



(12) 发明专利

(10) 授权公告号 CN 103309787 B

(45) 授权公告日 2014. 12. 10

(21) 申请号 201310264252. 0

审查员 赵小娟

(22) 申请日 2013. 06. 28

(73) 专利权人 飞天诚信科技股份有限公司

地址 100085 北京市海淀区学清路 9 号汇智大厦 B 楼 17 层

(72) 发明人 陆舟 于华章

(51) Int. Cl.

G06F 11/26 (2006. 01)

(56) 对比文件

CN 102117238 A, 2011. 07. 06, 全文.

CN 101727374 A, 2010. 06. 09, 全文.

CN 1633198 A, 2005. 06. 29, 全文.

KR 10-0684548 B1, 2007. 02. 20, 全文.

US 2002/0104046 A1, 2002. 08. 01, 全文.

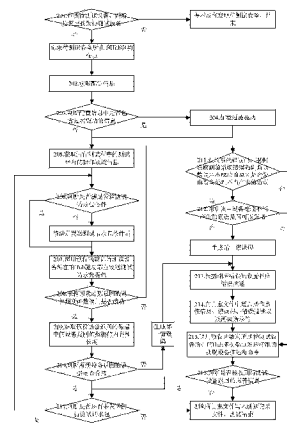
权利要求书3页 说明书12页 附图3页

(54) 发明名称

一种非标准 USB 协议兼容性检测方法

(57) 摘要

本发明公开了一种非标准 USB 协议兼容性检测方法,属于计算机领域。所述方法包括查找待测试设备,读取配置信息,当满足发送测试请求包条件时,调用预设函数向待测试设备所在的 Hub 驱动端点发送测试请求包并判断函数调用是否成功,若函数调用不成功则检查不成功的原因是否为由设备处理不当产生的错误,若函数调用成功则检查待测试设备返回的数据是否合法,根据检查结果生成相应错误码并记录相应信息,检查待测试设备是否失去响应,是则测试结束,否则继续测试。本发明的有益效果在于能够通过检测,提前发现设备相关的问题和缺陷,进而预防因软件或驱动发送非标准 USB 协议请求导致 USB 设备异常的问题,能够提高 USB 设备的健壮性。



1. 一种非标准 USB 协议兼容性检测方法,其特征在于,包括:

步骤 A:查找待测试设备,判断是否查找到待测试设备,是则记录待测试设备所在的 Hub 驱动端点,执行步骤 B,否则提示没有发现待测试设备,结束;

步骤 B:读取配置信息;

步骤 C:根据所述配置信息判断是否满足发送测试请求包条件,是则执行步骤 D,否则待满足发送测试请求包条件后执行步骤 D;

步骤 D:调用预设函数,根据所述配置信息中的当前测试信息向所述待测试设备所在的 Hub 驱动端点发送测试请求包,判断函数调用是否成功,是则执行步骤 E,否则执行步骤 F;

步骤 E:判断所述待测试设备返回的数据是否合法,是则更新当前测试信息,返回执行步骤 C,否则生成第一错误码,根据所述第一错误码设置错误描述,执行步骤 G;

步骤 F:检查函数调用不成功的原因,若为由设备处理不当产生的错误则生成第二错误码,根据所述第二错误码设置错误描述,执行步骤 G,否则更新当前测试信息,返回执行步骤 C;

步骤 G:记录相应错误码、错误描述和所述测试请求包,执行步骤 H;

步骤 H:检查待测试设备是否失去响应,是则测试结束,否则更新当前测试信息,返回执行步骤 C。

2. 如权利要求 1 所述的方法,其特征在于,所述步骤 C 具体为:判断配置信息中是否包含延时信息,是则延时相应时间,执行步骤 D,否则执行步骤 D,其中,所述相应时间等于所述延时信息的值。

3. 如权利要求 1 所述的方法,其特征在于,所述步骤 B 和所述步骤 C 之间还包括:检查所述配置信息中是否包含过滤驱动信息,是则先加载过滤驱动后再执行步骤 C,否则执行步骤 C。

4. 如权利要求 3 所述的方法,其特征在于,所述步骤 C 具体为:判断是否接收到过滤驱动发送的测试消息,是则执行步骤 D,否则等待接收所述过滤驱动发送的测试消息,待接收到所述过滤驱动发送的测试消息后执行步骤 D。

