(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 January 2001 (18.01.2001)

PCT

(10) International Publication Number
WO 01/04749 A1

(51) International Patent Classification[7]: G06F 9/445

(21) International Application Number: PCT/SE00/01450

(22) International Filing Date: 6 July 2000 (06.07.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
9902644-5 8 July 1999 (08.07.1999) SE

(71) Applicant *(for all designated States except US)*: **AXIS AB** [SE/SE]; Scheelevägen 34, S-223 63 Lund (SE).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **STEN, Per** [SE/SE]; Murarevägen 2 A, S-227 30 Lund (SE). **STARVIK, Mikael** [SE/SE]; Ystadsvägen 30 B, 2 tr, S-214 45 Malmö (SE).

(74) Agent: **AWAPATENT AB**; Box 5117, S-200 71 Mälmo (SE).

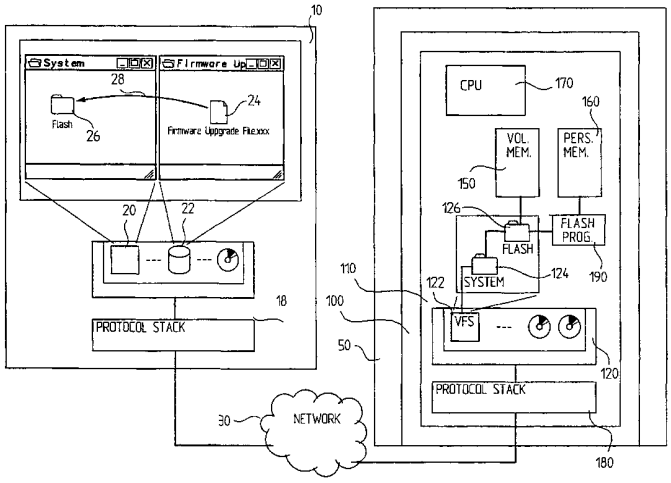
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AT (utility model), AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, CZ (utility model), DE, DE (utility model), DK, DK (utility model), DM, DZ, EE, EE (utility model), ES, FI, FI (utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KR (utility model), KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SK (utility model), SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*
— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD AND APPARATUS FOR UPGRADING FIRMWARE IN AN EMBEDDED SYSTEM



(57) Abstract: A method for upgrading firmware in an embedded system and a embedded system (110) provided with means for upgrading firmware received over a network (30) are disclosed. The method for upgrading firmware is performed from a client computer (10) to the embedded system (110) via a network connection by letting a virtual file system (122) of the embedded system (110) being shared with a client computer (10). The embedded system (110) is provided with means for upgrading firmware received over a network (30), and comprises a volatile memory (150), a persistent memory (160), a protocol stack (180) for communication over a network (30). Further it comprises a virtual file system (122), which is shareable with an operating system of a client computer (10) via the network (30), and a control means (128) for controlling a file intented for the virtual file system. Further the control means (128) comprises means for checking the characteristics of the file, and means for storing a file in the volatile memory.

WO 01/04749 A1

METHOD AND APPARATUS FOR UPGRADING FIRMWARE IN AN EMBEDDED SYSTEM

Technical field of the invention

The present invention relates to a method for upgrading firmware of an embedded system via a network connection. Further, the present invention relates to an embedded system provided with means for upgrading firmware received over a network, comprising; a volatile memory, a persistent memory, and a protocol stack for communication over a network.

Background of the Invention

The technology of data communication develops fast and the computer networks are expanding and consist of an increasing number of clients and servers.

Users and owners of networks require that servers and the embedded systems thereof are up to date in accordance with the rapidly changing data communication technology. Further, an increasing part of the development of embedded systems is today performed as development of firmware. Because of the rapid development of data communication technologies and requirements of new functionality from users the firmware of an embedded system has to be upgraded frequently.

Further, the use of computer networks is increasing, and today almost every company has a computer network of their own, or is at least connected to one. The increasing number of computer networks results in an increasing number of individuals responsible for the daily maintenance of networks. Many of those individuals are not specialised in computers or computer communications.

Thus, the rapid development of computer communication technologies and products, and the increasing number of individuals responsible for the daily maintenance

require that the firmware upgrade of the servers can be performed in a way that is easy and secure, in respect of low error probability.

The upgrading of firmware is today commonly managed by connecting a computer direct to the embedded system. Further it is known to perform upgrading of firmware over the network which the server is connected to.

A significant problem in upgrading the firmware of known embedded systems is that the individual performing the upgrade has to have detailed knowledge of the computer network and the embedded system and have to perform a number of rather complicated steps. Most individuals are not confident in performing said steps and are not confident in using all numerals and characteristics required to be input when performing the firmware upgrade. Further, the upgrading requires a specially designed program within the client to download the firmware upgrade to the embedded system. These aspects makes the operation of upgrading firmware of an embedded system difficult and time consuming for an individual, especially if the individual is not specially skilled in the art of computers and computer communications. In the known art an upgrade file is downloaded to the embedded system by means of a file transfer protocol. The download is performed with a special purpose program and the individual performing the download is required to specify, for example, where, in the form of a network address, the data is to be sent.

