque 1130, el mensaje de comando seleccionado es enviado al puerto serie por el VSP. El VSP entonces espera un mensaje de contestación del puerto serie, el cual es recibido en el bloque 1140. El mensaje de contestación es entonces enviado a la segunda fila de espera (Q2) en el bloque 1150. Finalmente, en el bloque 1160, la conexión de comunicación de datos con el puerto serie se reanuda. Cuando no hay ninguna conexión activa, la etapa 1160 es innecesaria.

30 Otro ejemplo de procesamiento del VSP 1200 se muestra en la FIGURA 12. El procesamiento 1200 es ejecutado una vez cada cuanto de tiempo. En este ejemplo, el puerto serie virtual determina si la fila de espera de mensajes de comando está vacía en el bloque 1210. Si la fila de espera de mensajes de comando está vacía continuamos el procesamiento en el bloque 1220; en otro caso, avanzamos hasta el bloque 1270. En el bloque 1220 el cálculo heurístico expuesto con anterioridad es ejecutado para determinar el intervalo de tiempo máximo durante el cual puede ser suspendida una conexión de datos.

35 Avanzando desde el bloque 1220 hasta el bloque 1230, la fila de espera de comandos es analizada para estimar y seleccionar los comandos de la fila de espera que pueden ser procesados durante el intervalo de tiempo de suspensión. Para determinar la cantidad de tiempo que se invierte para cada comando de la cola que va a ser procesado, puede llevarse a cabo otro cálculo en base al historial estadístico basado. Como alternativa una tabla de consulta puede ser utilizada cuando cada entrada de la tabla de consulta se corresponda con un tiempo de procesamiento medio para un comando determinado. Utilizando el máximo intervalo de tiempo de suspensión desde el bloque 1220 junto con las estimaciones del tiempo de procesamiento de los comandos, los comandos son seleccionados para su procesamiento desde la fila de espera de comandos. El procesamiento avanza desde el bloque 1230 hasta el bloque 1240.

La conexión con el puerto serie es suspendida en el bloque 1240, permitiendo que los mensajes de comando seleccionados del bloque 1230 sean procesados en el bloque 1250. Después de que los mensajes de comando seleccionados han sido completamente procesados, el procesamiento continúa en el bloque 1260 donde se reanuda la conexión del puerto serie. Desde el bloque 1260 el procesamiento avanza hasta el bloque 1270.

45 En el bloque 1270, la totalidad del cuanto de tiempo restante es utilizado para procesar los datos necesarios de enviar y recibir para el (los) programa(s) de aplicación. Dado que el procesamiento avanza desde el bloque 1210 hasta el bloque 1270 cuando la fila de espera está vacía, la totalidad del cuanto de tiempo puede ser disponible para manejar los datos de enviar y recibir entre el (los) programa(s) de aplicación y el puerto serie. El procesamiento avanza desde el bloque 1270 hasta el bloque 1280.

50 En el bloque 1280, la información estadística para el VSP es actualizada. Los datos estadísticos pueden incluir la información correspondiente a la cantidad de tiempo que cada comando que requiera para su procesamiento, la información relacionada con los cálculos heurísticos y los tiempos de procesamiento, así como otras estadísticas.

55 Es posible que ninguno de los comandos de la fila de espera pueda ser ejecutado dentro de un periodo de tiempo determinado (por ejemplo, si la fila de espera de comandos contiene un solo comando y ejecutar ese comando llevaría más tiempo que el intervalo de tiempo de suspensión máximo desde el bloque 1220). En este caso, no puede ser procesado ningún comando (esto es, los bloques 1240, 1250 y 1260 se saltan). Así mismo, en este caso, para asegurar que cada comando es finalmente procesado, al menos un comando de la cola de comandos es envejecido. El envejecimiento puede llevarse a cabo mediante la reducción gradual del tiempo de ejecución

esperado del comando, así como mediante otros procedimientos. Una vez que la edad de un mensaje de comando concreto alcanza una edad apropiada, el mensaje debe ser procesado aun cuando pueda no haber sido completado en el intervalo de tiempo de suspensión máximo calculado. De esta manera cada mensaje de comando es procesado.

5 El puerto serie virtual descrito con anterioridad es útil en dispositivos electrónicos móviles que incorporan múltiples funciones, como por ejemplo comunicaciones telefónicas y otras aplicaciones basadas en Internet. Por ejemplo, la exploración de la web, las aplicaciones de correo electrónico y otras que comunican utilizando el TCP / IP pueden ser ofrecidas en un dispositivo electrónico móvil. Los mensajes de comando periódicos son comunicados al puerto serie en el dispositivo electrónico móvil mediante la suspensión de la comunicación de datos durante un periodo de tiempo que se basa en un cálculo heurístico. Mediante la multiplexión entre las conexiones de comunicación activas en el modo datos, y un modo comando, pueden llevarse a cabo múltiples funciones en el mismo dispositivo móvil sin terminar la sesión de comunicación de datos. La técnica heurística ayuda a asegurar que la sesión de comunicación de datos no se interrumpa durante un periodo de tiempo largo lo que provocaría paquetes de datos perdidos debido al desbordamiento de la memoria intermedia. El parámetro del cuanto de tiempo y otros parámetros pueden ser ajustados para responder al tamaño limitado de la memoria intermedia existente en el puerto serie físico.

