

INTERFACE DEVICES, METHODS FOR CONTROLLING AN INTERFACE DEVICE, AND COMPUTER-READABLE MEDIA

Technical Field

[0001] Various embodiments generally relate to interface devices, methods for controlling an interface device, and computer-readable media.

Background

[0002] Most of the cordless mice in the market are not able to achieve true 1000Hz polling rate. As such, there may be a need for devices and methods to facilitate achieving true 1000Hz polling rate.

Summary of the Invention

[0003] According to various embodiments, an interface device may be provided. The interface device may include: an interface to a host computer; a wireless interface to a peripheral device; a difference determination circuit configured to determine a difference between timeslots in the host computer and time slots in the peripheral device; and a notification circuit configured to notify the peripheral device about the difference if the determined difference is larger than a pre-determined threshold. According to various embodiments, the threshold may be determined by the resolution of a computing circuit (and for example not by the host computer). A good computing circuit (for example the interface device, for example a universal serial bus (USB) dongle, which may be connected to the host computer, for example via the interface) may get better than 1 ppm resolution, and for practical purposes, the threshold may be in a range of 10 ppm to 100 ppm.

[0004] According to various embodiments, a method for controlling an interface device may be provided. The method may include: providing an interface to a host

computer; providing a wireless interface to a peripheral device; determining a difference between timeslots in the host computer and time slots in the peripheral device; and notifying the peripheral device about the difference if the determined difference is larger than a pre-determined threshold.

[0005] According to various embodiments, a computer-readable medium may be provided. The computer-readable medium may include instructions which, when executed by a computer, make the computer perform a method for controlling an interface device. The method may include: providing an interface to a host computer; providing a wireless interface to a peripheral device; determining a difference between timeslots in the host computer and time slots in the peripheral device; and notifying the peripheral device about the difference if the determined difference is larger than a pre-determined threshold.

Brief Description of the Drawings

[0006] In the drawings, like reference characters generally refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of the invention. The dimensions of the various features or elements may be arbitrarily expanded or reduced for clarity. In the following description, various embodiments of the invention are described with reference to the following drawings, in which:

[0007] FIG. 1A shows an interface device according to various embodiments;

[0008] FIG. 1B shows a flow diagram illustrating a method for controlling an interface device according to various embodiments;

[0009] FIG. 2 shows a computing system according to various embodiments;

[0010] FIG. 3 shows a flow diagram illustrating processing in and communication between the transmitter and the receiver according to various embodiments, wherein the transmitter may be (or may be provided in) a wireless mouse, and wherein the receiver is a dongle interface to USB port;

[0011] FIG. 4 shows an illustration of clock signals according to various embodiments;

[0012] FIG. 5 shows an illustration of computing the computer clock offset against the receiver clock offset according to various embodiments;

[0013] FIG. 6 shows an illustration of tracking the long term frequency drift of the computer clock according to various embodiments; and

[0014] FIG. 7 shows an illustration of track the long term frequency drift of the transmitter clock according to various embodiments.

Detailed Description

[0015] The following detailed description refers to the accompanying drawings that show, by way of illustration, specific details and embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention. Other embodiments may be utilized and structural, and logical changes may be made without departing from the scope of the invention. The various embodiments are not necessarily mutually exclusive, as some embodiments can be combined with one or more other embodiments to form new embodiments.

[0016] In this context, the interface device as described in this description may include a memory which is for example used in the processing carried out in the interface device. A memory used in the embodiments may be a volatile memory, for example a DRAM (Dynamic Random Access Memory) or a non-volatile memory, for example a PROM (Programmable Read Only Memory), an EPROM (Erasable PROM), EEPROM (Electrically Erasable PROM), or a flash memory, e.g., a floating gate memory, a charge trapping memory, an MRAM (Magnetoresistive Random Access Memory) or a PCRAM (Phase Change Random Access Memory).