Another common problem, when upgrading firmware in embedded systems, is the risk of communication failure during the firmware download. Such failure could render the embedded system inoperative, thus requiring considerate efforts to repair.

US 5 812 857 discloses a method for upgrading a download code set, where an embedded system uses the same network drivers as when in normal operation. Further it describes a download where different portions of the code

set are downloaded during different steps of the downloading operation. Some portions of the code set are downloaded to a flash memory via a volatile memory while some portions are downloaded directly to the flash memory. There is also described that a temporary downloader, downloaded during an initial step and replaced by yet another downloader during a later step, is used during the operation of download. The described system and method is designed for minimising the time of executing code out of volatile memory to the persistent memory and is rather complicated.

Summary of the Invention

The object of the present invention is to provide a new and improved method for upgrading firmware in an embedded system and a new and improved embedded system, in respect of upgrading firmware.

Another object of the invention is to provide a firmware upgrading method that is easy to use and that saves time.

Yet another object of the invention is to provide a firmware upgrading method in which the probability of an erroneous upgrade is minimised.

These objects, as well as other objects that will become apparent from the description below, are accomplished by a method according to claim 1 and 8 and by an embedded system according to claim 19. Preferred embodiments of the invention are disclosed in the dependent claims.

Within the context of the invention the term embedded system denotes a special-purpose computer built into and integral to a server device. Wherein the server device can be, for example, a CD-server, a printer server, a file server, a communication server, etc.

According to the invention, a virtual file system is used in the embedded system. This leads to the advantage that the individual starting the upgrade can be working

in a well-known environment and can be able to use the same commands as when handling files within the client computer. Such handling of files within the client system is performed daily by users of computers, thus, the individual is probably familiar with performing such handling and thereby the system according to the invention can be time saving.

Further, by using a virtual file system within an embedded system a simple way of transferring a file from the client computer to the embedded system can be used, e.g. by sharing the virtual file system with the client computer. Sharing a file system shall be interpreted as the file system being arranged in such a way that it can be used within a client computer and be handled as if it was a part of the file system of that client computer.

Yet another advantage of the invention is that the protocol stack of the embedded system can be the same for both the normal operation of the server and the upgrading operation. Thus, it is possible to save space in the persistent memory of the embedded system. It is to be understood that the virtual file system is operating independently of the protocol stack currently in use.

By storing the data of the firmware upgrade file in the volatile memory before writing it to the persistent memory, critical data can be checked before it is written to persistent memory. Thus, it is possible to check if the data represents a correct firmware upgrade file and if the data is intended for the product related to the embedded system.

In a preferred embodiment of the invention, the virtual file system of the embedded system is mapped within the client computer. Thereby no specially designed programs have to be used at the client computer. Thus, both time and memory space of the client computer is saved.

According to a specific embodiment of the method of the invention, the entire firmware upgrade file is stored

in the volatile memory of the embedded system before it is written to the persistent memory. Thereby an error in the firmware upgrade file can be identified, e.g. by letting control means check the file, before the file replaces the firmware currently stored in the persistent memory. Such errors are generally a result of interruptions in network traffic or other network related errors.

Brief Description of the Drawings

Other features and advantages of the present invention will become apparent from the following detailed description of a presently preferred embodiment, with reference to the accompanying drawings.

Fig. 1 is a schematic diagram of an environment of the embedded system of the invention.

Fig. 2 is a schematic diagram of an embedded system and a client computer during an upgrade of firmware of the embedded system.

Fig. 3 is a detailed view of a portion of the embedded system shown in Fig. 2

Fig. 4 is a diagram showing events according to the method of the invention.

Detailed Description of a Presently Preferred Embodiment

Referring to Fig. 1, an environment of the invention is shown where a client computer 10 and servers 100 are connected to a network 30. The servers 100 each provide a device 50 with means for communication with the client computer 10 over the network 30. The client computer 10 described herein could be any client computer of a plurality of client computers connected to the network 30. The servers 100 each comprise an embedded system 110 performing the processing of the servers 100. A firmware upgrade is started at the client computer 10 and is directed to one embedded system 110.

A server 100, comprising the embedded system 110, enables devices 50 to be reached or shared via the network. The server 100 could be built in to a device 50, arrangement b, or could be external to a device 50, arrangement a.

A device 50 can be any device that could be interesting to connect to a network. For example, it could be an industrial robot, a control unit or a gauge for a process or a peripheral device, such as a printer, a modem, a scanner, a fax machine, a photocopier or a multifunction peripheral.

Now referring to Fig. 2, the embedded system 110 of the preferred embodiment comprises a file system 120, a volatile memory 150, a persistent memory 160, a central processing unit 170 (CPU), at least one protocol stack 180 and a programmer 190.