20

25

30

35

REIVINDICACIONES

- 1.- Un sistema de gestión de un puerto que comprende una primera interfaz adaptada para establecer una conexión de datos con el puerto,
- 5 una segunda interfaz adaptada para recibir solicitudes de comando desde un programa de aplicación (366; 410; 420; 810; 812) y para almacenar los mensajes de comando en una memoria intermedia, en el que los mensajes de comando se corresponden con las solicitudes de comando traducidas en un comando apropiado para el hardware de puerto (470), y
- 10 estando la primera interfaz adaptada para suspender periódicamente la conexión de datos establecida con el puerto cuando los mensajes de comando están situados en la memoria intermedia, en el que la primera interfaz está adaptada para enviar de forma selectiva mensajes de comando desde la memoria intermedia hacia el puerto y para reanudar la conexión con el puerto dentro de un intervalo de tiempo prescrito, en el que el intervalo de tiempo prescrito se determina mediante un cálculo heurístico por medio de la primera interfaz.
- 2.- Un sistema de acuerdo con la Reivindicación 1, en el que el cálculo heurístico se basa en al menos una tasa de transmisión de bits de canal de comunicación, una tasa de transmisión de comunicación de puerto serie, y en un cuanto de tiempo, en el que el cuanto de tiempo es el cuanto de tiempo tomado como base del cálculo.
- 15 3.- Un sistema de acuerdo con la Reivindicación 1, en el que el cálculo heurístico incluye un cuanto de tiempo que es ajustado de forma dinámica, en el que el cuanto de tiempo es el cuanto de tiempo tomado como base del cálculo.
4. Un sistema de acuerdo con la reivindicación 1, en el que el cálculo heurístico es calculado al menos durante un estado energizado.
- 20 5.- Un sistema de acuerdo con la reivindicación 1, en el que el cálculo heurístico es calculado en un intervalo de tiempo predeterminado.
- 6.- Un sistema de acuerdo con la reivindicación 1, en el que la primera interfaz está adaptada para seleccionar unos mensajes de comando desde la memoria intermedia en base a una prioridad ponderada que está asociada con cada entrada existente en la memoria intermedia, o en un criterio de primero en entrar primero en salir, y en un criterio de último en entrar primero en salir.
- 25 7.- Un sistema de acuerdo con la reivindicación 1, en el que el puerto serie incluye una memoria intermedia receptora con un tamaño finito, estando la memoria intermedia receptora adaptada para recibir datos mientras que la conexión de datos está suspendida, y el cálculo heurístico responde al tamaño finito de la memoria intermedia receptora.
- 30 8.- Un sistema de acuerdo con la reivindicación 1, en el que la primera interfaz está adaptada para generar un identificador virtual hacia el puerto cuando la conexión de datos está establecida, y la comunicación de datos se lleva a cabo a través del identificador virtual.
- 9.- Un sistema de acuerdo con la reivindicación 1, en el que la conexión de datos es suspendida mediante el ajuste de un control de DTR de nivel débil en el puerto, y la conexión de datos es reanudada mediante el envío de un mensaje de comando hacia el puerto.
- 35 10.- Un procedimiento de gestión de un puerto que comprende:
- la generación de un identificador virtual cuando está establecida la comunicación de datos;
- la conmutación entre un modo datos y un modo comando en base a un cálculo heurístico;
- el encaminamiento de datos entre el puerto y el identificador virtual cuando el modo operativo es el modo datos;
- 40 la colocación de los mensajes de comando recibidos en una memoria intermedia;
- la selección de al menos un mensaje de comando de la memoria intermedia; y
- el envío del seleccionado al menos un mensaje de comando hasta el puerto cuando el modo operativo es el modo comando.
- 11.- EL procedimiento de la reivindicación 10, en el que la conexión de datos es activa cuando el modo operativo es un modo datos, y la conexión de datos es suspendida cuando el modo operativo es un modo datos.
- 45 12.- El procedimiento de la reivindicación 10, que comprende así mismo:
- la recepción de una solicitud de comando desde una aplicación; y
- el formateo de la solicitud de comando en un mensaje de comando.

- 13.- El procedimiento de la reivindicación 10, que comprende así mismo:
la recepción de un mensaje de contestación desde el puerto cuando el modo operativo es el modo comando; y
el envío del mensaje de contestación hasta una aplicación cuando el modo operativo es el modo comando.
- 14.- El procedimiento de la reivindicación 10, que comprende así mismo:
5 la recepción de un mensaje de contestación proveniente del puerto cuando el modo operativo es el modo comando;
el envío de mensaje de contestación a otra memoria intermedia cuando el modo operativo es el modo comando; y
la conmutación desde el modo comando hasta el modo datos después de que el mensaje de contestación es enviado a la otra memoria intermedia.
- 15.- El procedimiento de la reivindicación 10, en el que el puerto es parte de un dispositivo electrónico portátil.
- 10 16.- Un medio legible por computadora que incorpora unas instrucciones ejecutables por computadora para gestionar una conexión con el puerto, que comprende:
la activación de un modo llamada cuando está establecida la conexión de datos;
la desactivación del modo llamada cuando la conexión de datos se termina;
la determinación de un intervalo de tiempo en base a un cálculo heurístico;
- 15 el encaminamiento de datos entre una aplicación y el puerto durante el intervalo de tiempo cuando el modo operativo es el modo llamada;
la suspensión de la conexión de datos al final del intervalo de tiempo cuando el modo operativo es el modo llamada;
la activación de un modo comando cuando el modo llamada es desactivado;
la activación del modo comando cuando la conexión de datos es suspendida;
- 20 el procesamiento de los mensajes comando que están esperando su procesamiento cuando el modo operativo es el modo comando; y
la reanudación de una conexión de datos suspendida después de la culminación de procesamiento de los mensajes comando cuando el modo llamada está activo.
- 25 17.- Un medio legible por computadora de acuerdo con la reivindicación 16, en el que el cálculo heurístico se basa en al menos un criterio entre una velocidad de transferencia de datos asociada con el puerto, una velocidad de canal de comunicación, una longitud de memoria intermedia asociada con el puerto, y un cuanto de tiempo, en el que el cuanto de tiempo es el cuanto de tiempo tomado como base del cálculo.
- 30 18.- Un medio legible por computadora, de acuerdo con la reivindicación 17, en el que el intervalo de tiempo máximo durante el cual puede estar suspendida una llamada activa está asociado con una relación entre la velocidad de transferencia de datos y la velocidad de canal de comunicación.
- 19.- Un sistema de gestión de puerto serie (860) en un dispositivo portátil, que comprende:
unos medios para la activación de un modo llamada cuando una conexión de datos está establecida;
unos medios para la desactivación del modo llamada cuando la conexión de datos se ha terminado;
unos medios para la recepción de una solicitud de comando;
- 35 unos medios para la conversión de la solicitud de comando en un mensaje de comando;
unos medios para el envío del mensaje de comando hacia el puerto serie cuando opera en un modo comando;
unos medios para el encaminamiento de datos entre una aplicación y el puerto serie (860) cuando opera en un modo datos;
unos medios para la selección del modo datos cuando el modo comando está inactivo;
- 40 unos medios para la selección del modo comando cuando el modo llamada está activo; y
unos medios para la conmutación entre el modo datos y el modo comando en base a un cálculo heurístico, cuando el modo llamada está activo.

20.- Un sistema de acuerdo con la reivindicación 19, en el que los medios de conmutación comprenden así mismo:

unos medios para la suspensión de datos cuando el modo llamada está activo y el modo comando es seleccionado;
y

- 5 unos medios para la reanudación de la conexión de datos suspendida cuando el modo datos es seleccionado, donde el cálculo heurístico determina un intervalo de tiempo máximo durante el cual la conexión de datos puede ser suspendida en base a una relación entre una tasa de transmisión de bits de canal de comunicación y una tasa de comunicación de puerto serie.