[0017] In an embodiment, a "circuit" may be understood as any kind of a logic implementing entity, which may be special purpose circuitry or a processor executing software stored in a memory, firmware, or any combination thereof. Thus, in an

embodiment, a "circuit" may be a hard-wired logic circuit or a programmable logic circuit such as a programmable processor, e.g. a microprocessor (e.g. a Complex Instruction Set Computer (CISC) processor or a Reduced Instruction Set Computer (RISC) processor). A "circuit" may also be a processor executing software, e.g. any kind of computer program, e.g. a computer program using a virtual machine code such as e.g. Java. Any other kind of implementation of the respective functions which will be described in more detail below may also be understood as a "circuit" in accordance with an alternative embodiment.

[0018] In the specification the term “comprising” shall be understood to have a broad meaning similar to the term “including” and will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps. This definition also applies to variations on the term “comprising” such as “comprise” and “comprises”.

[0019] The reference to any prior art in this specification is not, and should not be taken as an acknowledgement or any form of suggestion that the referenced prior art forms part of the common general knowledge in Australia (or any other country).

[0020] In order that the invention may be readily understood and put into practical effect, particular embodiments will now be described by way of examples and not limitations, and with reference to the figures.

[0021] Various embodiments are provided for devices, and various embodiments are provided for methods. It will be understood that basic properties of the devices also hold for the methods and vice versa. Therefore, for sake of brevity, duplicate description of such properties may be omitted.

[0022] It will be understood that any property described herein for a specific device may also hold for any device described herein. It will be understood that any property described herein for a specific method may also hold for any method described herein. Furthermore, it will be understood that for any device or method described herein, not necessarily all the components or steps described must be enclosed in the device or method, but only some (but not all) components or steps may be enclosed.

[0023] The term “coupled” (or “connected”) herein may be understood as electrically coupled or as mechanically coupled, for example attached or fixed, or just in contact without any fixation, and it will be understood that both direct coupling or indirect coupling (in other words: coupling without direct contact) may be provided.

[0024] Most of the cordless mice in the market are not able to achieve true 1000Hz polling rate (in other words: USB (universal serial bus) frequency).

[0025] In commonly used mice, there may be an offset between the operating clock frequency between the host system and the cordless mouse which includes all forms of wireless mouse or mouse operating without any cables to the host system.

[0026] According to various embodiments, this clock offset issue is resolved using an innovative and power efficient method. This may allow achieving a true 1000Hz polling rate.

[0027] A commonly used system setup includes a host computer, a receiver and a cordless transmitter (for example provided in a computer mouse). The receiver may be attached to the host computer via a USB connector. The transmitter may send the mouse movement and buttons status commands wirelessly to the receiver and the receiver may send the mouse commands to the host system.

[0028] In commonly used systems, the host system, receiver and the transmitter each use an independent crystal clock to drive their respective functions. The user may set a polling frequency on the transmitter (for example on the cordless mouse) and the mouse may sample the data based on its own clock frequency. The sampled data may be sent to the receiver based on the transmitter operating frequency. The receiver may buffer the received data from the transmitter and wait for the host system to read the data. The host system may have its own clock frequency, and the data may be read based on the host system clock frequency.

[0029] Overflow and underflow of data may happen since the transmitter is not synchronized with the host system. The allowable USB frequency offset for a host system may be +/- 500ppm, which translates to 1 sample overflow/underflow for every 2 seconds under the worst case scenario. This means that a typical wireless mouse may never

achieve a steady polling frequency claimed versus a wired mouse where data is polled by host computer directly.

[0030] A commonly used way of achieving tracking of clock frequency from a host computer may be using a Phase Lock Loop circuit at the mouse transmitter side which may constantly adjust clock jitter and drift. However, this implementation may consume much higher power, because the mouse transmitter may have to constantly have the wireless circuit turned on to track the host computer USB synchronization clock typically the SOF (Start of Frame) signal.