5. 如权利要求 3 所述的方法,其特征在于,所述步骤 C 具体为:

步骤 i:判断是否接收到过滤驱动发送的测试消息,是则执行步骤 ii,否则执行步骤 iii;

步骤 ii:检查所述配置信息中是否包含延时信息,是则延时相应时间后执行步骤 D,否则执行步骤 D,其中,所述相应时间等于所述延时信息的值;

步骤 iii:等待接收所述过滤驱动发送的测试消息,待接收到所述过滤驱动发送的测试消息后返回执行步骤 ii。

6. 如权利要求 1 所述的方法,其特征在于,所述配置信息中的当前测试信息为当前测试请求包;

所述调用预设函数,根据所述配置信息中的当前测试信息向所述待测试设备所在的 Hub 驱动端点发送测试请求包,具体为:调用预设函数,向所述待测试设备所在的 Hub 驱动端点发送当前测试请求包。

7. 如权利要求 1 所述的方法,其特征在于,所述配置信息中的当前测试信息为当前待

执行测试；

所述调用预设函数，根据所述配置信息中的当前测试信息向所述待测试设备所在的 Hub 驱动端点发送测试请求包具为：根据当前待执行测试生成测试请求包，调用预设函数，向所述待测试设备所在的 Hub 驱动端点发送生成的测试请求包。

8. 如权利要求 7 所述的方法，其特征在于，所述根据当前待执行测试生成测试请求包具体包括：初始化测试请求包，读取所述配置信息中的当前待执行测试，根据当前待执行测试的类型设置所述测试请求包中的测试类型数据，根据请求长度设置所述测试请求包中的测试长度数据；

所述步骤 E 中，所述判断所述待测试设备返回的数据是否合法具体为：从所述预设函数返回的数据中获取所述待测试设备返回数据的内容和长度，根据所述测试类型数据判断所述待测试设备返回数据的内容是否合法，根据所述测试长度数据判断所述待测试设备返回数据的长度是否合法，若所述待测试设备返回数据的内容和长度均合法则所述待测试设备返回的数据合法，否则所述待测试设备返回的数据不合法。

9. 如权利要求 8 所述的方法，其特征在于，所述测试请求包由第一数据、第二数据、所述测试类型数据、第三数据以及请求长度数据组成；

所述初始化测试请求包具体为：初始化第一数据为第一预设数据、初始化第二数据为第二预设数据、初始化测试类型数据为第一初始值、初始化第三数据为第三预设数据以及初始化请求长度数据为第二初始值。

10. 根据权利要求 9 所述的方法，其特征在于，所述当前待执行测试的类型包括：获取设备描述符测试，和 / 或获取配置描述符测试，和 / 或获取字符串描述符测试，和 / 或获取接口描述符测试，和 / 或获取测试端点描述符测试，和 / 或获取人机接口设备描述符测试；

所述根据所述测试类型数据判断所述待测试设备返回数据的内容是否合法具体为，判断所述待测试设备返回数据是否为当前待执行测试的类型相对应的描述符，如果是，则确定所述待测试设备返回的数据合法；否则，确定所述待测试设备返回的数据不合法。

11. 根据权利要求 9 所述的方法，其特征在于，所述根据请求长度设置测试请求包中的测试长度数据具体包括：判断请求长度是否小于第五预设数据，是则更新所述请求长度，根据更新后的请求长度设置测试请求包中的测试长度数据，执行步骤 D，否则更新所述配置信息中的当前待执行测试并修改所述请求长度为初始值，返回执行步骤 B，其中所述请求长度的初始值为第四预设数据。

12. 如权利要求 9 所述的方法，其特征在于，所述根据所述测试长度数据判断设备返回数据的长度是否合法具体为，判断设备返回数据的长度是否大于请求长度，是则设备返回数据的长度不合法，否则设备返回数据的长度合法。

