

HOST DEVICE AND METHOD FOR COMMUNICATING DATA WITH PERIPHERAL DEVICE

Technical Field

[0001] This disclosure relates to a method for communicating data between a host device (e.g., a laptop computer, or a smart phone, etc) and at least one peripheral device (e.g., a keyboard, microphone, a camera, or a gamepad, etc). This disclosure also relates to a host device configured to communicate data with at least one peripheral device.

Background

[0002] Universal Serial Bus (USB) provides a standard, and low-cost interface for data transferring between host devices (e.g., a laptop computer, or a smart phone, etc) and peripheral devices (e.g., a keyboard, microphone, a camera, or a gamepad, etc).

[0003] Therefore, there exists a need for providing an improved method for communicating data between a host device and a peripheral device connected to the host device.

Summary

[0004] According to a first aspect of the present disclosure, a method for communicating data between a host device and a target peripheral device connected to the host device is provided. The method may comprise: receiving at least one identifier of the target peripheral device; generating a configuration key of the target peripheral device based on the at least one identifier of the target peripheral device; determining if the configuration key of the target peripheral device is in a look-up table, the look-up table comprising configuration keys for a plurality of peripheral devices and configuration data corresponding to each configuration key; obtaining an interface name of a USB interface of the target peripheral device and obtaining configuration data for the target peripheral device based on the interface name of the USB interface of the target peripheral device, when the configuration key of the target peripheral device is absent in the look-up table; and communicating data between the host device and the target peripheral device according to the configuration data for the target peripheral device.

[0005] According to a second aspect of the present disclosure, a host device configured for communicating data with a target peripheral device connected to the host device is provided.

The host device may comprise: a host interface configured to connect with the target peripheral device; and a processor coupled to the host interface, wherein the processor is configured to: receive at least one identifier of the target peripheral device; generate a configuration key of the target peripheral device based on the at least one identifier of the target peripheral device; determine if the configuration key of the target peripheral device is in a look-up table, the look-up table comprising configuration keys for a plurality of peripheral devices and configuration data corresponding to each configuration key; obtain an interface name of a USB interface of the target peripheral device and obtain configuration data for the target peripheral device based on the interface name of the USB interface of the target peripheral device, when the configuration key of the target peripheral device is absent in the look-up table, and communicate data between the host device and the target peripheral device according to the configuration data for the target peripheral device.

Brief Description of the Drawings

[0006] In the drawings, like reference characters generally refer to like parts throughout the different views. The drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of the disclosure. In the following description, various embodiments of the disclosure are described with reference to the following drawings, in which:

[0007] FIG. 1A shows a schematic illustration of a host device configured to communicate data with a target peripheral device, where the target peripheral device is not connected to the host device, in accordance with some embodiments of the present disclosure;

[0008] FIG. 1B shows the host device of FIG. 1A configured to communicate data with a target peripheral device, where the target peripheral device is connected to the host device, in accordance with some embodiments of the present disclosure;

[0009] FIG. 2 shows a schematic illustration of the host device configured to communicate data with a target peripheral device;

[0010] FIG. 3 is a flow chart depicting a method of communicating data between the host device and the target peripheral device, according to some embodiments of the present disclosure;

[0011] FIG. 4 is a flow chart depicting an exemplary method of communicating data between the host device and the target peripheral device;

[0001] FIG. 5 shows an exemplary process of communicating data between the host device and the target peripheral device according to the configuration data for the target peripheral device;

[0002] FIG. 6 is a flow chart depicting a method of communicating data between a host device and a target peripheral device connected to the host device, according to some embodiments of the present disclosure;

[0003] FIG. 7 is a flow chart depicting the step of obtaining the interface name of the USB interface of the target peripheral device;

[0004] FIG. 8A shows a schematic illustration of a host device configured to communicate data with a plurality of target peripheral devices, where the plurality of target peripheral devices are not connected to the host device, in accordance with some embodiments of the present disclosure;

[0005] FIG. 8B shows the host device of FIG. 8A configured to communicate data with a plurality of target peripheral devices, where the plurality of target peripheral devices are connected to the host device, in accordance with some embodiments of the present disclosure;

[0006] FIG. 9 shows a schematic illustration of the host device configured to communicate data with a plurality of target peripheral devices.

Detailed Description

[0007] The following detailed description refers to the accompanying drawings that show, by way of illustration, specific details, and embodiments in which the disclosure may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the disclosure. Other embodiments may be utilized, and structural, logical, optical and electrical changes may be made without departing from the scope of the disclosure. The various embodiments are not necessarily mutually exclusive, as some embodiments can be combined with one or more other embodiments to form new embodiments.

[0008] Embodiments described in the context of one of the methods or devices are analogously valid for the other methods or devices. Similarly, embodiments described in the context of a method are analogously valid for a device, and vice versa.

[0009] Features that are described in the context of an embodiment may correspondingly be applicable to the same or similar features in the other embodiments. Features that are described in the context of an embodiment may correspondingly be applicable to the other embodiments, even if not explicitly described in these other embodiments. Furthermore, additions and/or

combinations and/or alternatives as described for a feature in the context of an embodiment may correspondingly be applicable to the same or similar feature in the other embodiments.