10

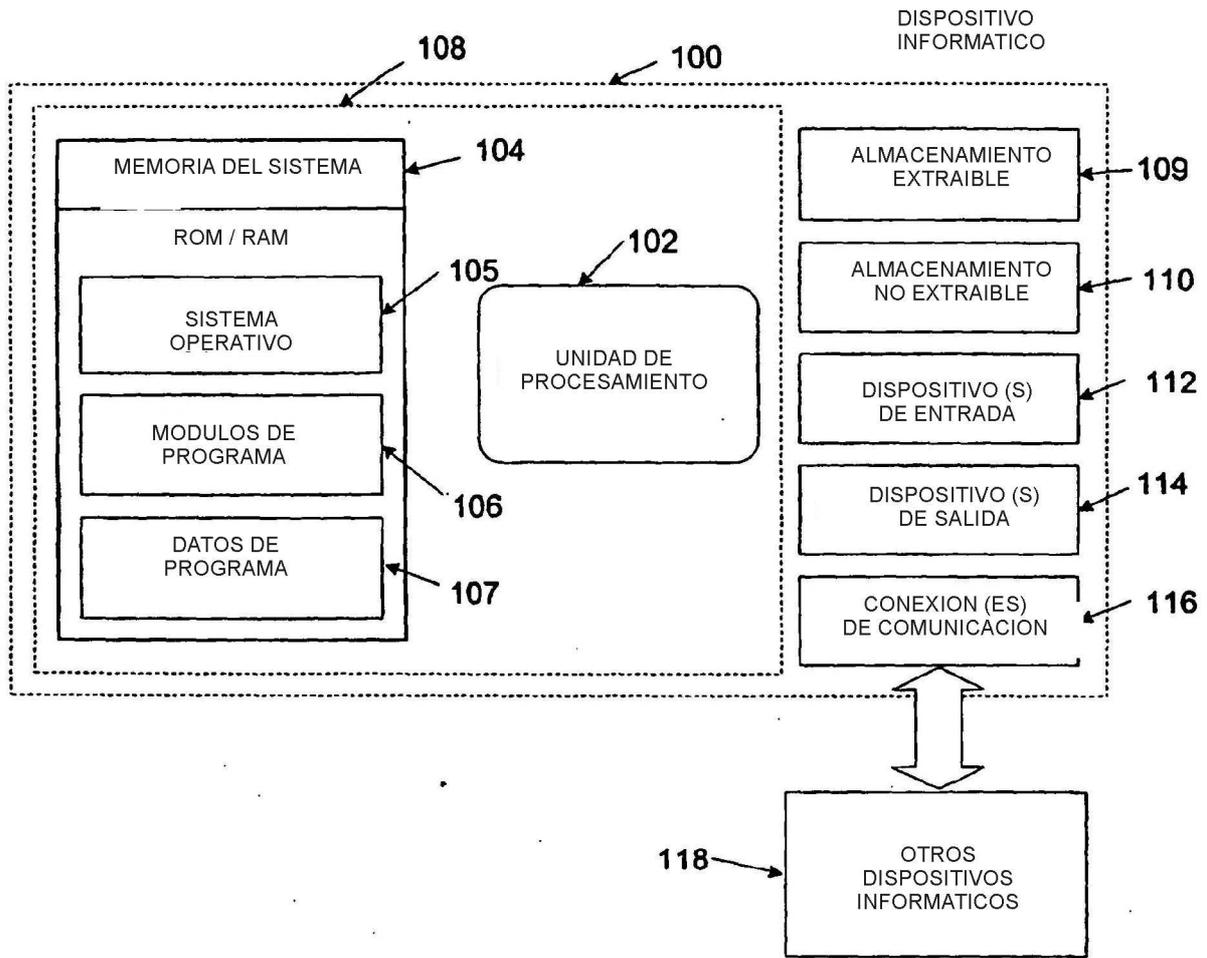


Fig.1

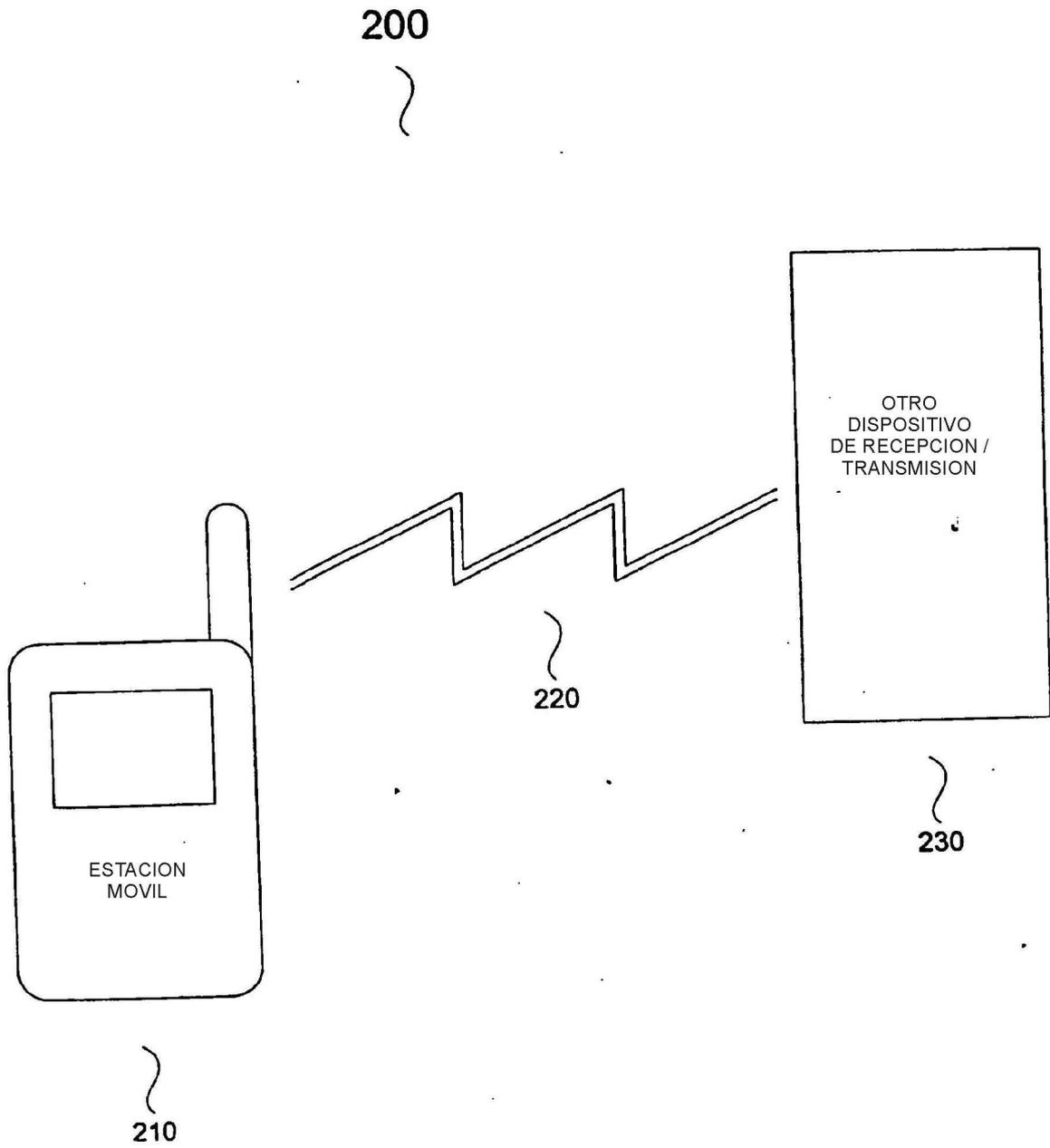


FIG. 2

