

B3

System and Method for establishing wireless connection**Technical field**

The present invention relates in general to wireless data communication between different computer devices, and in particular to a method and apparatus for automatically establishing a wireless connection between a peripheral device and a computer device.

Background art

These days it is more and more common to have one user working with several computer systems in his office. With the current technology, each computer system uses its own peripheral device to control cursor displacement on the computer display screen, such as a cabled or a wireless mouse device. The major drawback is that the user must use a different peripheral device each time he works with a different computer system which is boring, not convivial and could lead to fastidious manipulation.

Thus there is a need for a unique peripheral device to be used whatever the number of computer systems the user works with.

Another problem arising from the use of different peripheral devices is the software support of this latter. In fact, currently each manufacturer of peripheral devices such as International Business Machines Corp. provides the peripheral device along with a diskette including a device driver program which is to be installed on the computer system by the user. The present invention offers a method to automatically install the device driver program of the peripheral device without the need of such a diskette.



G00002

CONFIRMATION COPY



159159

Finally, the present invention is directed towards a user friendly system and method which offers to the user an easy way to establish a wireless link between a peripheral device and a selected computer system among a plurality of computer systems.

Summary of the invention

In accordance with the invention, there is provided a peripheral device for establishing wireless connection with an intelligent device. The peripheral device comprises means for controlling a cursor displacement on a display screen of the intelligent device, and further comprises:

- means coupled to the controlling means for generating a plurality of identification signals, the identification signals comprising at least a device driver signal identifying the device driver associated to said peripheral device, and
- means coupled to the controlling means for analyzing at least one acknowledgment signal received from said intelligent device in response to said plurality of identification signals.

Preferably, first actuation means, such as a push button located on the cover of the peripheral device are actuated by a user during the generation of the plurality of identification signals. Similarly, second actuation means such as a predetermined key of a keyboard connected to the intelligent device are also actuated by the user during the generation of the plurality of the identification signals.

In the preferred implementation, the identification signals and the acknowledgement signals are HDLC frames having a specific control field for defining the kind of frame, i.e. if the frame is transmit from the peripheral device or from the intelligent device, if the frame is a synchronization one or a

desynchronization one or a frame which contains the device driver to be download to the intelligent device.

Also in accordance with the present invention, a system for establishing wireless connection between a peripheral device
5 and an intelligent device in a multi-computers environment having a plurality of intelligent devices, is provided. The system is in part included within the peripheral device which is preferably a mouse device and also is included within the intelligent device which is preferably a computer system.
10 Specific routines of the system allow a user to work with any of the intelligent devices using only the one mouse device.

Brief description of the drawings

Fig.1 depicts a multi-computers environment wherein the cableless peripheral device of the invention may link to
15 anyone of the computers.

Fig.2 is a top view of the mechanical part of a cableless mouse for use in accordance with the present invention.

Fig.3 is a schematic of one embodiment of the electronic implementation of the cableless mouse of the present
20 invention.

Fig.4 is a detailed schematic of the support circuit of Figure 3.

Fig.5 shows the different formats of the HDLC frames exchanged between the mouse in accordance with the present invention and
25 the computer system.

Fig.6 is a flowchart showing the steps of the mouse device stand-by routine.

Fig.7 is a flowchart showing the steps of the interrupt routine associated to the synchronization/desynchronization push button.

Fig.8 is a flowchart showing the steps of the mouse serial communication controller interrupt routine.

Figs.9-a,9-b,9-c show a flowchart of the steps of the infrared PC serial communication controller interrupt routine.