Within the file system 120 a virtual file system 122 (VFS) is implemented and in the virtual file system a directory for upgrading 126, hereinafter called "FLASH", is provided. In the preferred embodiment of the invention the FLASH directory 126 is a subdirectory to a directory 124 called "SYSTEM", which also comprises other resources.

The volatile memory 150 is a memory that loses the information stored therein when the power to it is switched off and it is used to store temporary data. The persistent memory 160 is a reprogrammable memory where stored data is preserved even when the power to it is switched off. In the preferred embodiment the persistent memory 160 is a flash memory but could be any reprogrammable memory preserving data after the power has been switched of. The CPU 170 is processing the functions, protocols, etc of the embedded system. The protocol stack 180 contains the protocols for communication over the network. The programmer 190, herein after called flash programmer 190, is a means for writing code into the persistent memory 160.

The client computer 10 comprises an operating system having a file system 18 allowing mapping of resources 20 shared over the network 30.

The server 100 related to the device 50 is mapped within the file system of the client computer.

The server 100 is generally mapped in the file system 18 for other purposes than firmware upgrading. If, for example, the device 50 served by the server is a file storage device, the storage area is generally accessible from the client computer by the use of ordinary file handling commands of the operating system of the client computer 10. Thus, there is established a pointer from the file system 18 of the client computer 10 to a root directory of the file system 120 of the embedded system 110 of the server 100. Thereby, the subdirectories of the file system 120 of the embedded system 110 also are available within the file system 18 of the client computer 10. If the server 110 is not already mapped in the file system 18 of the client computer 10 for normal operation of the device 50, then the server 100 has to be mapped at the client computer 10 before performing the upgrading operation according to the invention.

This arrangement results in the directories of the embedded system 110 being possible to handle within the file structure of the operating system of the client computer 10. Further, the directories of the embedded system 110 can be accessed as if they where directories of the client computer 10, using ordinary file handling commands.

Referring now to Fig. 3, showing the FLASH directory 126 within the virtual file system 122 of the embedded system 110. There is a control means, realised by a data channel 128, associated with the FLASH directory 126. Hereinafter this data channel 128 is referred to as flash data channel 128. The flash data channel 128 is reset every time the FLASH directory 126 is accessed. The flash channel is allocated a portion of

the volatile memory 150 for storing data. The flash data channel 128 handles all data sent to the FLASH directory 126 and is specially designed to handle firmware upgrade files and to perform various control operations on the data received.

To perform said control operations the flash data channel 128 comprises means for checking the characteristics of the file, namely first checking means 131, second checking means 132 and third checking means 133, and means 130 for storing data of a file in the volatile memory 150. Said checking and storing means are realised by program code related to the flash data channel 128 .The first checking means 131 checks if the file is a firmware upgrade file by checking for a "magic number", described below. The second checking means 132 checks if the file stored in the volatile memory 150 is intended for the product related to the embedded system 110. The third checking means 133 checks the checksum of the file to decide if there are any errors in the file.

Further the flash data channel 128 provides a flash programmer 190 with the approved firmware upgrade files to have the firmware upgrade file written to the persistent memory 160. A more detailed description of the functions of the flash data channel is found below, in the description of the method.

Now referring to Figs. 2 and 4, describing the preferred method for upgrading firmware. We presume that the server 100 has been mapped in the file system 18 of the client computer 10.

Let us assume that an user is interested in upgrading the firmware of a server with a firmware upgrade file 24, which he has received from the manufacturer of the server 100 and has stored on the hard disk 22 of the client computer 10. It should be understood that the firmware upgrade file of course could be stored on any storage unit accessible from the client computer 10.

The user then uses the ordinary file handling commands, for the present operating system, to copy or move the firmware upgrade file 24 to the FLASH directory. In the preferred embodiment the individual "drags and drops" 28 the firmware upgrade file 24 to the representation of the FLASH directory 26 within the client computer.

The client computer 10 thereafter sends an instruction to the embedded system telling it to open a flash file, step 402, within the virtual file system 122 of the embedded system.

Thereafter, the flash data channel is reset, step 404, and data of the firmware upgrade file 24 is received and stored, step 406, in the volatile memory 150. The flash data channel controls the data during reception, step 408, checking for an identifier in the form of "a magic number" corresponding to an identifier confirming that the data received belongs to a firmware upgrade file.

The check, in step 408, for a magic number identifying the data as a firmware upgrade file, checks a predefined position in the data file for the "magic number". Preferably the magic number is positioned at the beginning of the firmware upgrade file.

If the magic number corresponding to a firmware upgrade file does not occur at the predefined position in step 408, then the data of the file does not belong to a firmware upgrade file. As a result thereof an error message is sent, step 422, to the client and the file transmission is terminated.