[0010] It should be understood that the terms “on”, “over”, “top”, “bottom”, “down”, “side”, “back”, “left”, “right”, “front”, “back”, “lateral”, “side”, “up”, “down”, “vertical”, “horizontal” etc., when used in the following description are used for convenience and to aid understanding of relative positions or directions, and not intended to limit the orientation of any device, or structure or any part of any device or structure. In addition, the singular terms “a”, “an”, and “the” include plural references unless context clearly indicates otherwise. Similarly, the “or” is intended to include “and” unless the context clearly indicates otherwise.

[0011] It will be further understood that the terms “comprise” (and any form of comprise, such as “comprises” and “comprising”), “have” (and any form of have, such as “has” and “having”), “include” (and any form of include, such as “includes” and “including”), and “contain” (and any form of contain, such as “contains” and “containing”) are open-ended linking verbs. As a result, a method or device that “comprises,” “has,” “includes” or “contains” one or more steps or elements possesses those one or more steps or elements, but is not limited to possessing only those one or more steps or elements. Likewise, a step of a method or an element of a device that “comprises,” “has,” “includes” or “contains” one or more features possesses those one or more features, but is not limited to possessing only those one or more features. Furthermore, a device or structure that is configured in a certain way is configured in at least that way, but may also be configured in ways that are not listed.

[0012] Approximating language, as used herein throughout the specification and claims, may be applied to modify any quantitative representation that could permissibly vary without resulting in a change in the basic function to which it is related. Accordingly, a value modified by a term or terms, such as “about,” “substantially”, is not limited to the precise value specified but within tolerances that are acceptable for the operation of the embodiment for an application for which it is intended. In some instances, the approximating language may correspond to the precision of an instrument for measuring the value.

[0013] The term “exemplary” may be used herein to mean “serving as an example, instance, or illustration”. Any aspect or design described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other aspects or designs.

[0014] The terms “at least one” and “one or more” may be understood to include a numerical quantity greater than or equal to one (e.g., one, two, three, four, [...], etc.). The term “a plurality” may be understood to include a numerical quantity greater than or equal to two (e.g., two, three, four, five, [...], etc.). The phrase “at least one of” with regard to a group of elements may be

used herein to mean at least one element from the group consisting of the elements. For example, the phrase “at least one of” with regard to a group of elements may be used herein to mean a selection of: one of the listed elements, a plurality of one of the listed elements, a plurality of individual listed elements, or a plurality of a multiple of listed elements.

[0015] The words “plural” and “multiple” in the description and the claims expressly refer to a quantity greater than one. Accordingly, any phrases explicitly invoking the aforementioned words (e.g., “a plurality of (objects)”, “multiple (objects)”) referring to a quantity of objects expressly refer to more than one of the said objects. The terms “group (of)”, “set (of)”, “collection (of)”, “series (of)”, “sequence (of)”, “grouping (of)”, etc., and the like in the description and in the claims, if any, refer to a quantity equal to or greater than one, i.e. one or more.

[0016] The term “first”, “second”, “third” detailed herein are used to distinguish one element from another similar element and may not necessarily denote order or relative importance, unless otherwise stated.

[0017] As used herein, the phrase of the form of “at least one of A or B” may include A or B or both A and B. Correspondingly, the phrase of the form of “at least one of A or B or C”, or including further listed items, may include any and all combinations of one or more of the associated listed items.

[0018] The following examples pertain to various aspects of the present disclosure.

[0019] Example 1 is a method for communicating data between a host device and a target peripheral device connected to the host device, the method comprising: receiving at least one identifier of the target peripheral device; generating a configuration key of the target peripheral device based on the at least one identifier of the target peripheral device; determining if the configuration key of the target peripheral device is in a look-up table, the look-up table comprising configuration keys for a plurality of peripheral devices and configuration data corresponding to each configuration key; obtaining an interface name of a USB interface of the target peripheral device and obtaining configuration data for the target peripheral device based on the interface name of the USB interface of the target peripheral device, when the configuration key of the target peripheral device is absent in the look-up table; and communicating data between the host device and the target peripheral device according to the configuration data for the target peripheral device.

[0020] In Example 2, the subject matter of Example 1 may optionally include that obtaining the interface name of the USB interface of the target peripheral device comprising: obtaining a raw descriptor for the USB interface of the target peripheral device; parsing the raw descriptor

to obtain a string index for the USB interface; and obtaining the interface name of the USB interface based on the string index.

[0021] In Example 3, the subject matter of Example 2 may optionally include that the interface name of the USB interface is obtained based on the string index using Human Interface Device (HID) instructions.

[0022] In Example 4, the subject matter of Example 2 may optionally include that obtaining the raw descriptor for the USB interface of the target peripheral device comprises: applying for USB permission to access the target peripheral device to obtain the raw descriptor for the USB interface of the target peripheral device.

[0023] In Example 5, the subject matter of Example 1 may optionally include enumerating a plurality of USB interfaces of the target peripheral device, obtaining an interface name of each of the plurality of USB interfaces of the target peripheral device, and obtaining configuration data for the target peripheral device based on the interface names of the plurality of USB interfaces of the target peripheral device, when the configuration key of the target peripheral device is absent in the look-up table.