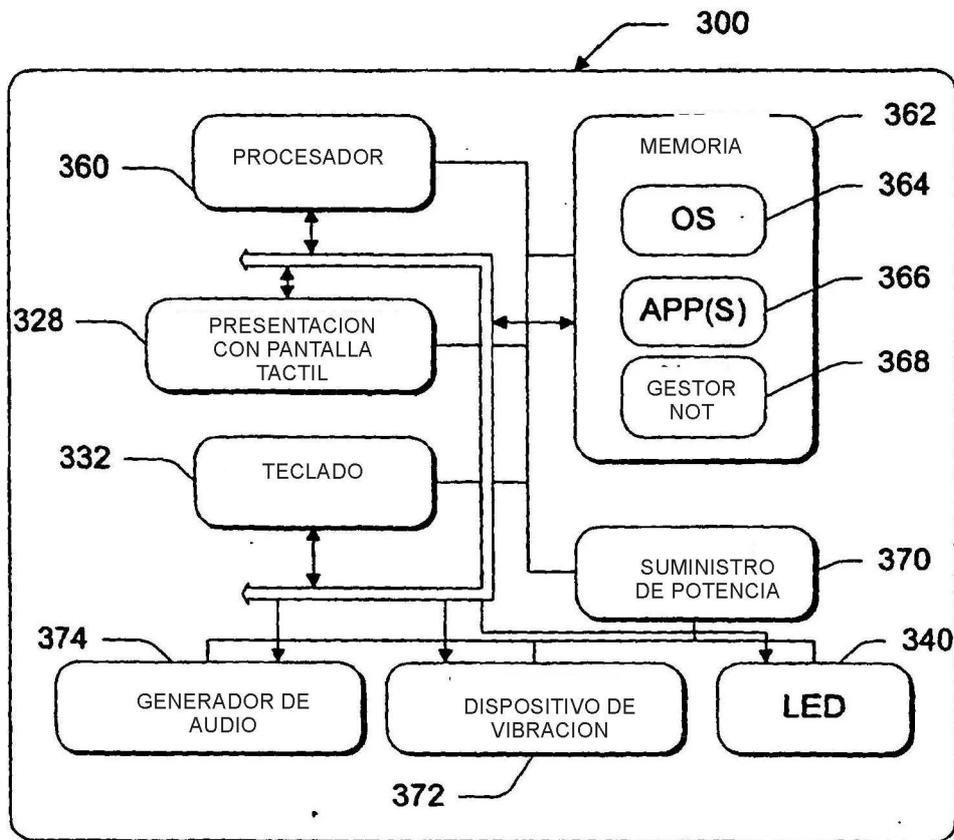


FIG. 3

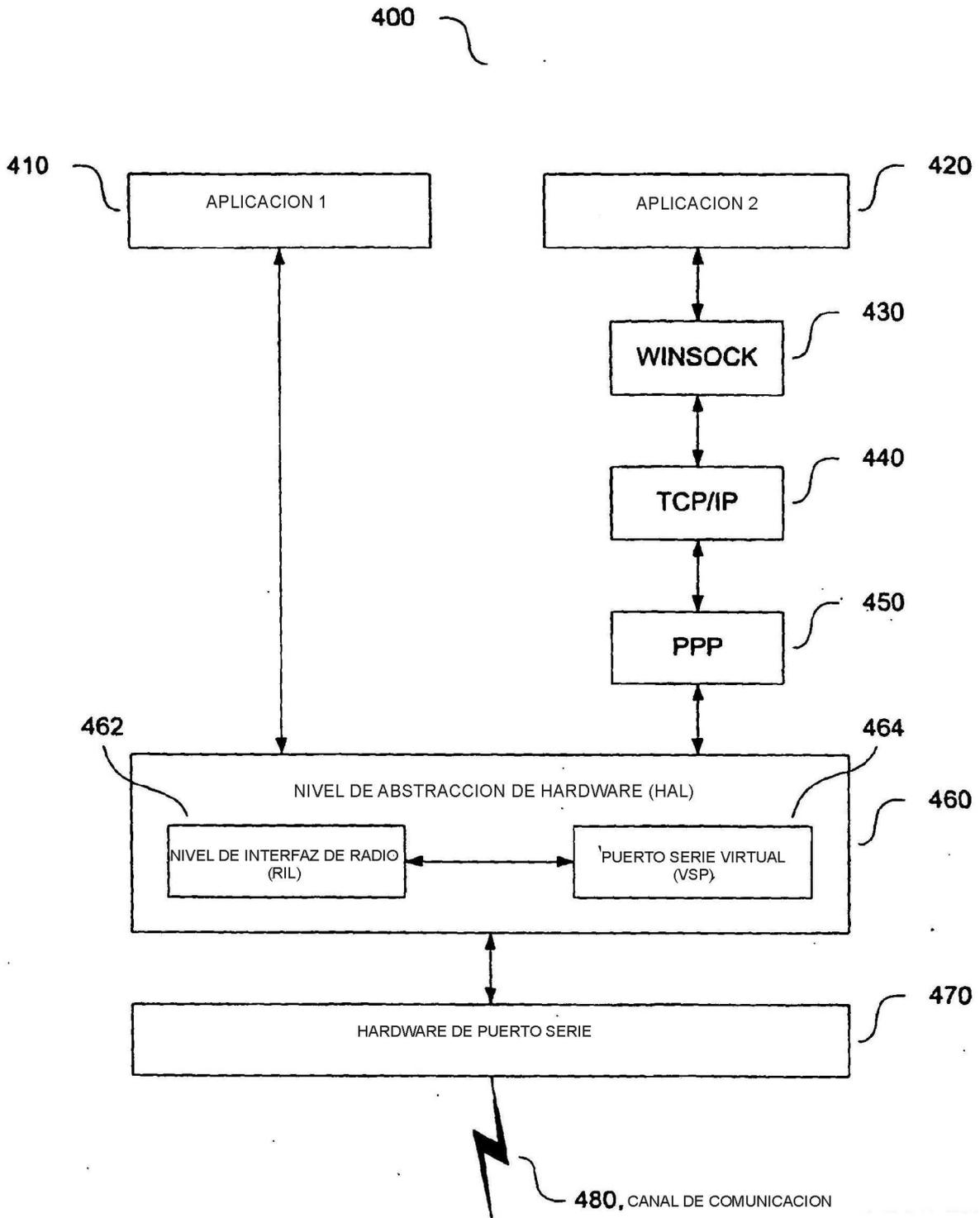


FIG. 4