13. 如权利要求 1 所述的方法，其特征在于，所述判断函数调用是否成功具体为，根据所述预设函数返回的数据判断函数调用是否成功。

14. 如权利要求 1 所述的方法，其特征在于，所述检查函数调用不成功的原因之前还包括：获取系统错误码；

所述检查函数调用不成功的原因具体为，判断所述系统错误码是否为第一预设错误码，是则函数调用不成功的原因不为由设备处理不当产生的错误，否则函数调用不成功的原因为由设备处理不当产生的错误。

15. 如权利要求 1 所述的方法,其特征在于,所述步骤 G 之前还包括获取测试平台的操作系统信息;

所述步骤 G 中还包括:向日志文件中写入所述操作系统信息。

16. 如权利要求 1 所述的方法,其特征在于,所述步骤 A 中查找到待测试设备时还包括:记录待测试设备路径以及所述路径上各设备的信息;

所述步骤 G 中还包括:向日志文件中写入所述待测试设备路径以及所述路径上各设备的信息。

17. 如权利要求 1 所述的方法,其特征在于,所述步骤 H 具体包括:调用预设函数向所述待测试设备所在的 Hub 驱动端点发送标准的获取设备描述符命令,判断是否接收到待测试设备返回的应答信息,是则待测试设备未失去响应,返回执行步骤 C,否则待测试设备失去响应,测试结束。

18. 如权利要求 1 所述的方法,其特征在于,所述步骤 F 中,当函数调用不成功的原因因为设备处理不当产生的错误时,还包括:判断所述设备处理不当产生的错误是否可以忽略,是则返回执行步骤 H,否则生成第二错误码,根据所述第二错误码设置错误描述,执行步骤 G。

19. 如权利要求 14 所述的方法,其特征在于,所述步骤 F 中,当函数调用不成功的原因因为设备处理不当产生的错误时,还包括:判断所述系统错误码是否为第二预设错误码,是则重新打开待测试设备,返回执行步骤 D,否则生成第二错误码,根据所述第二错误码设置错误描述,执行步骤 G。

20. 如权利要求 1 所述的方法,其特征在于,

所述步骤 B 和所述步骤 C 之间还包括:从所述配置信息中选择一条未被使用过的测试信息,将选择出的测试信息作为当前测试信息;

所述步骤 E 中,当设备返回的数据合法时,还包括:判断是否已完成所述配置信息中的全部测试,是则测试结束,否则从所述配置信息中选择一条未被使用过的测试信息,将选择出的测试信息作为当前测试信息,返回执行步骤 C;

所述步骤 F 中,当函数调用不成功的原因不是由设备处理不当产生的错误时,还包括:判断是否已完成所述配置信息中的全部测试,是则测试结束,否则从所述配置信息中选择一条未被使用过的测试信息,将选择出的测试信息作为当前测试信息,返回执行步骤 C;

所述步骤 H 中,当待测试设备未失去响应时,还包括:判断是否已完成所述配置信息中的全部测试,是则测试结束,否则从所述配置信息中选择一条未被使用过的测试信息,将选择出的测试信息作为当前测试信息,返回执行步骤 C。

一种非标准 USB 协议兼容性检测方法

技术领域

[0001] 本发明涉及计算机领域,尤其涉及一种非标准 USB 协议兼容性检测方法。

背景技术

[0002] USB 设备常常会因为接收到一些软件或驱动发送的非标准 USB 协议请求而导致设备不能正常工作甚至失去响应。现有技术中会对 USB 设备进行兼容性检测,但对于上述 USB 设备因接收到非标准 USB 协议请求而导致的问题尚没有相应的检测方法。

发明内容

[0003] 本发明的目的是为了克服已有技术的缺陷,为了解决 USB 设备因为非标准 USB 协议请求导致设备不能正常工作甚至失去响应的问题,提出一种非标准 USB 协议兼容性检测工具的工作方法。

[0004] 本发明方法是通过下述技术方案实现的:

[0005] 一种非标准 USB 协议兼容性检测方法,其基本实施过程如下:

[0006] 步骤 A:查找待测试设备,判断是否查找到待测试设备,是则记录待测试设备所在的 Hub 驱动端点,执行步骤 B,否则提示没有发现待测试设备,结束;

[0007] 步骤 B:读取配置信息;

[0008] 步骤 C:根据所述配置信息判断是否满足发送测试请求包条件,是则执行步骤 D,否则待满足发送测试请求包条件后执行步骤 D;