[0031] According to various embodiments, devices and methods are provided to resolve this deficiency of currently used cordless mouse design. Various embodiments may reduce power and may thus provide a low power mode concept.

[0032] According to various embodiments, systems and methods to synchronize cordless mouse transmission clock with computer USB clock may be provided.

[0033] FIG. 1A shows an interface device 100 according to various embodiments. The interface device 100 may include an interface 102 (for example a first interface; for example a wired interface; for example a host computer interface) to a host computer (not shown in FIG. 1A). The interface device 100 may further include a wireless interface 104 to a peripheral device (not shown in FIG. 1A). The interface device 100 may further include a difference determination circuit 106 configured to determine a difference between timeslots (for example a clock signal) in the host computer and time slots (for example a clock signal) in the peripheral device. The interface device 100 may further include a notification circuit 108 configured to notify the peripheral device about the difference if the determined difference is larger than a pre-determined threshold. The interface 102, the wireless interface 104, the difference determination circuit 106, and the notification circuit 108 may be coupled with each other, like indicated by lines 110, for example electrically coupled, for example using a line or a cable, and/ or mechanically coupled.

[0034] In other words, the interface device 100 may monitor timeslots (or clock shifts) between the host computer and the peripheral device, and may inform the

peripheral device about a difference in timeslots (or about at clock shift), so that the peripheral device can update its internal clock accordingly.

[0035] According to various embodiments, the interface 102 may include or may be or may be included in a universal serial bus interface.

[0036] According to various embodiments, the difference determination circuit 106 may further be configured to determine the difference upon a power up of the interface device 100.

[0037] According to various embodiments, the notification circuit 108 may be configured to notify the peripheral device about the difference upon a power up of the interface device 100.

[0038] According to various embodiments, the pre-determined threshold may be based on the timeslots in the host computer.

[0039] According to various embodiments, the pre-determined threshold may be based on the timeslots in the peripheral device.

[0040] According to various embodiments, the pre-determined threshold may be in a range of 10 ppm to 50 ppm.

[0041] According to various embodiments, the difference determination circuit 106 may be configured to determine the difference based on a start of frame signal.

[0042] According to various embodiments, the interface device 100 may include or may be or may be included in a universal serial bus dongle.

[0043] According to various embodiments, the peripheral device may include or may be or may be included in a cordless mouse.

[0044] FIG. 1B shows a flow diagram 112 illustrating a method for controlling an interface device according to various embodiments. In 114, an interface to a host computer may be provided. In 116, a wireless interface to a peripheral device may be provided. In 118, a difference between timeslots in the host computer and time slots in the peripheral device may be determined. In 120, the peripheral device may be notified about the difference if the determined difference is larger than a pre-determined threshold.

[0045] According to various embodiments, the interface may include or may be or may be included in a universal serial bus interface.

[0046] According to various embodiments, the difference may be determined upon a power up of the interface device.

[0047] According to various embodiments, the method may further include notifying the peripheral device about the difference upon a power up of the interface device.

[0048] According to various embodiments, the pre-determined threshold may be based on the timeslots in the host computer.

[0049] According to various embodiments, the pre-determined threshold may be based on the timeslots in the peripheral device.

[0050] According to various embodiments, the pre-determined threshold may be in a range of 10 ppm to 50 ppm.

[0051] According to various embodiments, the difference may be determined based on a start of frame signal (for example of a USB protocol).

[0052] According to various embodiments, the interface device may include or may be or may be included in a universal serial bus dongle.

[0053] According to various embodiments, the peripheral device may include or may be or may be included in a cordless mouse.

[0054] According to various embodiments, a computer-readable medium may be provided. The computer-readable medium may include instructions which, when executed by a computer, make the computer perform a method for controlling an interface device. The method may include: providing an interface to a host computer; providing a wireless interface to a peripheral device; determining a difference between timeslots in the host computer and time slots in the peripheral device; and notifying the peripheral device about the difference if the determined difference is larger than a pre-determined threshold.