If a magic number corresponding to a firmware upgrade file 24 is received, step 408, then the reception of data continues, steps 412, 414, until all data of the firmware upgrade file is stored in the volatile memory 150.

When all data of a firmware upgrade file has been received the flash data channel controls the checksum,

step 418, of the received file. If the checksum is not correct then the received file is corrupt. As a result thereof an error message is sent, step 422, to the client computer and the upgrading operation of the embedded system is terminated.

If the checksum is correct then the flash data channel is checking, step 420, for a product identifier within the file telling what product the firmware upgrade file is intended for. If the identifier does not correspond to the server product then an error message is sent, step 422, to the client computer and the upgrading operation of the embedded system is terminated.

If the product identifier is correct, then the flash programmer 190 is ordered to erase at least a portion of the memory and then write, step 424, the firmware upgrade file in the volatile memory 150 to the flash memory 160. When the firmware upgrade file has been written to the flash memory 160 the flash file is closed, which is indicated at the client computer in a similar way as when an ordinary file transfer has been completed.

CLAIMS

1. A method for upgrading firmware in an embedded system (110) from a client computer (10) via a network connection, comprising the steps of:

sharing a virtual file system (122) of the embedded system (110) with the client computer (10) in such a way that the virtual file system (122) can be utilised within the operating system of the client computer (10),

sending data of a firmware upgrade file over the network (30) from the client computer (10) to the embedded system (110) as a result of the upgrade file being copied or moved (28) to a directory (26), within the file system (18) of the client computer (10), which directory (26) represents a link to the virtual file system (122) of the embedded system (110),

receiving (406, 412), from the client computer (10), data of a firmware upgrade file at the virtual file system (122) of the embedded system (110) via the link,

storing (406, 412) the data of the firmware upgrade file in a volatile memory (150) of the embedded system (110) connected to the virtual file system (122), and

writing (424) the firmware upgrade file from the volatile memory (150) of the embedded system (110) to a persistent memory (160) of the embedded system (110).

2. Method according to claim 1, further comprising the step of controlling (408) if data received at the virtual file system (122) corresponds to a firmware upgrade file.

3. Method according to claim 2, further comprising, when the data received corresponds to a firmware upgrade file, the step of checking (420) if the firmware upgrade file stored in the volatile memory (150) is intended for the product type of the embedded system (110).

4. Method according to claim 2 or 3, further comprising, when the data received corresponds to a

firmware upgrade file, the step of checking (418) the checksum of the received file.

5. Method according to any of claims 2-4, comprising the step of sending an error message (422) from the embedded system (110) to the client computer (10) when the data received does not correspond to a valid firmware upgrade file.

6. Method according to any of claims 1-5, wherein the protocols (180) used during the reception of the firmware upgrade file are the protocols (180) previously used by the embedded system (110) during the normal operation.

7. Method according to any of claims 1-6, wherein the received data of a firmware upgrade file is controlled within the embedded system by a data channel (128) of the virtual file system (122).

8. A method for upgrading firmware of an embedded system (110) over a network connection, comprising the steps of:

sharing a virtual file system (122) of the embedded system (110) with an external computer (10) via the network connection in such a way that the virtual file system (122) can be utilised by the operating system of the external computer (10) as if it was part of the file system of the external computer (10),

receiving (406) data of a firmware upgrade file at the virtual file system (122) of the embedded system (110) via the network connection.

storing (406) the data of the firmware upgrade file in a volatile memory (150) connected to the virtual file system (122), and

writing (424) the firmware upgrade file from the volatile memory (150) to a persistent memory (160).

9. Method according to claim 8, further comprising the step of controlling (408) if data received at the virtual file system (122) corresponds to a firmware upgrade file.

10. Method according to claim 9, further comprising, when the data received corresponds to a firmware upgrade file, the step of checking (420) if the stored firmware upgrade file is intended for the product type of the device (50).

11. Method according to claim 9 or 10, further comprising, when the data received corresponds to a firmware upgrade file, the step of checking (418) the checksum of the received file.

12. Method according to claim 9, comprising the step of sending an error message (422) when the data received does not correspond to a firmware upgrade file.

13. Method according to claim 10, comprising the step of sending an error message (422) when the stored firmware upgrade file is not intended for the product type of the device (50).

14. Method according to any of claims 8-13, comprising the step of erasing at least a portion of the persistent memory (160) after reception of a correct firmware upgrade file and before writing (424) said file to the persistent memory (160).

15. Method according to any of claims 8-14, wherein the protocols (180) used during the reception of the firmware upgrade file are the protocols (180) previously used by the device (50) during the normal operation.

16. Method according to any of claims 8-15, wherein the received data of a firmware upgrade file is controlled by a data channel (128) of the virtual file system (122).

17. Method according to any of claims 8-16, wherein the step of writing (424) the firmware upgrade file from the volatile memory (150) to the persistent memory (160) is preceded by all data of the firmware upgrade file being located in the volatile memory (150).