[0009] 步骤 D:调用预设函数,根据所述配置信息中的当前测试信息向所述待测试设备所在的 Hub 驱动端点发送测试请求包,判断函数调用是否成功,是则执行步骤 E,否则执行步骤 F;

[0010] 步骤 E:判断所述待测试设备返回的数据是否合法,是则更新当前测试信息,返回执行步骤 C,否则生成第一错误码,根据所述第一错误码设置错误描述,执行步骤 G;

[0011] 步骤 F:检查函数调用不成功的原因,若为由设备处理不当产生的错误则生成第二错误码,根据所述第二错误码设置错误描述,执行步骤 G,否则更新当前测试信息,返回执行步骤 C;

[0012] 步骤 G:记录相应错误码、错误描述和所述测试请求包;

[0013] 步骤 H:检查待测试设备是否失去响应,是则测试结束,否则更新当前测试信息,返回执行步骤 C。

[0014] 其中,所述步骤 C 具体为:判断配置信息中是否包含延时信息,是则延时相应时间,执行步骤 D,否则执行步骤 D,其中,所述相应时间等于所述延时信息的值;

[0015] 或者,所述步骤 B 和所述步骤 C 之间还包括:检查所述配置信息中是否包含过滤驱动信息,是则先加载过滤驱动后再执行步骤 C,否则执行步骤 C;所述步骤 C 具体为:判断是否接收到过滤驱动发送的测试消息,是则执行步骤 D,否则等待接收所述过滤驱动发送的测试消息,待接收到所述过滤驱动发送的测试消息后执行步骤 D;

[0016] 或者,所述步骤B和所述步骤C之间还包括:检查所述配置信息中是否包含过滤驱动信息,是则先加载过滤驱动后再执行步骤C,否则执行步骤C;所述步骤C具体为:

[0017] 步骤i:判断是否接收到过滤驱动发送的测试消息,是则执行步骤ii,否则执行步骤iii;

[0018] 步骤ii:检查所述配置信息中是否包含延时信息,是则延时相应时间后执行步骤D,否则执行步骤D,其中,所述相应时间等于所述延时信息的值;

[0019] 步骤iii:等待接收所述过滤驱动发送的测试消息,待接收到所述过滤驱动发送的测试消息后返回执行步骤ii。

[0020] 所述配置信息中的当前测试信息为当前测试请求包;所述调用预设函数,根据所述配置信息中的当前测试信息向所述待测试设备所在的Hub驱动端点发送测试请求包,具体为:调用预设函数,向所述待测试设备所在的Hub驱动端点发送当前测试请求包;

[0021] 或者,所述配置信息中的当前测试信息为当前待执行测试;所述调用预设函数,根据所述配置信息中的当前测试信息向所述待测试设备所在的Hub驱动端点发送测试请求包具体为:根据当前待执行测试生成测试请求包,调用预设函数,向所述待测试设备所在的Hub驱动端点发送生成的测试请求包;

[0022] 其中,所述根据当前待执行测试生成测试请求包具体包括:初始化测试请求包,读取所述配置信息中的当前待执行测试,根据当前待执行测试的类型设置所述测试请求包中的测试类型数据,根据请求长度设置所述测试请求包中的测试长度数据;

[0023] 所述测试请求包由第一数据、第二数据、所述测试类型数据、第三数据以及请求长度数据组成;所述初始化测试请求包具体为:初始化第一数据为第一预设数据、初始化第二数据为第二预设数据、初始化测试类型数据为第一初始值、初始化第三数据为第三预设数据以及初始化请求长度数据为第二初始值;

[0024] 所述当前待执行测试的类型包括:获取设备描述符测试,和/或获取配置描述符测试,和/或获取字符串描述符测试,和/或获取接口描述符测试,和/或获取测试端点描述符测试,和/或获取人机接口设备描述符测试;

[0025] 所述根据请求长度设置测试请求包中的测试长度数据具体包括:判断请求长度是否小于第五预设数据,是则更新所述请求长度,根据更新后的请求长度设置测试请求包中的测试长度数据,执行步骤D,否则更新所述配置信息中的当前待执行测试并修改所述请求长度为初始值,返回执行步骤B,其中所述请求长度的初始值为第四预设数据。