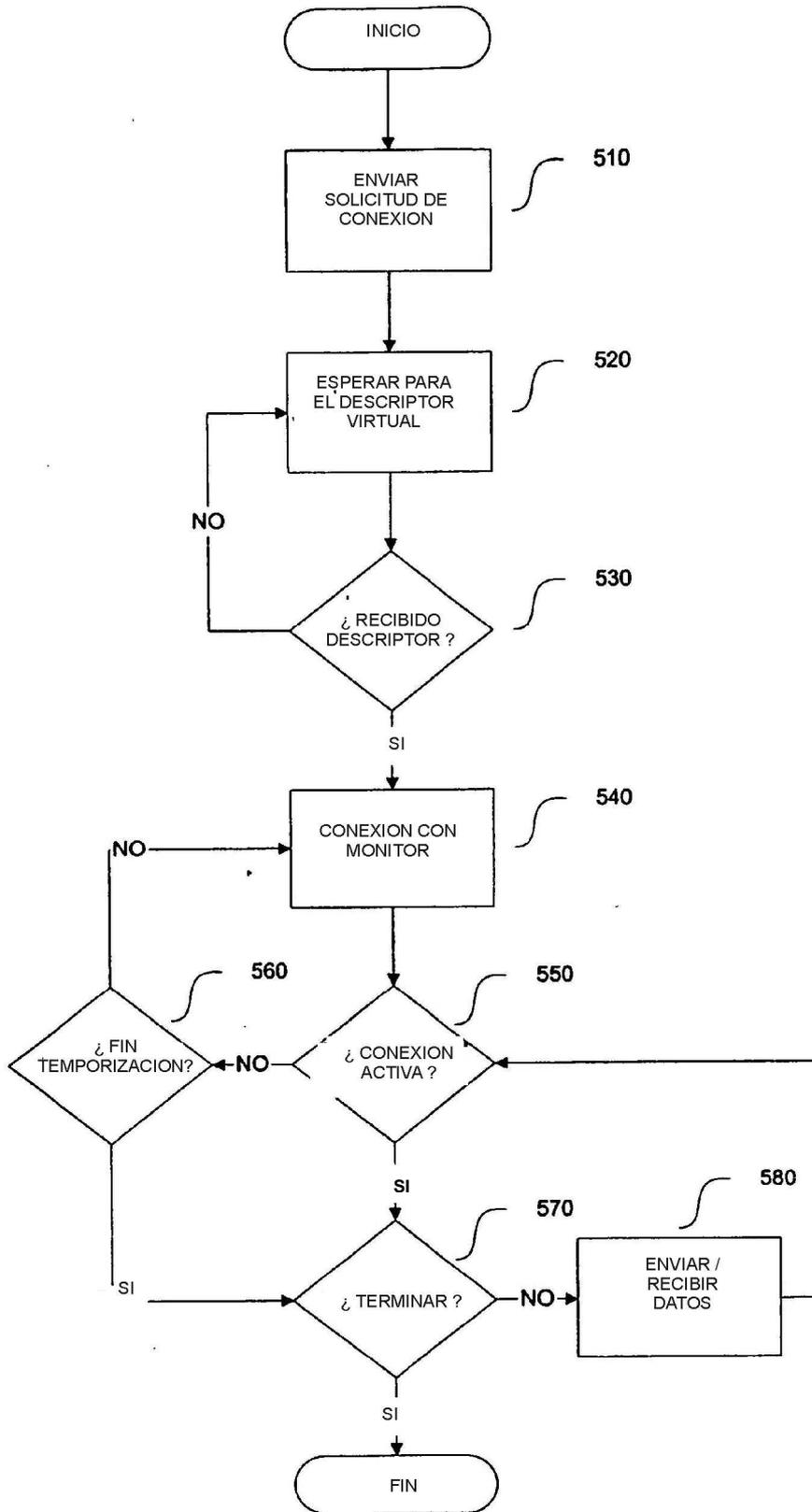


FIG. 5

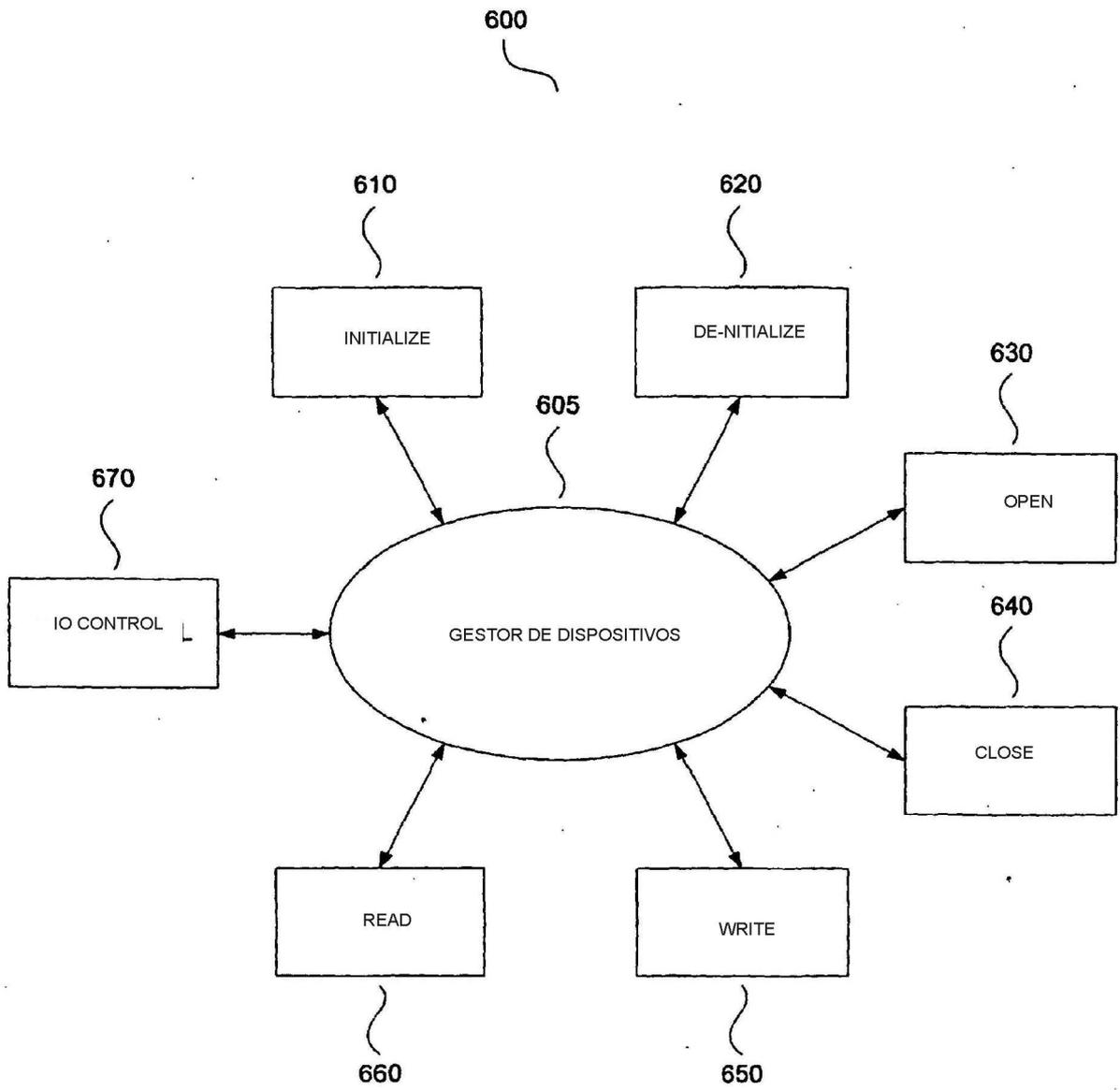


FIG. 6