18. Method according to any of claims 8-17, wherein the virtual file system (122) is mapped within a client computer (10) in such way that the virtual file system

(122) can be utilised within the operating system of the client computer (10).

19. An embedded system (110) provided with means for upgrading firmware received over a network (30), comprising:

a volatile memory (150),

a persistent memory (160),

a protocol stack (180) for communication over a network (30), and

a virtual file system (122) which is shareable with an operating system of a client computer (10) via the network (30), for upgrading said firmware.

20. A system (110) according to claim 19, further comprising a control means (128) for controlling a file intended for the virtual file system (122),

wherein the control means (128) comprises

means (131, 132, 133) for checking the characteristics of the file, and

means (130) for storing a file in the volatile memory (150).

21. A system (110) according to claim 20, wherein the virtual file system (122) comprises an upgrading directory (126) and wherein the control means (128) is arranged to control a file intended for the upgrading directory (126).

22. A system (110) according to claim 20 or 21, wherein a first means (131) for checking the characteristics of the file is arranged to check that the file is a firmware upgrade file.

23. A system (110) according to any one of claims 20-22, wherein a second means (132) for checking the characteristics of the file is arranged to check that a received firmware upgrade file is a firmware upgrade file intended for the product the embedded system (110) is representing.

24. A system (110) according to any one of claims 20-23, wherein a third means (133) for checking the

characteristics of the file is arranged to check the checksum of a received firmware upgrade file.

25. A system (110) according to any one of claims 19-24, wherein a common protocol stack (180) for communication over the network (30) is used during normal operation and during download.