[0024] In Example 6, the subject matter of Example 1 may optionally include that the configuration data for the target peripheral device comprises USB mode type, endpoint type, endpoint transmission size and direction of data for endpoint.

[0025] In Example 7, the subject matter of Example 6 may optionally include determining that the interface name of the USB interface of the target peripheral device supports haptic data.

[0026] In Example 8, the subject matter of Example 1 may optionally include obtaining configuration data of the target peripheral device according to the configuration key of the target peripheral device and the look-up table, when the configuration key of the target peripheral device is in the look-up table.

[0027] In Example 9, the subject matter of Example 7 may optionally include communicating haptic data between the host device and the peripheral device.

[0028] In Example 10, the subject matter of Example 1 may optionally include that the at least one identifier comprises vendor identification and product identification.

[0029] Example 11 is a host device configured for communicating data with a target peripheral device connected to the host device, the host device comprising: a host interface configured to connect with the target peripheral device; and a processor coupled to the host interface, wherein the processor is configured to: receive at least one identifier of the target peripheral device; generate a configuration key of the target peripheral device based on the at least one identifier of the target peripheral device; determine if the configuration key of the target peripheral device

is in a look-up table, the look-up table comprising configuration keys for a plurality of peripheral devices and configuration data corresponding to each configuration key; obtain an interface name of a USB interface of the target peripheral device and obtain configuration data for the target peripheral device based on the interface name of the USB interface of the target peripheral device, when the configuration key of the target peripheral device is absent in the look-up table, and communicate data between the host device and the target peripheral device according to the configuration data for the target peripheral device.

[0030] In Example 12, the subject matter of Example 11 may optionally include that the processor is further configured to: obtain a raw descriptor for the USB interface of the target peripheral device; parse the raw descriptor to obtain a string index for the USB interface; obtain the interface name of the USB interface based on the string index.

[0031] In Example 13, the subject matter of Example 12 may optionally include that the interface name of the USB interface is obtained based on the string index using Human Interface Device (HID) instructions.

[0032] In Example 14, the subject matter of Example 12 may optionally include that USB permission to access the target peripheral device is applied for to obtain the raw descriptor for the USB interface of the target peripheral device.

[0033] In Example 15, the subject matter of Example 11 may optionally include that the processor is further configured to enumerate a plurality of USB interfaces of the target peripheral device, obtain an interface name of each of the plurality of USB interfaces of the target peripheral device, and obtain configuration data for the target peripheral device based on the interface names of the plurality of USB interfaces of the target peripheral device, when the configuration key of the target peripheral device is absent in the look-up table.

[0034] In Example 16, the subject matter of Example 11 may optionally include that the configuration data for the target peripheral device comprises USB mode type, endpoint type, endpoint transmission size and direction of data for endpoint.

[0035] In Example 17, the subject matter of Example 16 may optionally include that the processor is further configured to: determine that the interface name of the USB interface of the target peripheral device supports haptic data.

[0036] In Example 18, the subject matter of Example 11 may optionally include that the processor is further configured to: obtain configuration data of the target peripheral device according to the configuration key of the target peripheral device and the look-up table, when the configuration key of the target peripheral device is in the look-up table.

[0037] In Example 19, the subject matter of Example 17 may optionally include that haptic data is communicated between the host device and the peripheral device.

[0038] In Example 20, the subject matter of Example 11 may optionally include that the at least one identifier comprises vendor identification and product identification.

[0039] In various embodiments, the host device and method for communicating data between the host device and the at least one target peripheral devices will now be described by way of the following non-limiting examples.

[0040] FIG. 1A shows a schematic illustration of a host device 100 configured to communicate data with a target peripheral device 10, where the target peripheral device 10 is not connected to the host device 100, in accordance with some embodiments of the present disclosure. FIG. 1B shows the host device 100 of FIG. 1A configured to communicate data with a target peripheral device 10, where the target peripheral device 10 is connected to the host device 100, in accordance with some embodiments of the present disclosure. The host device 100 may include a laptop computer, a smart phone, or a mobile phone etc. The target peripheral device 10 may include a keyboard, a mouse, a printer, a digital scanner, a headset, a headphone, a speaker, a microphone, a camera, or a gamepad, etc. The target peripheral device 10 may be connected to the host device 100 by a USB connector 20.

[0041] FIG. 2 shows a schematic illustration of the host device 100 configured to communicate data with a target peripheral device 10. The host device 100 may include a host interface 110 and a processor 120 coupled to the host interface 110.

[0042] The host interface 110 may be configured to receive a USB connector 20 for connecting with the target peripheral device 10. The host interface 110 may be a receptacle configured to receive a plug of a USB connector 20, by which a target peripheral device 10 may be connected to the host device 100. The host interface 110 may be a USB type-C interface. The target peripheral device 10 may be wirelessly connected to the host device 100. In some embodiments, the host interface 110 of the host device 100 may comprise a USB port and a dongle (e.g., a Bluetooth Low Energy (BLE) dongle) connected to the USB port. The target peripheral device 10 may be wirelessly connected to the host device 100 via the dongle. The method for communicating data between a host device (e.g., the host device 100) and a target peripheral device (e.g., the target peripheral device 10) in the present disclosure may be applicable for wired and wireless connection between the host device and the target peripheral device.