10 Detailed description of the invention

Fig.1 shows a general block diagram of a preferred infrared mouse environment of the invention. Generally stated, the present invention comprises a convenient technique for allowing a peripheral device to be connected through a wireless communication to a selected device among a plurality of intelligent devices in a dense computing environment. The selected intelligent device could be a stationary device or a mobile computing device. In the detailed embodiments presented below, the peripheral device is assumed to comprise a data mouse. However, those skilled in the art should recognize that the concepts presented herein are equally applicable to other types of peripheral data storage or computing devices. On figure 1 a mouse device 10 is wirelessly coupled, e.g., using infrared waves 12 and 14, to a selected computer system 16 among a plurality of neighboured computer systems (16-2 to 16-n). System 16 is equipped with graphical user interface software, such as that supported by the IBM OS/2 operating system of International Business Machines Corp.. Use of infrared waves 12 and 14 allows greater freedom of placement of device 10 compared with a hardwired connection of the devices. Other wireless communication techniques could also be employed. For example, a wireless optical communication link or electromagnetic wave may be used if desired. A preferred

mouse configuration is shown with two "clicker" buttons 18, a "push" button 20 and a mouse ball assembly (not shown). Computer system 16 has a display screen, a keyboard and additional hardware 22 to communicate with mouse device 10. As explained briefly above, a significant advantage of the present invention is the ability to establish a wireless connection between a mouse device and a selected computing system in a dense computing environment having a plurality of computing systems (16-2 to 16-n) in close proximity to one another. Each computer system (16, 16-2 to 16-n) comprises specific features 22 made of hardware circuitry and software programs which allow in accordance with the present invention the automatic establishment of wireless connection between mouse device 10 and one of the multiple computer systems.

15

Figure 2 shows a mechanical view of one preferred embodiment of a mouse device 10 in accordance with the present invention. This device 10 has a built-in means for controlling the pointer on a graphical user interface equipped computing device. Although the present invention is described herein in connection with a mouse, the concepts presented are not limited to that particular portable structure. Device 10 contains standard mouse components such as a mouse ball 24 and mouse buttons 18. An infrared (IR) communications means 26 is added to wireless mouse 10 as well as a push button 20. These components are discussed further below in connection with FIG. 3.

Figure 3 shows the internal circuitry 30 of the mouse device 10. The electronic circuit 30 includes a set of light emitting diodes (LED) 26, an infrared LED driver 32, an infrared LED receiver 34 and wireless communication support circuitry 36. Support circuitry 36 incorporates the lowest hardware level of protocol for the communications, and is detailed with reference to FIG. 4.

Figure 4 is a detailed view of the support circuit 36 of Figure 3. Infrared LED driver 32 and infrared LED receiver 34 are connected to a multiplexer 40. One data input of multiplexer 40 is coupled to a mouse hardware control circuitry 41, the other data input of multiplexer 40 is coupled to a serial communication controller (SCC) 42. The mouse hardware control circuitry 41 is not described in detail as being a standard infrared implementation and not being part of the present invention. SCC 42 is also connected to a local bus 43 on which is also connected a mouse identification Programmable Read Only Memory (PROM) 44 and a microcontroller 45. A control I/O of microcontroller 45 is connected to the control input of multiplexer 40. Microcontroller 45 has two interrupt lines (INTP1,INTP2), the highest priority interrupt line 'INTP1' is connected to SCC 42, the lowest priority interrupt line 'INTP2' is connected to push button 20. Microcontroller 45 is a standard commercial component including well-known Random Access Memory (RAM) device and Read Only Storage (ROS) device. The ROS device stores the wireless communication program. The mouse identification PROM 44 stores a vendor identification pattern which allows to recognize the mouse manufacturer, and a device driver of the mouse.

A user willing to work with a desired computer system (PC) must synchronize the mouse to the PC. To do so, he has to press and hold one predefined key of the PC keyboard while the push button 20 located on the mouse device 10 is also pressed as long as the mouse pointer on the display screen stays inactive. When the mouse pointer becomes active such as having a moving appearance, the user stops pressing both the PC keyboard key and the mouse push button. Thereafter the user works in a usual wireless environment to begin any application on the selected computer system.

The predefined synchronization key on the keyboard which in the preferred implementation is chosen to be the 'S' key

letter is analyzed by the IR software running in the PC as it will be further described with reference to figure 9.