[0026] 所述步骤E中,所述判断设备返回的数据是否合法具体为:从所述预设函数返回的数据中获取设备返回数据的内容和长度,根据所述测试类型数据判断设备返回数据的内容是否合法,根据所述测试长度数据判断设备返回数据的长度是否合法,若设备返回数据的内容和长度均合法则设备返回的数据合法,否则设备返回的数据不合法;其中,所述根据所述测试类型数据判断所述待测试设备返回数据的内容是否合法具体为,判断所述待测试设备返回数据是否为当前待执行测试的类型相对应的描述符,如果是,则确定所述待测试设备返回的数据合法;否则,确定所述待测试设备返回的数据不合法;所述根据所述测试长度数据判断设备返回数据的长度是否合法具体为,判断设备返回数据的长度是否大于请求长度,是则设备返回数据的长度不合法,否则设备返回数据的长度合法。

[0027] 所述判断函数调用是否成功具体为,根据所述预设函数返回的数据判断函数调用

是否成功；

[0028] 进一步地,所述检查函数调用不成功的原因之前还包括:获取系统错误码;所述检查函数调用不成功的原因具体为,判断所述系统错误码是否为第一预设错误码,是则函数调用不成功的原因不为由设备处理不当产生的错误,否则函数调用不成功的原因为由设备处理不当产生的错误。

[0029] 所述步骤 G 之前还包括获取测试平台的操作系统信息;所述步骤 G 中还包括:向日志文件中写入所述操作系统信息。

[0030] 所述步骤 A 中查找到待测试设备时还包括:记录待测试设备路径以及所述路径上各设备的信息;所述步骤 G 中还包括:向日志文件中写入所述待测试设备路径以及所述路径上各设备的信息。

[0031] 所述步骤 H 具体包括:调用预设函数向所述待测试设备所在的 Hub 驱动端点发送标准的获取设备描述符命令,判断是否接收到待测试设备返回的应答信息,是则待测试设备未失去响应,返回执行步骤 C,否则待测试设备失去响应,测试结束。

[0032] 所述步骤 F 中,当函数调用不成功的原因由设备处理不当产生的错误时,还包括:判断所述设备处理不当产生的错误是否可以忽略,是则返回执行步骤 H,否则生成第二错误码,根据所述第二错误码设置错误描述,执行步骤 G。

[0033] 所述步骤 F 中,当函数调用不成功的原因由设备处理不当产生的错误时,还包括:判断所述系统错误码是否为第二预设错误码,是则重新打开待测试设备,返回执行步骤 D,否则生成第二错误码,根据所述第二错误码设置错误描述,执行步骤 G。

[0034] 所述步骤 B 和所述步骤 C 之间还包括:从所述配置信息中选择一条未被使用过的测试信息,将选择出的测试信息作为当前测试信息;

[0035] 所述步骤 E 中,当设备返回的数据合法时,还包括:判断是否已完成所述配置信息中的全部测试,是则测试结束,否则从所述配置信息中选择一条未被使用过的测试信息,将选择出的测试信息作为当前测试信息,返回执行步骤 C;

[0036] 所述步骤 F 中,当函数调用不成功的原因不是由设备处理不当产生的错误时,还包括:判断是否已完成所述配置信息中的全部测试,是则测试结束,否则从所述配置信息中选择一条未被使用过的测试信息,将选择出的测试信息作为当前测试信息,返回执行步骤 C;

[0037] 所述步骤 H 中,当待测试设备未失去响应时,还包括:判断是否已完成所述配置信息中的全部测试,是则测试结束,否则从所述配置信息中选择一条未被使用过的测试信息,将选择出的测试信息作为当前测试信息,返回执行步骤 C。

[0038] 本发明方法的有益效果在于,能够通过检测,提前发现设备相关的问题和缺陷,进而预防因软件或驱动发送非标准 USB 协议请求导致 USB 设备异常的问题,能够提高 USB 设备的健壮性,并且上述检测可以在不影响上层应用程序使用的情况下进行。

附图说明

[0039] 为了更清楚的说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单的介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以