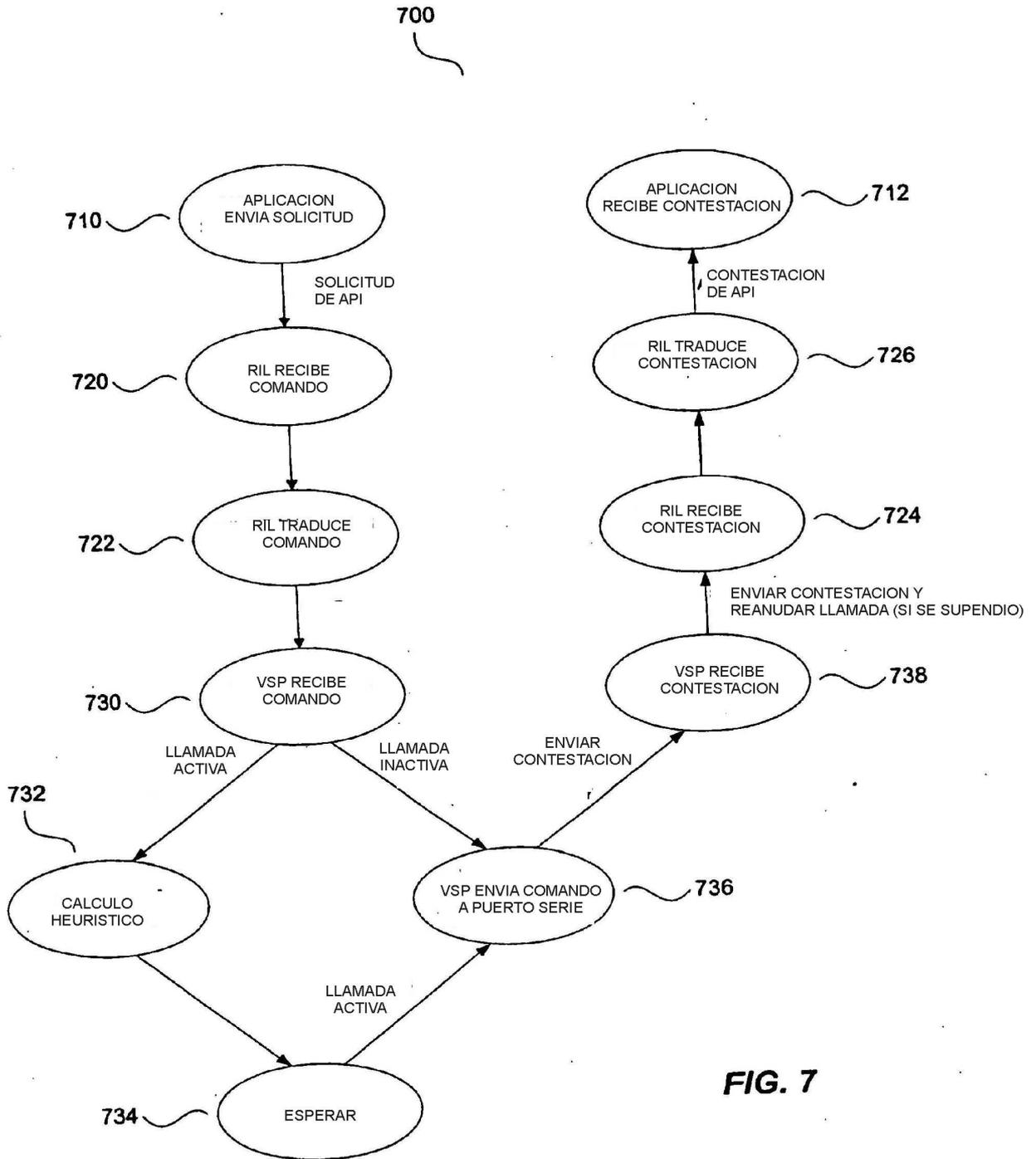


FIG. 7

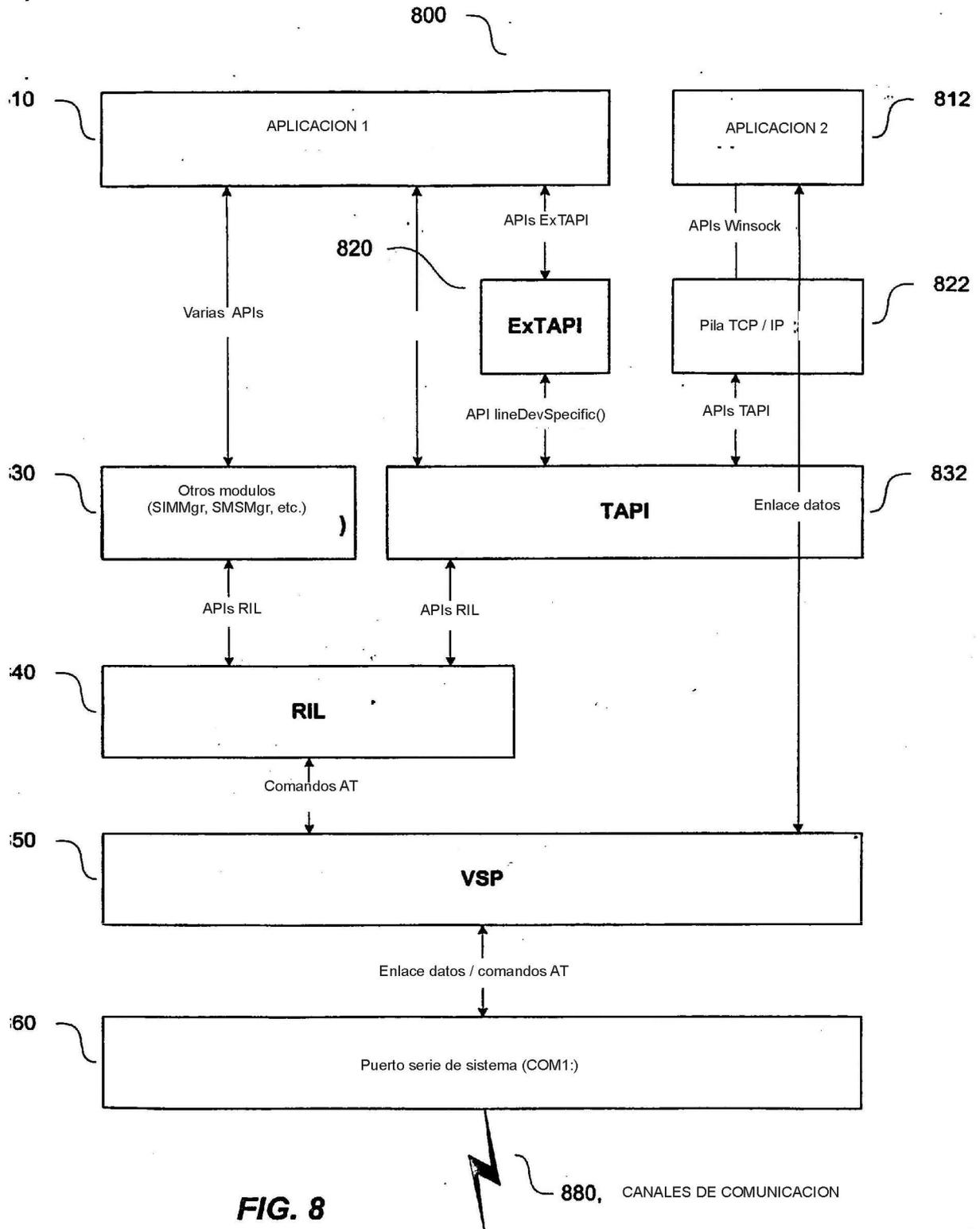


FIG. 8