The user willing to work with another PC with the same mouse device has to first desynchronize from the active PC and then
5 to begin the synchronization operation with the new selected PC in the way as previously described. To desynchronize, the user must press and hold one predefined key of the PC keyboard while the push button 20 located on the mouse device 10 is pressed as long as the mouse pointer on the display screen is
10 active. This predefined desynchronization key which in the preferred implementation is chosen to be the 'D' key letter is analyzed by the IR software running in the PC as it will be further described with reference to figure 9. One skilled in the art will understand that any other predefined keys of a
15 keyboard could be selected for both the synchronization and the desynchronization operations.

Preferably these synchronization and desynchronization operations between the mouse device and the computer system use the well-known HDLC frames, but other protocols could be
20 used without departing from the spirit of the invention

Figure 5 shows the different formats of the HDLC frames exchanged between the mouse in accordance with the present invention and the computer device. On top of figure 5, lines 50, 51 and 52 illustrate the HDLC transmit frames (XMT frames)
25 sent by the mouse device 10 to the PCs (16, 16-i, 16-n). Similarly, the three bottom lines 53, 54 and 55 illustrate the HDLC receive frames (RCV frames) sent by the PCs (16, 16-i, 16-n) to the mouse device 10.

Generally speaking, the HDLC frames are made of a starting
30 flag (Fs), a control byte (CNTL), data bytes (V, Range), two Cyclic Redundancy Check (CRC) bytes and an ending flag (Fe). It is one feature of the invention that the control byte be

characteristic of each frame. It is composed of two parts: the four highest bits are referred as Acknowledge bits (ACK) and the four lowest bits are referred as Command bits (CMD). The command bits (CMD) are set by the mouse device while the 5 Acknowledge bits (ACK) are set by the PC.

The control byte (CNTL) specifies the type of the transmit and receive frame. In the preferred implementation, three types of frames are defined for each group of three frames, but it is to be understood that a different number of frames could be 10 defined.

For the transmit group, a first transmit frame is referred as the synchronization frame (SYNC). The control byte is set by the mouse device to an hexa pattern 01 (X'01'). For a second transmit frame referred as the desynchronization frame 15 (DE-SYNC), the control byte is set by the mouse device to an hexa pattern 04 (X'04'). For a third transmit frame referred as the device driver frame (DD) the control byte is set by the mouse device to an hexa pattern 02 (X'02').

20 For the receive group, a first receive frame referred as the PC synchronization acknowledgment frame (SYNC-PC-ACK), the control byte is set by the PC to an hexa pattern 11 (X'11'). For a second receive frame referred as the PC desynchronization acknowledgment frame (DE-SYNC-PC-ACK), the 25 control byte is set by the PC to an hexa pattern 44 (X'44'). For a third receive frame referred as the PC device driver acknowledgment frame (DD-PC-ACK) the control byte is set by the PC to an hexa pattern 22 (X'22').

In the data bytes area, a first byte 'V' defines the vendor 30 identification. This number is defined and given to a vendor by the infrared standard organization. Each vendor has a unique number.

The other data bytes are dependent on the type of the HDLC frames. For the referenced 50,51,53 and 54 frames, these bytes represent the address of the mouse device which is unique for each mouse device produced by a vendor. In the preferred 5 embodiment, the address is defined by four bytes which is a correct number to address more than four giga different mice, but the skilled man could easily defined another number of bytes if necessary.

For the referenced 52 and 55 frames, these bytes represent the 10 device driver of the mouse device which is downloaded from the mouse device to the PC. The number of bytes depends on the manufacturer device driver development.

As now illustrated by figure 6, the mouse device stand-by 15 routine is described. This routine starts running from the mouse power ON by the activation of a power-on switch (not shown of the figures). Microcontroller 45 performs a mouse device self-test at step 60. Next, block 62 tests if any error. In case of failure (branch YES) a visuable error LED 20 (not shown on the figures) is switched ON at step 64 and microcontroller 45 stops at step 66. In case there is no failure at the test device (branch NO), microcontroller 45 executes the common communication protocol layers on step 68. Any event such as a user action on push button 20 or status 25 reception from serial communication controller 42 generate an interruption of the stand-by routine to enter the respective event process.