[0055] According to various embodiments, the interface may include or may be or may be included in a universal serial bus interface.

[0056] According to various embodiments, the difference may be determined upon a power up of the interface device.

[0057] According to various embodiments, the method may further include notifying the peripheral device about the difference upon a power up of the interface device.

[0058] According to various embodiments, the pre-determined threshold may be based on the timeslots in the host computer.

[0059] According to various embodiments, the pre-determined threshold may be based on the timeslots in the peripheral device.

[0060] According to various embodiments, the pre-determined threshold may be in a range of 10 ppm to 50 ppm.

[0061] According to various embodiments, the difference may be determined based on a start of frame signal.

[0062] According to various embodiments, the interface device may include or may be or may be included in a universal serial bus dongle.

[0063] According to various embodiments, the peripheral device may include or may be or may be included in a cordless mouse.

[0064] FIG. 2 shows a computing system 200 according to various embodiments. The computing system 200 may include a host computer 202, a receiver 206 and a cordless transmitter 210 (for example provided in a computer mouse). The receiver 206 may be attached to the host computer 202 via a USB connector 204. The transmitter 210 may send the mouse movement and buttons status commands wirelessly (like indicated by arrows 208) to the receiver 206, and the receiver 206 may send the mouse commands to the host system 210 via the USB connector 204.

[0065] According to various embodiments, a power efficient method may be provided to synchronize the transmitter (cordless mouse) polling rate to the computer USB clock frequency.

[0066] According to various embodiments, three methods may be provided to track the computer clock and transmitter clock:

[0067] In a first method, the clock offset between the host computer and the receiver may be computed upon power up. The first method may be referred to as USB clock offset tracking.

[0068] In a second method, the computer clock may be tracked during active operation. The second method may be referred to as active tracking.

[0069] In a third method, the transmitter clock may be tracked during active operation. The third method may be referred to as TX (transmitter) tracking.

[0070] FIG. 3 shows a flow diagram 300 illustrating processing in and communication between the transmitter and the receiver according to various embodiments. Processing in the transmitter is indicated on the left side 302 of FIG. 3. Processing in the receiver is indicated on the right side 304 of FIG. 3. In 312, the receiver may perform USB clock offset tracking. In 306, the transmitter may announce its presence. In 316, the receiver may scan for a transmitter. As a result of USB clock offset tracking (as carried out in 312), an offset factor may (in 314) be sent to the transmitter (TX) for timeslot interval calibration. In 308, the transmitter may perform a joining procedure. In 318, the receiver may perform a joining procedure. In 310, the transmitter may be in active mode. In 320, the receiver may be in active mode. In 322, the receiver may perform USB clock long term drift tracking. In 324, the receiver may perform TX clock tracking. As a result of USB clock long term drift tracking and TX clock tracking, the receiver may in 326 send an offset factor to the transmitter for clock adjustment.

[0071] FIG. 4 shows an illustration 400 of clock signals according to various embodiments. The “heartbeat” of the receiver and of the transmitter is the respective periodic timeslot (in other words the respective clock signal). The timeslot (or clock signal) may drive the functions and states of receiver (for example USB dongle) and of the transmitter (for example provided in a wireless mouse). RX timeslots 402 and TX timeslots 404 are shown. RX timeslots 410 during scanning and joining phase may be longer than RX timeslots 412 during active phase 408. TX timeslots 414 during scanning and joining phase may be longer than TX timeslots 416 during active phase 408.