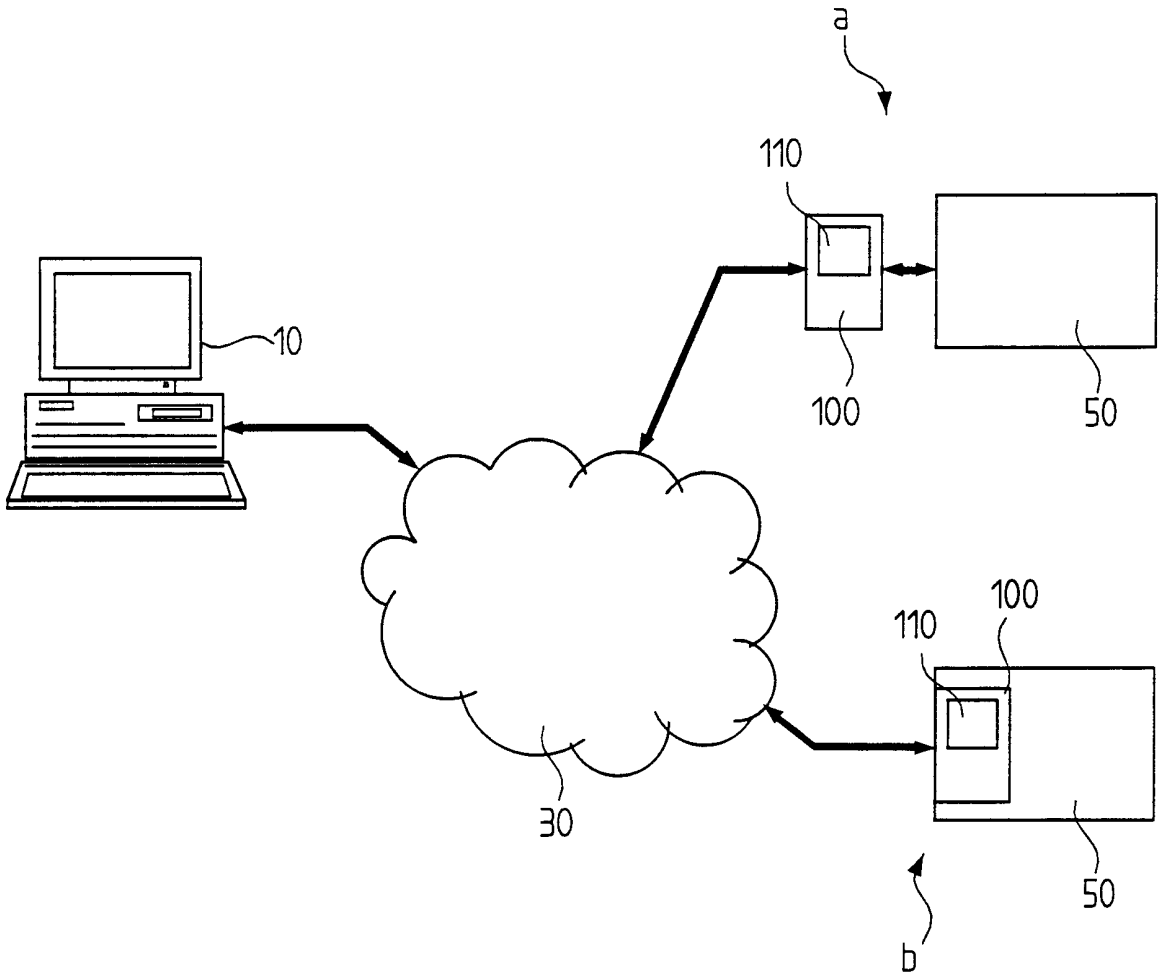


FIG. 1

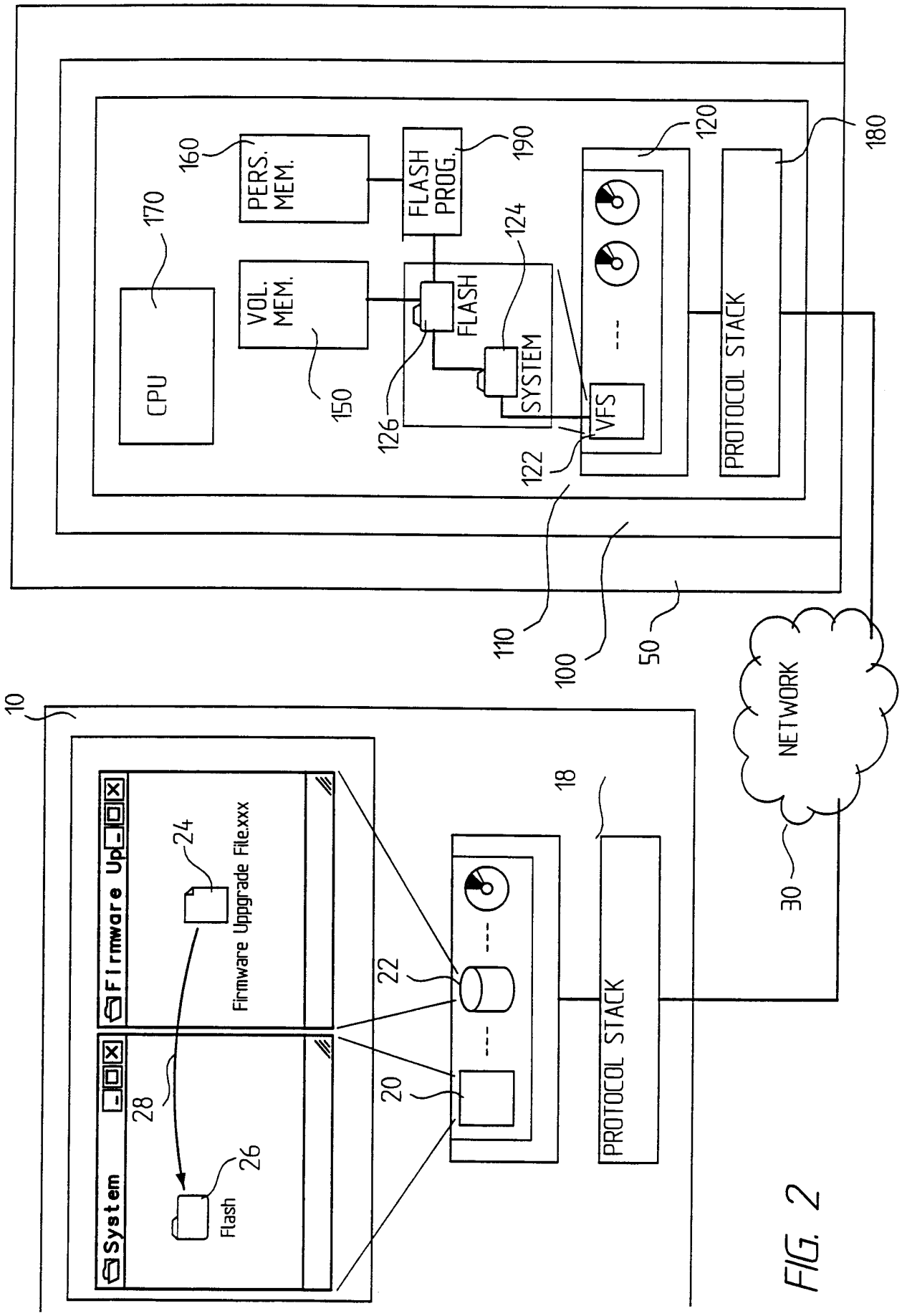

CPU
170
PERS. MEM.
160
VOL. MEM.
150
FLASH PROG.
190
FLASH
126
SYSTEM
124
VFS
122
120
PROTOCOL STACK
180
110
100
50
NETWORK
30
10
Firmware Up
Firmware Upgrade File.xxx
24
System
28
Flash
26
22
20
18
PROTOCOL STACK