Coming from reset, the mouse device is in IDLE state. When the user presses the push button 20, microcontroller 45 30 executes the steps of the interrupt routine associated to the synchronization/desynchronization push button as shown on figure 7. On block 700, the multiplexer is set to a port B which means that SCC 42 is connected to the infrared interface. On block 701, a software variable called

'SYNC-MODE' is tested active. Because the previous state of the mouse device is IDLE, the variable 'SYNC-MODE' is not active and the routine goes to block 702 where a synchronization frame 50 is sent. On next block 704, 5 processing holds for a predefined time interval. During this time interval, a serial communication controller interrupt routine is executed as it will be further described in details with reference to figure 8.

At the end of the time interval, the 'SYNC-PC-ACK' software 10 variable is tested to be active on block 706. If the variable is not active which means that the interrogated PC is not yet responding to the mouse request, then the routine loops to block 702 where a frame (50) is sent again.

If the variable is active, the process goes to block 708 where 15 the device driver frame (52) is sent to the responding PC.

On next block 710, processing holds for a predefined time interval. At the end of the time interval, the 'DD-PC-ACK' software variable is tested to be active on block 712. If the 20 variable is not active then the routine loops to block 708, where frame 52 is sent again.

If the variable is active (branch YES), the process goes to block 714 where the software variable SYNC-MODE is set active. On block 715, the multiplexer is set back to port A which 25 means that the mouse hardware control is connected to the infrared interface and the process exit the routine.

Coming back to block 701, when the SYNC-MODE variable is active, the process goes to block 716, where the desynchronization frame 51 is sent. On next block 718, 30 processing holds for a predefined time interval. At the end of the time interval, the 'DE-SYNC-PC-ACK' software variable is tested to be active on block 720. If the variable is not

active, then the routine loops to block 716 until the active PC sends an active 'DE-SYNC-PC-ACK' frame. In this case (branch YES of block 720), the process goes to block 722 where the software variable SYNC-MODE is set inactive. On block 715, 5 the multiplexer is set back to a port A which means that the mouse hardware control 41 is connected to the infrared interface and the synchronization process exit the routine.

Fig.8 is a flowchart showing the steps of interrupt routine of the serial communication controller 42 located in the mouse 10 device. This process is executed each time the serial communication controller transmits or receives a frame. On block 800, microcontroller 45 reads the content of an interrupt control register. Block 802 determines the source of the interrupt signal (INTP1). On branch YES, the source 15 interruption is an end of transmission and the process exits the routine. On branch NO, the source of the interruption corresponds to the reception of a frame sent by a PC, and the process goes on with step 804 which checks if the reception is a frame of the type PC synchronization acknowledgment (53). If 20 YES, the process goes to step 816 where the software variable SYNC-PC-ACK is set active and the process exits the routine. If NO, the process goes to step 806 which checks if the reception is a frame of the type PC device driver acknowledgment (55). If YES, the process goes to step 814 25 where the software variable DD-PC-ACK is set active and the process exits the routine. If NO, the process goes to step 808 which checks if the reception is a frame of the type PC desynchronization acknowledgment (54). If YES, the process goes to step 812 where the software variable DE-SYNC-PC-ACK is 30 set active and the process exits the routine. If NO, the process goes to step 810 which handles an erroneous received frame and then the process exits the routine.

It is to be noted that steps 804, 806 and 808 could be operated in another sequence without changing the functionality of the routine.

Figures 9-a, 9-b and 9-c depict one embodiment of an interrupt
5 routine flowchart of the infrared PC serial communication controller in accordance with the present invention. This process is executed each time the PC serial communication controller transmits or receives a frame.

On block 900 of figure 9-a, the processor of the PC reads the
10 interrupt control register to determine the source of the interrupt signal. Branch Yes of block 902 determines a receive interruption while branch NO determines a transmit interruption. From branch YES, the process goes to step 904 to check if a synchronization frame of the type 50 has been
15 received from the mouse device. If YES, a timer is initialized on block 906 to allow the PCs which are not selected to exit the routine when the timer elaps. Next, according to a time out decision block 908, the process either exits the routine (branch YES) or jumps to block 910 which checks a keyboard
20 input from the user. If the user presses the 'S' key, then the process goes to step 912 otherwise the process loops to step 908. It should be noted at this stage that the user must enter the 'S' key to synchronize the device mouse with the selected computer system. On block 912, the timer is reset and the PC
25 serial communication controller sends on block 914 the PC synchronization acknowledgment frame (53) to the mouse device. Next, a software variable PC synchronization (SYNC-PC) is set on block 916, and finally, the routine ends.