[0072] FIG. 5 shows an illustration 500 of computing the computer clock offset against the receiver clock offset according to various embodiments. A plurality of USB SOF 502 is shown, which are space 1 ms +/- offset space, like indicated by 504. An N-cycle of SOF 506 is shown. A plurality of RX clock signals 508, and a total number 510

of RX clock cycle, amounting to a period of time T_0 , are shown. A constant offset F_0 (based on USB SOF interval) may be computed as follows:

$$F_0 = T_0 / N.$$

[0073] FIG. 6 shows an illustration 600 of tracking the long term frequency drift of the computer clock according to various embodiments. A plurality of USB SOF 602, and an enlarged view of USB SOF 604 of an N-cycle 606 of SOF are shown. A plurality of RX clock signals 608, and a total number 610 of RX clock cycle, amounting to a period of time T_1 , are shown. An average USB SOF interval F_1 may be computed as follows:

$$F_1 = T_1 / N.$$

[0074] An average frequency drift s may be computed as follows:

$$s = |F_0 - F_1|.$$

[0075] FIG. 7 shows an illustration 700 of track the long term frequency drift of the transmitter clock according to various embodiments. According to various embodiments, the offset 706 between the RX timeslot 702 and the TX timeslot 704 may be tracked. A plurality of RX clock signals 708 is shown.

[0076] According to various embodiments, the USB dongle may compute the delta between the USB clock and the dongle clock, and may send this delta to the mouse to set its operating clock. The dongle may track the USB clock and may simultaneously track the mouse clock against its own clock. The dongle may send an adjustment factor to the mouse when the delta is greater than a predefined threshold.

[0077] Various embodiments may provide a low power design, as the tracking process is performed in USB dongle which is powered by the PC. The mouse (which may act as a passive device) may merely need to adjust its clock when it is told to do so.

[0078] Various embodiments may provide high accuracy, as the USB dongle (as a centralized device) may track the clock frequency difference between the PC clock and the mouse clock.

[0079] According to various embodiments, the mouse does not need to track the clock of the (USB) dongle and the host. The tracking process may be performed in the (USB) dongle, which may draw power from the PC host. Various embodiments may

provide a low power design (for example for battery powered devices), as the power consumption in the (cordless) device (for example the mouse) is minimized.

[0080] According to various embodiments, a true 1000Hz polling rate may be achieved for a cordless mouse synchronized to a computer clock.

[0081] Various embodiments provide a power efficient tracking method as the tracking process is performed by the receiver which draws power from a computer USB port. In other words: power in a cordless mouse (in other words: wireless mouse) may be saved.

[0082] According to various embodiments, the overflow/underflow problem may be eliminated without dropping/adding any data sample.

[0083] According to various embodiments, a wireless tracking mouse system may be provided.

[0084] The following examples pertain to further embodiments.

[0085] Example 1 is an interface device comprising: an interface to a host computer; a wireless interface to a peripheral device; a difference determination circuit configured to determine a difference between timeslots in the host computer and time slots in the peripheral device; and a notification circuit configured to notify the peripheral device about the difference if the determined difference is larger than a pre-determined threshold.

[0086] In example 2, the subject-matter of example 1 can optionally include that the interface comprises a universal serial bus interface.

[0087] In example 3, the subject-matter of any one of examples 1 to 2 can optionally include that the difference determination circuit is further configured to determine the difference upon a power up of the interface device.

[0088] In example 4, the subject-matter of any one of examples 1 to 3 can optionally include that the notification circuit is configured to notify the peripheral device about the difference upon a power up of the interface device.

[0089] In example 5, the subject-matter of any one of examples 1 to 4 can optionally include that the pre-determined threshold is based on the timeslots in the host computer.

[0090] In example 6, the subject-matter of any one of examples 1 to 5 can optionally include that the pre-determined threshold is based on the timeslots in the peripheral device.

[0091] In example 7, the subject-matter of any one of examples 1 to 6 can optionally include that the pre-determined threshold is in a range of 10 ppm to 50 ppm.

[0092] In example 8, the subject-matter of any one of examples 1 to 7 can optionally include that the difference determination circuit is configured to determine the difference based on a start of frame signal.

[0093] In example 9, the subject-matter of any one of examples 1 to 8 can optionally include that the interface device comprises a universal serial bus dongle.

[0094] In example 10, the subject-matter of any one of examples 1 to 9 can optionally include that the peripheral device comprises a cordless mouse.