[0043] FIG. 3 is a flow chart depicting a method 300 for communicating data between a host device (e.g., the host device 100) and a target peripheral device (e.g., the target peripheral

device 10), according to some embodiments of the present disclosure. As shown in FIG. 3, the processor 120 may be configured to receive at least one identifier of the target peripheral device 10 connected to the host device 100 (Step 302). The at least one identifier may include a first identifier (e.g., vendor identification (VID)) and/or a second identifier (e.g., a product identification (PID)).

[0044] The processor 120 may be configured to generate a configuration key of the target peripheral device 10 based on the at least one identifier of the target peripheral device 10 (Step 304). The processor 120 may be configured to determine if the configuration key of the target peripheral device 10 is in a look-up table (Step 306). The look-up table may be stored in a memory of the host device 100. The look-up table may include configuration keys for a plurality of peripheral devices and configuration data corresponding to each configuration key.

[0045] The processor 120 may be configured to obtain configuration data of the target peripheral device 10 according to the configuration key of the target peripheral device 10 and the look-up table, when the configuration key of the target peripheral device 10 is in the look-up table (Step 308).

[0046] The processor 120 may be configured to obtain an interface name of a USB interface of the target peripheral device 10 (Step 310) and obtain configuration data for the target peripheral device 10 based on the interface name of the USB interface of the target peripheral device 10 (Step 312), when the configuration key of the target peripheral device 10 is absent in the look-up table, that is, the configuration key of the target peripheral device 10 is not included in the look-up table.

[0047] The processor 120 may be configured to communicate data between the host device 100 and the target peripheral device 10 according to the configuration data for the target peripheral device 10 (Step 350).

[0048] The processor 120 may be configured to obtain a raw descriptor of the USB interface of the target peripheral device 10, parse the raw descriptor to obtain a string index for the USB interface and obtain the interface name of the USB interface based on the string index. In some embodiments, the interface name of the USB interface may be obtained based on the string index using Human Interface Devices (HID) instructions.

[0049] According to some non-limiting embodiments, after the host device 100 accesses the target peripheral device 10, the host device 100 may obtain a USB device connection data of the target peripheral device 10, and obtain a raw descriptor for the USB interface of the target peripheral device 10 based on the USB device connection data. The host device 100 may parse the raw descriptor to obtain an interface descriptor list and obtain a string index (iInterface) for

the USB interface of the target peripheral device 10 based on the interface descriptor list. The host device 100 may obtain string descriptor length and string descriptor code based on the string index through control transfer instructions. The host device 100 may obtain string descriptor, which represents the interface name of the USB interface of the target peripheral device 10, based on the string descriptor length and string descriptor code.

[0050] The interface name of a USB interface of the target peripheral device 10 may be represented by a string descriptor, e.g., Gamepad (HID, XINPUT), Touch and Media, Notify (*), Protocol2.5 (Interhaptics), Audio Class Control, Audio Speaker, and Audio Recorder, etc.

[0051] The configuration data for the target peripheral device 10 may include USB mode type, endpoint type, endpoint transmission size and direction of data for endpoint, etc. The configuration data may include notification configuration data, haptic configuration data, etc. In the notification configuration data, the USB mode type may be bulk transfer, the endpoint type may be interrupt endpoint type, the endpoint transmission size may be a maximum packet size, and the direction of data for endpoint may be IN. In the haptic configuration data, the USB mode type may be control transfer, the endpoint type may be interrupt endpoint type, the endpoint transmission size may be default packet size and the direction of data for endpoint may be IN/OUT.

[0052] FIG. 4 is a flow chart depicting an exemplary method 400 for communicating data between the host device 100 and the target peripheral device 10. As shown in the exemplary process of FIG. 4, the host device 100 monitors whether a target peripheral device 10 is connected to the host device 100 (Step 402). If the host device 100 determines that there is no target peripheral device 10 is connected to the host device 100, the host device 100 does not take any action (Step 404). If the host device 100 detects a target peripheral device 10 is connected to the host device 100, the host device 100 may determine if a vendor identification (VID) of the target peripheral device 10 is a predefined VID (Step 406). If the host device 100 determines that the VID of the target peripheral device 10 is not the predefined VID, the host device 100 may indicate that the target peripheral device 10 is not supported by the host device 100 (Step 408). If the host device 100 determines that the VID of the target peripheral device 10 is the pre-defined VID, the host device 100 may generate a configuration key for the target peripheral device 10 based on the VID and a product identification (PID) of the target peripheral device 10 (Step 410). The host device 100 may further determine if the configuration key of the target peripheral device 10 is in a look-up table of the host device 100 (Step 412). The look-up table may be stored in a memory of the host device 100. The look-up table may include configuration keys for a plurality of peripheral devices and configuration data

corresponding to each configuration key. When the configuration key of the target peripheral device 10 is in the look-up table of the host device 100, the processor 120 of the host device 100 may obtain configuration data of the target peripheral device 10 according to the configuration key of the target peripheral device 10 and the look-up table (Step 414). The configuration data of the target peripheral device 10 may be configuration data that corresponds to the configuration key of the target peripheral device 10 in the look-up table. The processor 120 may be configured to communicate data between the host device 100 and the target peripheral device 10 according to the configuration data for the target peripheral device 10 (Step 450).