30 Going back to step 904 with reference to figure 9-b, if the received frame is not a synchronization frame, block 918 checks if the frame is a device driver frame (52). If YES, the process installs and activates the received device driver on step 920. The PC serial communication controller sends on block

922 the PC device driver acknowledgment frame (55) to the mouse device. Next, a software variable PC device driver (DD-PC) is set on block 924, and finally, the routine ends.

Going back to step 918, if the received frame is not a device driver frame, block 926 checks if the frame is a desynchronization frame (51). If YES, a timer is initialized on block 928 to allow the PCs which are not selected to exit the routine when the timer elaps. Next, according to a time out decision block 930, the process either exits the routine (branch YES) or jumps to block 932 which checks a keyboard input from the user. If the user presses the 'D' key, then the process goes to step 934 otherwise the process loops to step 930. It should be noted at this stage that the user must enter the 'D' key to desynchronize the device mouse with the selected computer system. On block 934, the timer is reset and the PC serial communication controller sends on block 936 the PC desynchronization acknowledgment frame (54) to the mouse device. Next, a software variable PC desynchronization (DE-SYNC-PC) is set on block 938 and finally the routine ends.

20 Going back to step 926, if the received frame is not a desynchronization frame (branch NO), the routine ends and the system enters the common wireless communication protocol for reception as previously defined on block 68 of figure 6.

25 Going back again to block 902 with reference to figure 9-c, branch NO determines a transmit interruption. The process goes to step 940 to check if the PC synchronization acknowledgment frame of the type 53 has been sent to the mouse device. If YES, the software variable SYNC-PC is reset at step 942 and the routine ends. If NO, the process goes to step 944 to check if the PC device driver acknowledgment frame of the type 55 has been sent to the mouse device. If YES, the software variable DD-PC is reset at step 946 and the routine ends. If NO, the process goes to step 948 to check if the PC

desynchronization acknowledgment frame of the type 54 has been sent to the mouse device. If YES, the software variable DE-SYNC-PC is reset at step 950 and the routine ends. If NO, the routine ends and the system enters the common wireless communication protocol for transmission as previously defined on block 68 of figure 6.

Although specific embodiments of the present invention have been illustrated in the accompanying drawings and described in the foregoing detailed description, it will be understood that the invention is not limited to the particular embodiments described herein, but is capable of numerous rearrangements, modifications and substitutions without departing from the scope of the invention. The following claims are intended to encompass all such modifications.

Claims

1. A peripheral device 10 for establishing wireless connection with an intelligent device and comprising means (26,32,34,41) for controlling a cursor displacement on a display screen of the intelligent device, the peripheral device further comprising:
 - means (20,40,42,43,44,45) coupled to the controlling means for generating a plurality of identification signals, said identification signals comprising at least a device driver signal identifying the device driver associated to said peripheral device, and
 - means (20,40,42,43,44,45) coupled to the controlling means for analyzing at least one acknowledgment signal received from said intelligent device in response to said plurality of identification signals, and setting said peripheral device in an active wireless mode.
2. The peripheral device of claim 1 wherein the generating means comprises first actuation means (20) to be actuated by a user during the generation of the plurality of identification signals.
3. The peripheral device of claim 2 wherein the first actuation means is a push button located external to the peripheral device.
4. The peripheral device of claims 1 to 3 wherein the peripheral device is a wireless mouse device (10).
5. The peripheral device of anyone of claims 1 to 4 wherein the generating means further comprises means for generating the

plurality of identification signals according to a predetermined sequence.