[0095] Example 11 is a method for controlling an interface device, the method comprising: providing an interface to a host computer; providing a wireless interface to a peripheral device; determining a difference between timeslots in the host computer and time slots in the peripheral device; and notifying the peripheral device about the difference if the determined difference is larger than a pre-determined threshold.

[0096] In example 12, the subject-matter of example 11 can optionally include that the interface comprises a universal serial bus interface.

[0097] In example 13, the subject-matter of any one of examples 11 to 12 can optionally include that the difference is determined upon a power up of the interface device.

[0098] In example 14, the subject-matter of any one of examples 11 to 13 can optionally include notifying the peripheral device about the difference upon a power up of the interface device.

[0099] In example 15, the subject-matter of any one of examples 11 to 14 can optionally include that the pre-determined threshold is based on the timeslots in the host computer.

[00100] In example 16, the subject-matter of any one of examples 11 to 15 can optionally include that the pre-determined threshold is based on the timeslots in the peripheral device.

[00101] In example 17, the subject-matter of any one of examples 11 to 16 can optionally include that the pre-determined threshold is in a range of 10 ppm to 50 ppm.

[00102] In example 18, the subject-matter of any one of examples 11 to 17 can optionally include that the difference is determined based on a start of frame signal.

[00103] In example 19, the subject-matter of any one of examples 11 to 18 can optionally include that the interface device comprises a universal serial bus dongle.

[00104] In example 20, the subject-matter of any one of examples 11 to 19 can optionally include that the peripheral device comprises a cordless mouse.

[00105] Example 21 is a computer-readable medium comprising instructions which, when executed by a computer, make the computer perform a method for controlling an interface device, the method comprising: providing an interface to a host computer; providing a wireless interface to a peripheral device; determining a difference between timeslots in the host computer and time slots in the peripheral device; and notifying the peripheral device about the difference if the determined difference is larger than a pre-determined threshold.

[00106] In example 22, the subject-matter of example 21 can optionally include that the interface comprises a universal serial bus interface.

[00107] In example 23, the subject-matter of any one of examples 21 to 22 can optionally include that the difference is determined upon a power up of the interface device.

[00108] In example 24, the subject-matter of any one of examples 21 to 23 can optionally include instructions which, when executed by a computer, make the computer perform: notifying the peripheral device about the difference upon a power up of the interface device.

[00109] In example 25, the subject-matter of any one of examples 21 to 24 can optionally include that the pre-determined threshold is based on the timeslots in the host computer.

[00110] In example 26, the subject-matter of any one of examples 21 to 25 can optionally include that the pre-determined threshold is based on the timeslots in the peripheral device.

[00111] In example 27, the subject-matter of any one of examples 21 to 26 can optionally include that the pre-determined threshold is in a range of 10 ppm to 50 ppm.

[00112] In example 28, the subject-matter of any one of examples 21 to 27 can optionally include that the difference is determined based on a start of frame signal.

[00113] In example 29, the subject-matter of any one of examples 21 to 28 can optionally include that the interface device comprises a universal serial bus dongle.

[00114] In example 30, the subject-matter of any one of examples 21 to 29 can optionally include that the peripheral device comprises a cordless mouse.

[00115] While the invention has been particularly shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced.

CLAIMS

1. An interface device comprising:
 - an interface to a host computer;
 - a wireless interface to a peripheral device;
 - a difference determination circuit configured to:
 - determine a first clock offset between the host computer and the interface device,
 - determine a frequency drift based on a clock of the host computer and a clock of the interface device,
 - determine a second clock offset between the interface device and the peripheral device, and
 - determine a difference between timeslots in the host computer and timeslots in the peripheral device; and
 - a notification circuit configured to :
 - send a first offset factor based on the first clock offset, between the host computer and the interface device, to the peripheral device,
 - send a second offset factor based on both the frequency drift, based on the clock of the host computer and the clock of the interface device, and the second clock offset, between the interface device and the peripheral device, to the peripheral device, and
 - notify the peripheral device about the determined difference if the determined difference is larger than a pre-determined threshold.
2. The interface device of claim 1,
 - wherein the interface comprises a universal serial bus interface.
3. The interface device of any one of claims 1 to 2,
 - wherein the difference determination circuit is further configured to determine the difference upon a power up of the interface device.
4. The interface device of any one of claims 1 to 3,

wherein the notification circuit is configured to notify the peripheral device about the difference upon a power up of the interface device.