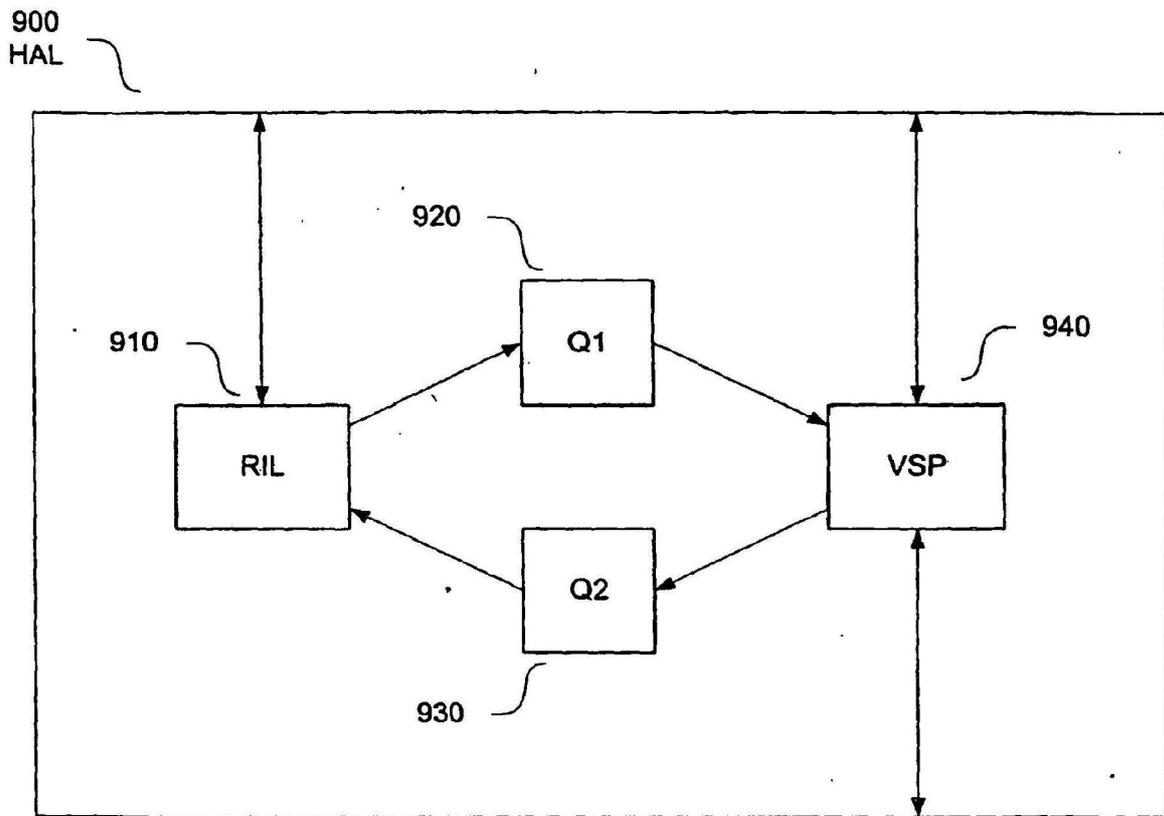


FIG. 9

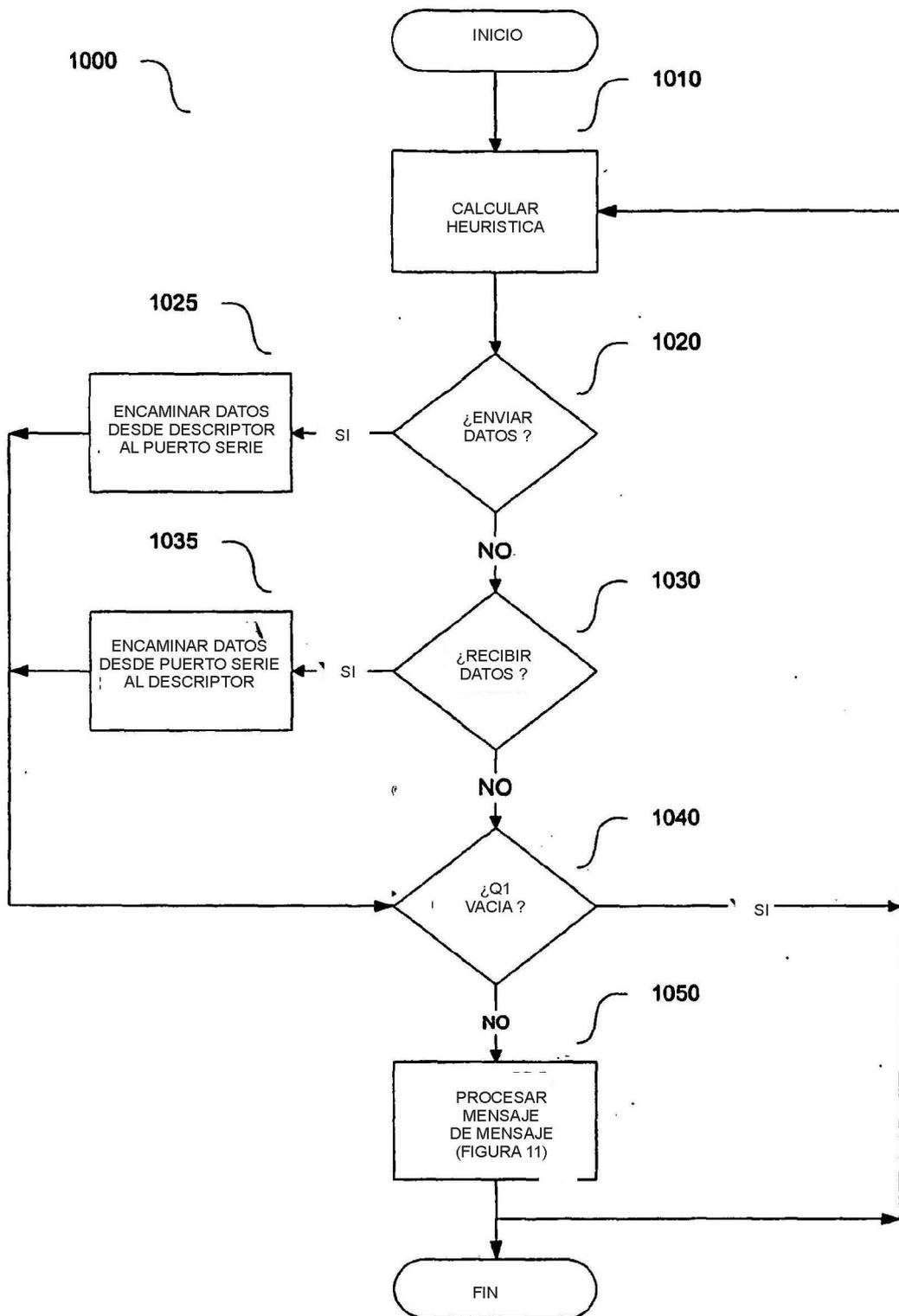


FIG. 10