6. The peripheral device of the claim 5 wherein the first generated identification signal is a synchronization signal.
- 5 7. The peripheral device of anyone of claims 1 to 6 wherein the means for analyzing the acknowledgement signals further comprises means for setting the peripheral device in a state corresponding to the received acknowledgment signals.
8. The peripheral device of anyone of claims 1 to 7 wherein the
10 acknowledgment signals comprises at least a device driver acknowledgement signal meaning that the intelligent device has stored the device driver of the peripheral device.
9. The peripheral device of anyone of claims 1 to 8 wherein the
15 intelligent device comprising means (S-key,D-key,22) for processing the plurality of identification signals received from said wireless peripheral device.
10. The peripheral device of the preceeding claim wherein the
20 processing means further comprises second actuation means (S-key,D-key) to be actuated by a user during the generation of the plurality of the identification signals.
11. The peripheral device of the preceeding claim wherein the second actuation means are predefined keyboard keys of a keyboard connected to said intelligent device.
- 25 12. The peripheral device of the preceeding claim wherein the processing means further comprises means for selecting a specific routine according to the first identification signal received from said wireless peripheral device and means for generating at least one acknowledgment signal.

13. The peripheral device of anyone of claims 1 to 12 further comprising means for indicating to the user once at least one of the plurality of acknowledgment signals has been received by said peripheral device.
- 5 14. The peripheral device of the preceeding claim wherein the indication to the user is the activation or the desactivation of the cursor on the display screen of said intelligent device.
- 10 15. The peripheral device of anyone of claims 1 to 14 wherein said intelligent device is a computer system.
16. The peripheral device of anyone of claims 1 to 15 wherein the plurality of identification signals is in the form of HDLC frames including a control data stream for defining the
15 respective identification frame.
17. The peripheral device of anyone of claims 1 to 16 wherein the plurality of acknowledgment signals is in the form of HDLC frames including a control data stream for defining the respective acknowledgment frame.
- 20 18. In a multi-computers environment having a plurality of intelligent devices, a system for establishing wireless connection between a peripheral device and one of the plurality of intelligent devices, comprising:
- at least one peripheral device having means according
25 to anyone of claims 1 to 17, and
 - at least one intelligent device having means according to anyone of claims 1 to 17.

19.A method for establishing wireless connection between a peripheral device and at least one of a plurality of intelligent devices, comprising the steps of:

- 5 • generating from the peripheral device a plurality of identification signals, said identification signals comprising at least a device driver signal identifying the device driver associated to said wireless peripheral device,
- 10 • generating from the at least one intelligent device at least one acknowledgment signal in response to the plurality of identification signals, and
- analyzing within the peripheral device the at least one acknowledgment signal to set the peripheral device in a wireless active mode.

15 20.The method of claim 19 further comprising a first steps of:

- actuating a first actuation means coupled to said peripheral device, and
- simultaneously, actuating a second actuation means coupled to the at least one of the plurality of
20 intelligent devices.

21.The method of claim 20 wherein the first actuation means is a push button located external to the peripheral device.

22.The method of claim 21 wherein the second actuation means is a predefined keyboard key of keyboard of the at least one of
25 the plurality of intelligent devices.

23.The method of claims 19 to 22 wherein said peripheral device is a wireless mouse device (10).

24. The method of claims 19 to 23 wherein the step of generating a plurality of identification signals further comprising the steps of generating a first synchronization signal.

25. The method of claims 19 to 24 wherein the step of analyzing
5 the at least one acknowledgement signal further comprising the steps of setting a register of the peripheral device in a state corresponding to the received acknowledgment signals.

26. The method of claims 19 to 25 wherein the step of generating
10 at least one acknowledgment signal further comprising the steps of storing the device driver address within a storage area of the at least one intelligent device.

27. The method of claims 19 to 26 further comprising the steps
15 of signalling the user once at least one of the plurality of acknowledgment signals has been received by said peripheral device.

28. The method of claims 19 to 27 wherein the at least one intelligent device is a computer system.

20 29. The method of claims 19 to 28 wherein the plurality of identification signals is in the form of HDLC frames including a control data stream for defining the respective identification frame.

30. The method of claims 19 to 29 wherein the plurality of
25 acknowledgment signals is in the form of HDLC frames including a control data stream for defining the respective acknowledgment frame.