FIG. 2

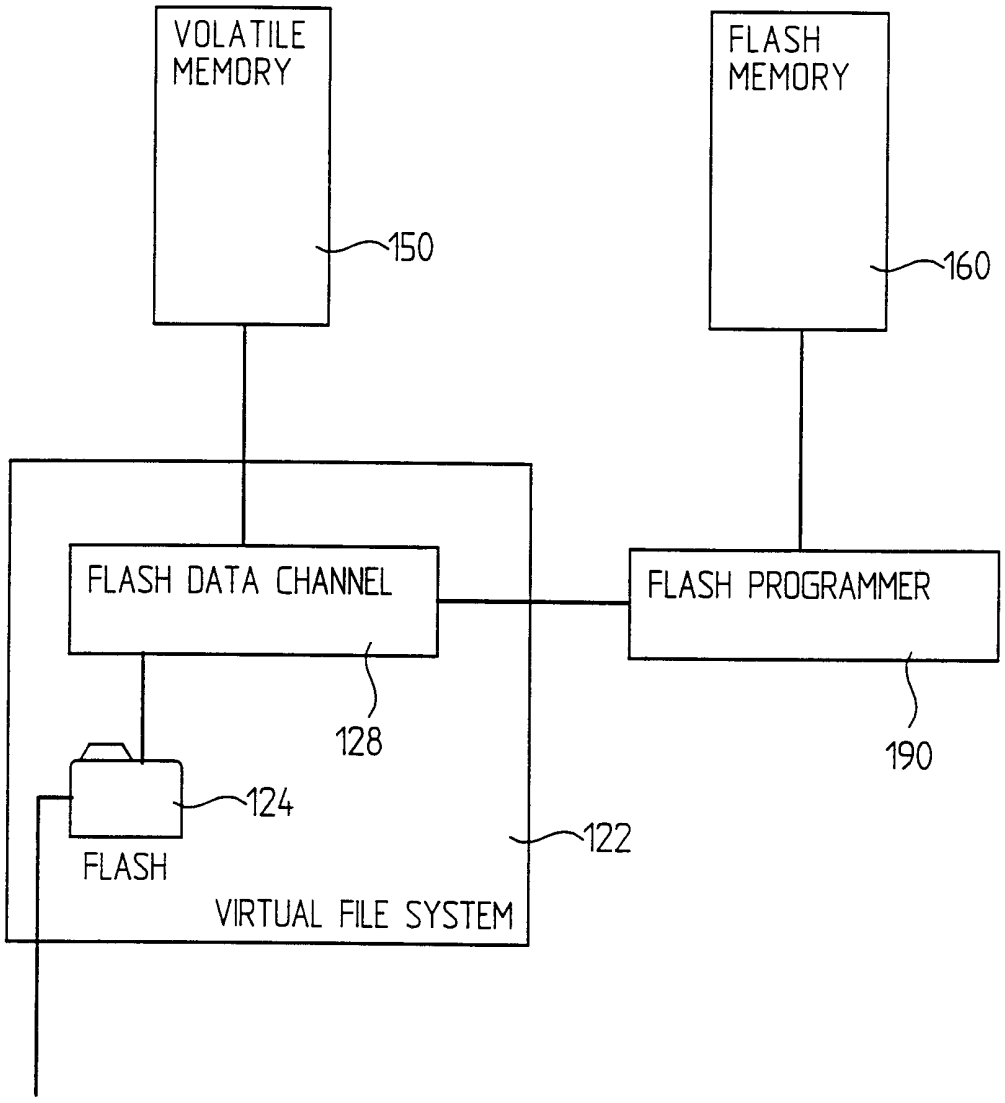

VOLATILE MEMORY
150
FLASH MEMORY
160
FLASH DATA CHANNEL
128
FLASH PROGRAMMER
190
124
FLASH
122
VIRTUAL FILE SYSTEM

FIG. 3

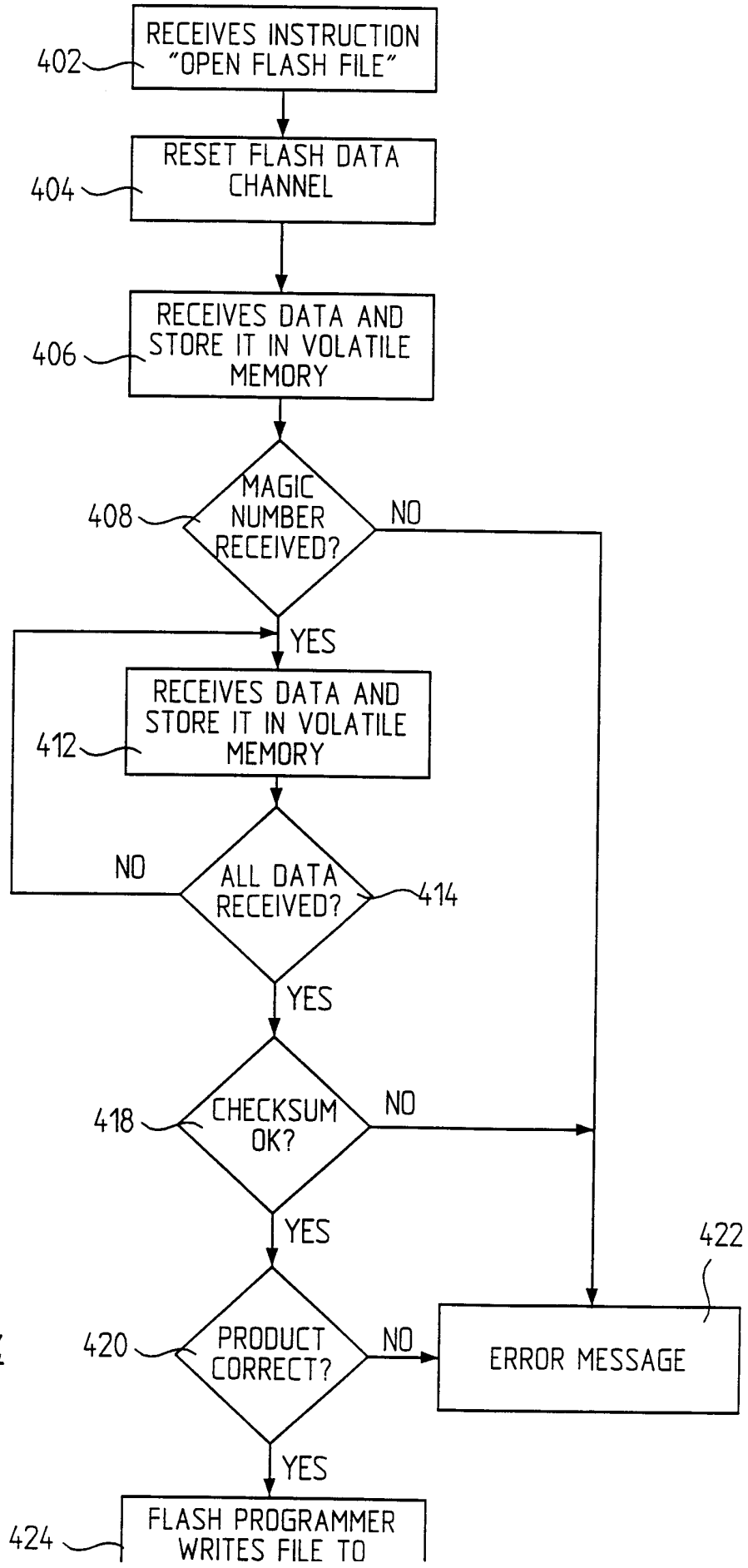

RECEIVES INSTRUCTION "OPEN FLASH FILE"
402
RESET FLASH DATA CHANNEL
404
RECEIVES DATA AND STORE IT IN VOLATILE MEMORY
406
MAGIC NUMBER RECEIVED?
408
NO
YES
RECEIVES DATA AND STORE IT IN VOLATILE MEMORY
412
ALL DATA RECEIVED?
414
NO
YES
CHECKSUM OK?
418
NO
YES
PRODUCT CORRECT?
420
NO
YES
ERROR MESSAGE
422
FLASH PROGRAMMER WRITES FILE TO
424

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE 00/01450

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: G06F 9/445
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC7: G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5812857 A (E.L. NELSON ET AL), 22 Sept 1998 (22.09.98), column 1, line 48 - line 67; column 4, line 13 - line 26, figure 1, abstract ---	1-25
A	US 4982430 A (W.A. FREZZA ET AL), 1 January 1991 (01.01.91), abstract ---	9-13
A	US 5452454 A (T.K. BASU), 19 Sept 1995 (19.09.95), the whole document ---	1-25
A	EP 0601704 A1 (CANON INFORAMTION SYSTEMS, INC.), 15 June 1994 (15.06.94), the whole document --- --------	1-25

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
9 October 2000	10 -11- 2000
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer Oskar Pihlgren / MRo Telephone No. +46 8 782 25 00

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/SE 00/01450

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0601704 A1 (CANON INFORAMTION SYSTEMS, INC.), 15 June 1994 (15.06.94), the whole document -- --------	1-25

INTERNATIONAL SEARCH REPORT

Information on patent family members

01/08/00

International application No.
PCT/SE 00/01450

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5812857 A	22/09/98	US 5937198 A	10/08/99
US 4982430 A	01/01/91	CA 1295412 A	04/02/92
		DE 3679711 D	00/00/00
		EP 0200704 A,B	05/11/86
		SE 0200704 T3	
		JP 2074607 C	25/07/96
		JP 7087455 B	20/09/95
		JP 61248636 A	05/11/86
US 5452454 A	19/09/95	US 5842011 A	24/11/98
EP 0601704 A1	15/06/94	DE 69323840 D,T	19/08/99
		JP 7073042 A	17/03/95
		US 5623604 A	22/04/97