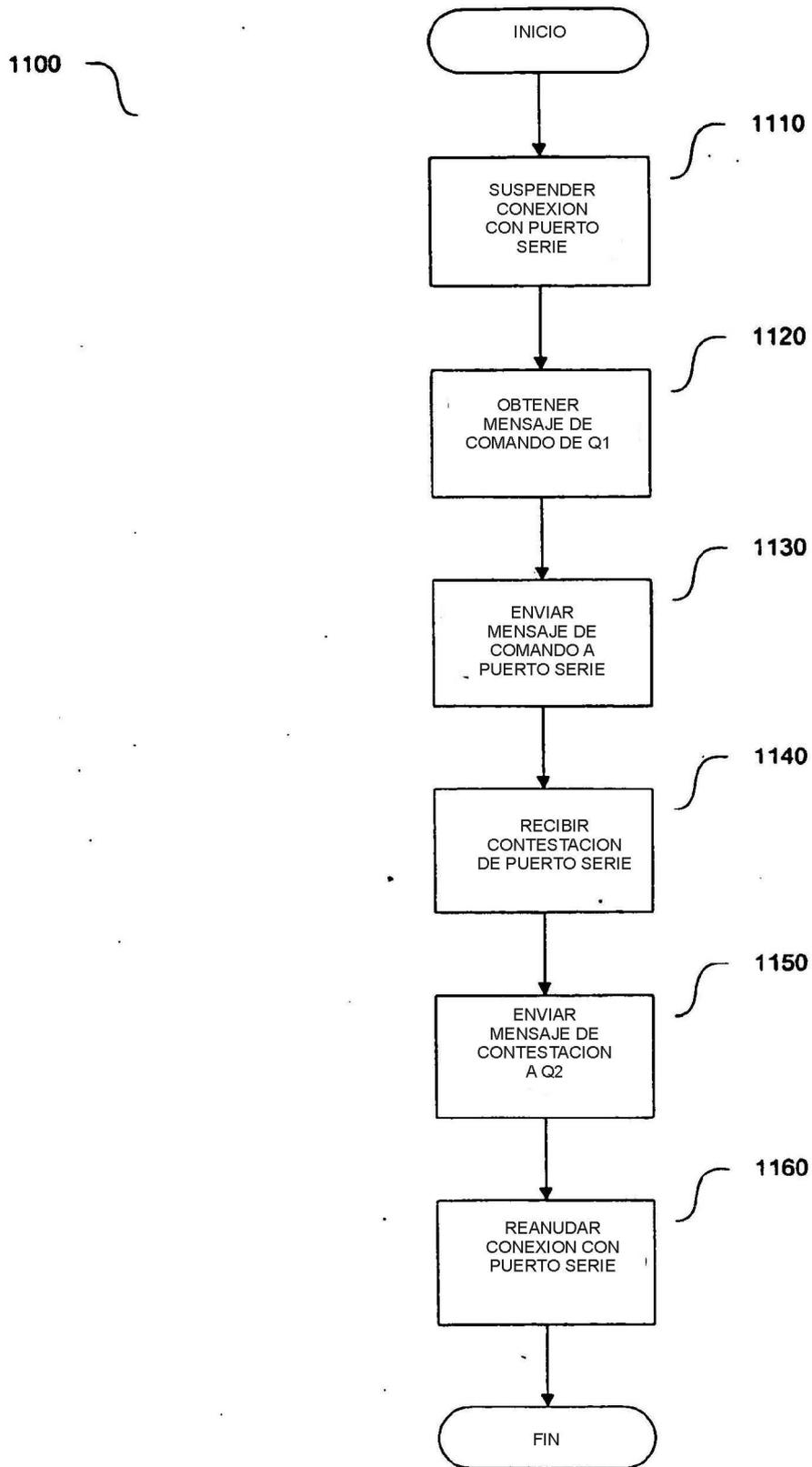


FIG. 11

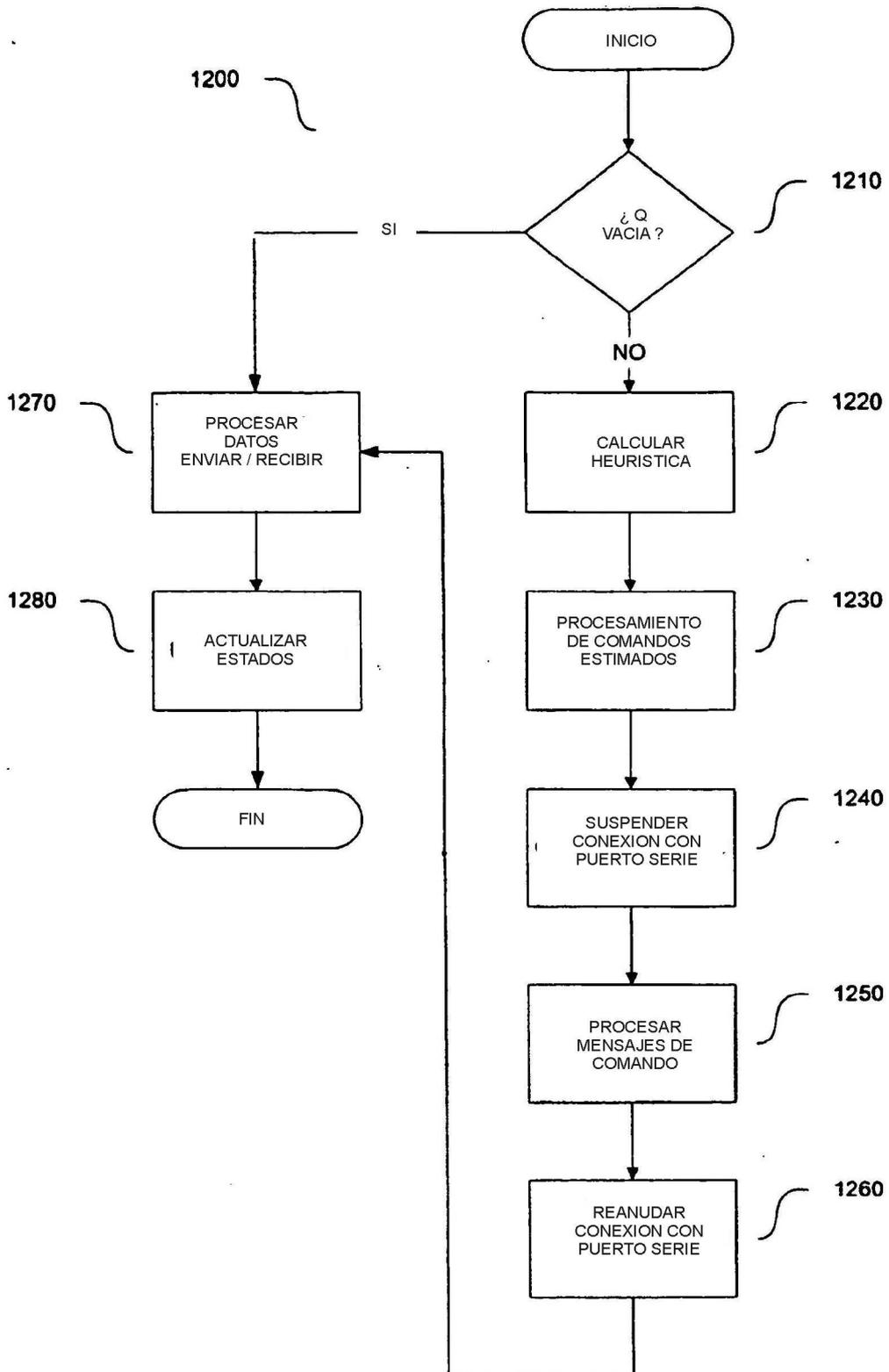


FIG. 12