[0053] When the configuration key of the target peripheral device 10 is absent in the look-up table of the host device 100, the host device 100 may check if the target peripheral device 10 is equipped with a predefined/acceptable operating system, e.g., Android 5.0 or higher, Windows, MacOS, iOS, Linux (Step 418). If the target peripheral device 10 is equipped with the predefined/acceptable operating system, the host device 100 may obtain an interface name of a USB interface of the target peripheral device 10 (Step 420). If the target peripheral device 10 is not equipped with the predefined/acceptable operating system, the host device 100 may determine if the host device 100 has permission to access the target peripheral device 10 (Step 422). If the host device 100 has permission to access the target peripheral device 10, the host device 100 may access the target peripheral device 10 (Step 424). If the host device 100 does not have permission to access the target peripheral device 10, the host device 100 may apply for permission to access the target peripheral device 10 (Step 426), and access the target peripheral device 10 (Step 424) after receiving the permission. After accessing the target peripheral device 10, the host device 100 may obtain a raw descriptor for the USB interface of the target peripheral device 10 (Step 428), parse the raw descriptor to obtain a string index for the USB interface, e.g., using Human Interface Devices (HID) instructions (Step 430), and obtain the interface name of the USB interface of the target peripheral device 10 based on the string index (Step 432). The host device 100 may obtain configuration data for the target peripheral device 10 based on the interface name of the USB interface of the target peripheral device 10 (Step 434). Data may be communicated between the host device 100 and the target peripheral device 10 according to the configuration data for the target peripheral device 10 (Step 452).

[0054] FIG. 5 shows an exemplary process 550 of communicating data between the host device 100 and the target peripheral device 10 according to the configuration data for the target peripheral device 10.

[0055] The processor 120 of the host device 100 may determine if the interface name of the USB interface of the target peripheral device 10 supports notification data (Step 502), for example, the processor 120 of the host device 100 may determine if the interface name of the USB interface of the target peripheral device 10 contains “Notify”, or “NOTIFICATION”, etc. When the processor 120 of the host device 100 determines that the interface name of the USB interface of the target peripheral device 10 supports notification data, , configuration for monitoring notification data (i.e., notification configuration data) may be set for communicating data between the host device 100 and the target peripheral device 10 according to notification configuration data (Step 504). For example, for communicating data between the host device 100 and the target peripheral device 10 according to the notification configuration data, the USB mode type may be set as bulk transfer, the endpoint type may be set as interrupt endpoint type, the endpoint transmission size may be set as a maximum packet size and the direction of data for endpoint may be set as IN. The host device 100 may initialize communication thread for monitoring notification data (Step 506), and may monitor notification data in the data being communicated between the host device 100 and the target peripheral device 10 (Step 508). The host device 100 may determine if the target peripheral device 10 is closed (Step 510). When the target peripheral device 10 is not closed, the communication of data between the host device 100 and the target peripheral device 10 may continue (Step 512). When the target peripheral device 10 is closed, the communication of data between the host device 100 and the target peripheral device 10 may stop (Step 514).

[0056] When the processor 120 of the host device 100 determines that the interface name of the USB interface of the target peripheral device 10 does not support notification data, the processor 120 of the host device 100 may determine if the interface name of the USB interface of the target peripheral device 10 supports haptic data (Step 516), for example, the processor 120 of the host device 100 may determine if the interface name of the USB interface of the target peripheral device 10 contains “HAPTIC”, “INTERHAPTICS” or “PROTOCOL25”, etc. When the processor 120 of the host device 100 determines that the interface name of the USB interface of the target peripheral device 10 supports haptic data, configuration for communicating haptic data (i.e., haptic configuration data) may be set for communicating data between the host device 100 and the target peripheral device 10 according to haptic configuration data (Step 518). For example, for communicating data between the host device 100 and the target peripheral device 10 according to the haptic configuration data, the USB mode type may be set as control transfer, the endpoint type may be set as interrupt endpoint type, the endpoint transmission size may be set as a default packet size and the direction of data

for endpoint may be set as IN/OUT. Haptic data may be communicated between the host device 100 and the target peripheral device 10 (Step 520). The host device 100 may determine if the target peripheral device 10 is closed (Step 510). When the target peripheral device 10 is not closed, the communication of data between the host device 100 and the target peripheral device 10 may continue (Step 512). When the target peripheral device 10 is closed, the communication of data between the host device 100 and the target peripheral device 10 may stop (Step 514).

[0057] When the USB interface of the target peripheral device 10 does not support notification data or haptic data, no additional action may be taken, and data may be communicated between the host device 100 and the target peripheral device 10 (Step 522). The host device 100 may determine if the target peripheral device 10 is closed (Step 510). When the target peripheral device 10 is not closed, the communication of data between the host device 100 and the target peripheral device 10 may continue (Step 512). When the target peripheral device 10 is closed, the communication of data between the host device 100 and the target peripheral device 10 may stop (Step 514).