5. The interface device of any one of claims 1 to 4,
wherein the pre-determined threshold is based on at least one of the timeslots in the host computer and the timeslots in the peripheral device.
6. The interface device of any one of claims 1 to 5,
wherein the pre-determined threshold is in a range of 10 ppm to 50 ppm.
7. The interface device of any one of claims 1 to 6,
wherein the difference determination circuit is configured to determine the difference based on a start of frame signal.
8. The interface device of any one of claims 1 to 7,
wherein the interface device comprises a universal serial bus dongle.
9. The interface device of any one of claims 1 to 8,
wherein the peripheral device comprises a cordless mouse.
10. A method for controlling an interface device, the method comprising:
providing an interface to a host computer;
providing a wireless interface to a peripheral device;
determining a first clock offset between the host computer and the interface device;
determine a frequency drift based on a clock of the host computer and a clock of the interface device;
determining a second clock offset between the interface device and the peripheral device;
determining a difference between timeslots in the host computer and timeslots in the peripheral device; and

sending a first offset factor based on the first clock offset, between the host computer and the interface device, to the peripheral device;

sending a second offset factor based on both the frequency drift, based on the clock of the host computer and the clock of the interface device, and the second clock offset, between the interface device and the peripheral device, to the peripheral device; and

notifying the peripheral device about the determined difference if the determined difference is larger than a pre-determined threshold.

11. The method of claim 10,
wherein the difference is determined upon a power up of the interface device.
12. The method of any one of claims 10 to 11, further comprising:
notifying the peripheral device about the difference upon a power up of the interface device.
13. The method of any one of claims 10 to 12,
wherein the pre-determined threshold is based on at least one of the timeslots in the host computer and the timeslots in the peripheral device.
14. The method of any one of claims 10 to 13,
wherein the difference is determined based on a start of frame signal.
15. A non-transitory computer-readable medium comprising instructions which, when executed by a computer, make the computer perform a method for controlling an interface device, the method comprising:
providing an interface to a host computer;
providing a wireless interface to a peripheral device;
determining a first clock offset between the host computer and the interface device;
determine a frequency drift based on a clock of the host computer and a clock of the interface device;

determining a second clock offset between the interface device and the peripheral device;

determining a difference between timeslots in the host computer and timeslots in the peripheral device; and

sending a first offset factor based on the first clock offset, between the host computer and the interface device, to the peripheral device;

sending a second offset factor based on both the frequency drift, based on the clock of the host computer and the clock of the interface device, and the second clock offset, between the interface device and the peripheral device, to the peripheral device; and

notifying the peripheral device about the determined difference if the determined difference is larger than a pre-determined threshold.

16. The non-transitory computer-readable medium of claim 15,
wherein the interface comprises a universal serial bus interface.
17. The non-transitory computer-readable medium of any one of claims 15 to 16,
wherein the difference is determined upon a power up of the interface device.
18. The non-transitory computer-readable medium of any one of claims 15 to 17,
further comprising instructions which, when executed by a computer, make the computer perform:
notifying the peripheral device about the difference upon a power up of the interface device.
19. The non-transitory computer-readable medium of any one of claims 15 to 18,
wherein the pre-determined threshold is based on at least one of the timeslots in the host computer and the timeslots in the peripheral device.
20. The non-transitory computer-readable medium of any one of claims 15 to 19,
wherein the difference is determined based on a start of frame signal.