[0058] According to some non-limiting embodiments, the processor 120 may be configured to enumerate a plurality of USB interfaces of the target peripheral device 10, obtain an interface name of each of the plurality of USB interfaces of the target peripheral device 10, and obtain configuration data for the target peripheral device 10 based on the interface names of the plurality of USB interfaces of the target peripheral device 10, when the configuration key of the target peripheral device 10 is absent in the look-up table. The processor 120 may be configured to communicate data between the host device 100 and the target peripheral device 10 according to the configuration data for the target peripheral device 10. The processor 120 may be configured to communicate data between the host device 100 and the target peripheral device 10 through each of the plurality of USB interfaces of the target peripheral device 10 according to the configuration data for the target peripheral device 10. The configuration data may include respective configuration data for each of the plurality of USB interfaces of the target peripheral device 10. For example, the plurality of USB interfaces may include a first USB interface and a second USB interface, the configuration data may include notification configuration data for the first USB interface, and haptic configuration data for the second USB interface. When the interface name of the first USB interface of the target peripheral device 10 supports notification data, for example, the interface name of the first USB interface contains “Notify” or “NOTIFICATION”, etc, the data may be communicated between the host device 100 and the target peripheral device 10 through the first USB interface according to the notification configuration data, e.g., the USB mode type may be set as bulk transfer, the

endpoint type may be set as interrupt endpoint type, the endpoint transmission size may be set as a maximum packet size and the direction of data for endpoint may be set as IN. When the interface name of the second USB interface of the target peripheral device 10 supports haptic data, for example, the interface name of the second USB interface contains “HAPTIC”, “INTERHAPTICS” or “PROTOCOL25”, etc, the data may be communicated between the host device 100 and the target peripheral device 10 through the second USB interface according to the haptic configuration data, e.g., the USB mode type may be set as control transfer, the endpoint type may be set as interrupt endpoint type, the endpoint transmission size may be set as a default packet size and the direction of data for endpoint may be set as IN/OUT.

[0059] FIG. 6 is a flow chart depicting a method 600 for communicating data between a host device and a target peripheral device connected to the host device, according to some embodiments of the present disclosure. The method 600 may include:

- a step 610 of receiving at least one identifier of the target peripheral device;
- a step 620 of generating a configuration key of the target peripheral device based on the at least one identifier of the target peripheral device;
- a step 630 of determining if the configuration key of the target peripheral device is in a look-up table, the look-up table including configuration keys for a plurality of peripheral devices and configuration data corresponding to each configuration key;
- a step 640 of obtaining an interface name of a USB interface of the target peripheral device and obtaining configuration data for the target peripheral device based on the interface name of the USB interface of the target peripheral device, when the configuration key of the target peripheral device is absent in the look-up table; and
- a step 650 of communicating data between the host device and the target peripheral device according to the configuration data for the target peripheral device.

[0060] The at least one identifier may include a first identifier (e.g., vendor identification (VID)) and/or a second identifier (e.g., a product identification (PID)).

[0061] FIG. 7 is a flow chart depicting the step 640 of obtaining the interface name of the USB interface of the target peripheral device. The step 640 of obtaining the interface name of the USB interface of the target peripheral device may include:

- a step 642 of obtaining a raw descriptor for the USB interface of the target peripheral device;
- a step 644 of parsing the raw descriptor to obtain a string index for the USB interface; and

- a step 646 of obtaining the interface name of the USB interface based on the string index.

[0062] The interface name of the USB interface may be obtained based on the string index using Human Interface Device (HID) instructions.

[0063] The step 642 of obtaining the raw descriptor for the USB interface of the target peripheral device may include applying for USB permission to access the target peripheral device to obtain the raw descriptor for the USB interface of the target peripheral device.

[0064] The method 600 may further include enumerating a plurality of USB interfaces of the target peripheral device, obtaining an interface name of each of the plurality of USB interfaces of the target peripheral device, and obtaining configuration data for the target peripheral device based on the interface names of the plurality of USB interfaces of the target peripheral device, when the configuration key of the target peripheral device is absent in the look-up table.

[0065] The method 600 may further include obtaining configuration data of the target peripheral device according to the configuration key of the target peripheral device and the look-up table, when the configuration key of the target peripheral device is in the look-up table.

[0066] The configuration data for the target peripheral device may include USB mode type, endpoint type, endpoint transmission size and direction of data for endpoint.

[0067] The method 600 may further include determining that the interface name of the USB interface of the target peripheral device supports notification data; and setting the USB mode type as bulk transfer, setting the endpoint type as interrupt endpoint type, setting the endpoint transmission size as a maximum packet size and setting the direction of data for endpoint as IN. The method 600 may further include monitoring notification data in the data being communicated between the host device and the target peripheral device.

[0068] The method 600 may further include determining that the interface name of the USB interface of the target peripheral device supports haptic data; and setting the USB mode type as control transfer, setting the endpoint type as interrupt endpoint type, setting the endpoint transmission size as a default packet size and setting the direction of data for endpoint as IN/OUT. The method 600 may further include communicating haptic data between the host device and the peripheral device.

[0069] According to some non-limiting embodiments of the present disclosure, there is provided a method 700 for communicating data between a host device and a plurality of target peripheral devices, wherein each of the plurality of target peripheral devices is connected to the host device. The method 700 may include communicating data between the host device and

each of the plurality of target peripheral devices according to the method 300, 400, 600 as described above.

[0070] According to some non-limiting embodiments of the present disclosure, there is provided a host device 200 configured to communicate data with a plurality of target peripheral devices 10, each of the plurality of target peripheral devices 10 is connected to the host device 200. The host device 200 may include a plurality of host interfaces 210 and a processor 220 coupled to the plurality of host interfaces 210. Each of the plurality of host interfaces 210 may be configured to connect with a respective target peripheral device 10 of the plurality of target peripheral devices 10. The host device 200 may be configured to communicate data with each of the plurality of target peripheral devices 10 according to the method 300, 400, 600 as described above.

[0071] FIG. 8A shows a schematic illustration of a host device 200 configured to communicate data with a plurality of target peripheral devices 10, where the plurality of target peripheral devices 10 are not connected to the host device 200, in accordance with some embodiments of the present disclosure. FIG. 8B shows the host device 200 of FIG. 8A configured to communicate data with a plurality of target peripheral devices 10, where the plurality of target peripheral devices 10 are connected to the host device 200, in accordance with some embodiments of the present disclosure. The host device 200 may include a laptop computer, a smart phone, or a mobile phone etc. Each of the plurality of target peripheral devices 10 may include a keyboard, a mouse, a printer, a digital scanner, a headset, a headphone, a speaker, a microphone, a camera, or a gamepad, etc. Each of the plurality of target peripheral devices 10 may be connected to the host device 200 by a respective USB connector 20.

[0072] FIG. 9 shows a schematic illustration of the host device 200 configured to communicate data with a plurality of target peripheral devices 10. The host device 200 may be configured to communicate data with each of the plurality of target peripheral devices 10 according to method 300, 400, 600 as described above. The host device 200 may include a plurality of host interfaces 210 and a processor 220 coupled to the plurality of host interfaces 210.

[0073] Each of the plurality of host interfaces 210 may be configured to receive a respective USB connector 20 for connecting with the target peripheral device 10. Each of the plurality of host interfaces 210 may be a receptacle configured to receive a plug of a USB connector 20, by which a target peripheral device 10 may be connected to the host device 200. Each of the plurality of host interfaces 210 may be a USB type-C interface. Each of the plurality of target peripheral devices 10 may be wirelessly connected to the host device 200. In some embodiments, the host interface 210 of the host device 200 may comprise a USB port and a

dongle (e.g., a Bluetooth Low Energy (BLE) dongle) connected to the USB port. Each of the plurality of target peripheral devices 10 may be wirelessly connected to the host device 200 via the dongle. The method for communicating data between a host device (e.g., the host device 200) and a plurality of target peripheral devices (e.g., the plurality of target peripheral devices 10) in the present disclosure may be applicable for wired and wireless connection between the host device and the target peripheral device.

[0074] When a peripheral device is connected to a conventional host device, the conventional host device receives vendor identification and product identification of the peripheral device, generates a configuration key of the peripheral device based on the vendor identification and product identification, obtains configuration data of the peripheral device according to the configuration key and a look-up table stored in the conventional host device, and communicates data with the peripheral device according to the configuration data. When there is a need to connect a new peripheral device to a conventional host device where configuration key generated based on the vendor identification and product identification of the new peripheral device is not in the look-up table of the conventional host device, it is required to update the look-up table to include configuration data that corresponds to configuration key generated based on the vendor identification and product identification of the new peripheral device in the look-up table.

[0075] When a peripheral device is connected to the host device in the present disclosure, the host device receives at least one identifier (e.g., the vendor identification and product identification) of the peripheral device, generates a configuration key of the peripheral device based on the at least one identifier (e.g., vendor identification and product identification), obtains configuration data of the peripheral device according to the configuration key and a look-up table stored in the conventional host device, and communicates data with the peripheral device according to the configuration data, when the configuration key generated based on the at least one identifier (e.g., vendor identification and product identification) of the peripheral device is in the look-up table of the host device. When the configuration key generated based on the at least one identifier (e.g., vendor identification and product identification) of the peripheral device is absent in the look-up table of the host device, the host device in the present disclosure obtains an interface name of a USB interface of the peripheral device, obtains configuration data for the peripheral device based on the interface name of the USB interface of the peripheral device, and communicate data with the peripheral device according to the configuration data for the peripheral device.

[0076] Therefore, when there is a need to connect a new peripheral device to the host device of the present disclosure, where configuration key generated based on the at least one identifier (e.g., vendor identification and product identification) of the new peripheral device is not in the look-up table of the host device, it is not required to update the look-up table to include configuration data that corresponds to configuration key generated based on the at least one identifier (e.g., vendor identification and product identification) of the new peripheral device in the look-up table. In the meantime, when a peripheral device is connected to the host device of the present disclosure, where configuration key generated based on the at least one identifier (e.g., vendor identification and product identification) of the peripheral device is in the look-up table of the host device, the host device in the present disclosure follows the same steps in the conventional host device, i.e., obtains configuration data of the peripheral device according to the configuration key and a look-up table stored in the host device, and communicates data with the peripheral device according to the configuration data. Therefore, the present disclosure provides a time efficient and cost-effective method of communicating data between a host device and a peripheral device.

[0077] While the disclosure 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 disclosure as defined by the appended claims. The scope of the disclosure 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

What is claimed is:

1. A method for communicating data between a host device and a target peripheral device connected to the host device, the method comprising:
 - receiving at least one identifier of the target peripheral device;
 - generating a configuration key of the target peripheral device based on the at least one identifier of the target peripheral device;
 - determining if the configuration key of the target peripheral device is in a look-up table, the look-up table comprising configuration keys for a plurality of peripheral devices and configuration data corresponding to each configuration key;
 - obtaining an interface name of a USB interface of the target peripheral device and obtaining configuration data for the target peripheral device based on the interface name of the USB interface of the target peripheral device, when the configuration key of the target peripheral device is absent in the look-up table; and
 - communicating data between the host device and the target peripheral device according to the configuration data for the target peripheral device.
2. The method of claim 1, wherein obtaining the interface name of the USB interface of the target peripheral device comprising:
 - obtaining a raw descriptor for the USB interface of the target peripheral device;
 - parsing the raw descriptor to obtain a string index for the USB interface; and
 - obtaining the interface name of the USB interface based on the string index.
3. The method of claim 2, wherein the interface name of the USB interface is obtained based on the string index using Human Interface Device (HID) instructions.
4. The method of claim 2, wherein obtaining the raw descriptor for the USB interface of the target peripheral device comprises:
 - applying for USB permission to access the target peripheral device to obtain the raw descriptor for the USB interface of the target peripheral device.

5. The method of claim 1, further comprising
 - enumerating a plurality of USB interfaces of the target peripheral device,
 - obtaining an interface name of each of the plurality of USB interfaces of the target peripheral device, and obtaining configuration data for the target peripheral device based on the interface names of the plurality of USB interfaces of the target peripheral device, when the configuration key of the target peripheral device is absent in the look-up table.
6. The method of claim 1, wherein the configuration data for the target peripheral device comprises USB mode type, endpoint type, endpoint transmission size and direction of data for endpoint.
7. The method of claim 6, further comprising:
 - determining that the interface name of the USB interface of the target peripheral device supports haptic data.
8. The method of claim 1, further comprises:
 - obtaining configuration data of the target peripheral device according to the configuration key of the target peripheral device and the look-up table, when the configuration key of the target peripheral device is in the look-up table.
9. The method of claim 7, further comprising:
 - communicating haptic data between the host device and the peripheral device.
10. The method of claim 1, wherein the at least one identifier comprises vendor identification and product identification.
11. A host device configured for communicating data with a target peripheral device connected to the host device, the host device comprising:
 - a host interface configured to connect with the target peripheral device; and
 - a processor coupled to the host interface, wherein the processor is configured to:
 - receive at least one identifier of the target peripheral device;

generate a configuration key of the target peripheral device based on the at least one identifier of the target peripheral device;

determine if the configuration key of the target peripheral device is in a look-up table, the look-up table comprising configuration keys for a plurality of peripheral devices and configuration data corresponding to each configuration key;

obtain an interface name of a USB interface of the target peripheral device and obtain configuration data for the target peripheral device based on the interface name of the USB interface of the target peripheral device, when the configuration key of the target peripheral device is absent in the look-up table, and

communicate data between the host device and the target peripheral device according to the configuration data for the target peripheral device.

12. The host device of claim 11, wherein the processor is further configured to:
 - obtain a raw descriptor for the USB interface of the target peripheral device;
 - parse the raw descriptor to obtain a string index for the USB interface;
 - obtain the interface name of the USB interface based on the string index.
13. The host device of claim 12, wherein the interface name of the USB interface is obtained based on the string index using Human Interface Device (HID) instructions.
14. The host device of claim 12, wherein USB permission to access the target peripheral device is applied for to obtain the raw descriptor for the USB interface of the target peripheral device.
15. The host device of claim 11, wherein the processor is further configured to enumerate a plurality of USB interfaces of the target peripheral device, obtain an interface name of each of the plurality of USB interfaces of the target peripheral device, and obtain configuration data for the target peripheral device based on the interface names of the plurality of USB interfaces of the target peripheral device, when the configuration key of the target peripheral device is absent in the look-up table.

16. The host device of claim 11, wherein the configuration data for the target peripheral device comprises USB mode type, endpoint type, endpoint transmission size and direction of data for endpoint.
17. The host device of claim 16, wherein the processor is further configured to:
determine that the interface name of the USB interface of the target peripheral device supports haptic data.
18. The host device of claim 11, wherein the processor is further configured to:
obtain configuration data of the target peripheral device according to the configuration key of the target peripheral device and the look-up table, when the configuration key of the target peripheral device is in the look-up table.
19. The host device of claim 17, wherein haptic data is communicated between the host device and the peripheral device.
20. The host device of claim 11, wherein the at least one identifier comprises vendor identification and product identification.