根据这些附图获得其他的附图。

[0040] 图 1 为本发明实施例一提供的一种非标准 USB 协议兼容性检测方法的流程图；

[0041] 图 2 为本发明实施例一提供的一种非标准 USB 协议兼容性检测方法中查找待测试设备过程的流程图；

[0042] 图 3 为本发明实施例二提供的一种非标准 USB 协议兼容性检测方法的流程图。

具体实施方式

[0043] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0044] 实施例一

[0045] 本发明实施例提供一种非标准 USB 协议兼容性检测方法,参见图 1,所述方法由测试平台运行测试程序实现,具体包括如下步骤:

[0046] 步骤 101:查找待测试设备,判断是否查找到待测试设备,是则记录待测试设备所在的 Hub (集线器) 驱动端点,执行步骤 102,否则提示没有发现待测试设备,结束;

[0047] 具体地,参见图 2,所述查找待测试设备具体包括:

[0048] 步骤 a:查找 USB HCD (USB 主控制器),判断是否还有未检测的 USB HCD,是则执行步骤 b,否则查找待测试设备过程结束;

[0049] 步骤 b:查找当前 USB HCD 对应的 Root Hub (根集线器),判断是否还有未检测的 Root Hub,是则执行步骤 c,否则返回执行步骤 a;

[0050] 步骤 c:查询当前 Root Hub 的信息,执行步骤 d;

[0051] 具体地,调用 DeviceIoControl 函数查询当前 Root Hub 的信息;

[0052] 步骤 d: 查找当前 Root Hub 的端点,判断是否还有未检测的端点,是则执行步骤 e,否则返回执行步骤 b;

[0053] 步骤 e:查询当前 Root Hub 当前端点的信息,执行步骤 f;

[0054] 具体地,调用 DeviceIoControl 函数查询当前 Root Hub 当前端点的信息;

[0055] 步骤 f:判断当前 Root Hub 当前端点连接的设备是否为 Hub (集线器),是则执行步骤 g,否则执行步骤 j;

[0056] 步骤 g:查找当前 Hub 的端点,判断是否还有未检测的端点,是则执行步骤 h,否则返回执行步骤 d;

[0057] 步骤 h:查询当前 Hub 当前端点的信息,执行步骤 i;

[0058] 步骤 i:判断当前 Hub 当前端点连接的设备是否为待测试设备,是则查找待测试设备过程结束,否则返回执行步骤 g;

[0059] 步骤 j:判断当前 Root Hub 当前端点连接的设备是否为待测试设备,是则查找待测试设备过程结束,否则返回执行步骤 d;

[0060] 更进一步地,查找到待测试设备后还包括:记录待测试设备路径以及该路径上各设备的信息;

[0061] 需要说明的是,上述过程是以待测试设备最多通过一级 Hub 与测试工具相连为例

进行说明的,但本发明并不限制于此;

[0062] 步骤 102:读取配置信息;

[0063] 具体地,所述配置信息中包括待执行测试信息;

[0064] 优选地,在本实施例中,所述待执行测试包括:获取设备描述符测试,和/或获取配置描述符测试,和/或获取字符串描述符测试,和/或获取字符串描述符(厂商信息)测试,和/或获取字符串描述符(产品信息)测试,和/或获取接口描述符测试,和/或获取测试端点【0】描述符测试,和/或获取测试端点【1】描述符测试,和/或获取测试端点【2】描述符测试,和/或获取测试端点【3】描述符测试、.....,和/或获取 HID(Human Interface Device,人机接口设备)描述符测试,和/或获取其他描述符测试;

[0065] 步骤 103:判断配置信息中是否包含过滤驱动的信息,是则执行步骤 104,否则执行步骤 105;

[0066] 步骤 104:加载过滤驱动,执行步骤 105;

[0067] 具体的,在本实施例中,过滤驱动对上层驱动发送给底层驱动的 IRP(I/O 请求包)进行拦截,并对拦截到的 IRP 进行识别,若为特定 IRP 的特定子请求则发送测试消息给测试程序并将拦截到的 IRP 转发给底层驱动程序,若不为特定 IRP 的特定子请求则将拦截到的 IRP 转发给底层驱动程序;其中,所述特定 IRP 如:IRP_MJ_WRITE、IRP_MJ_READ、IRP_MJ_DEVICE_CONTROL、IRP_INTERNAL_DEVICE_CONTROL、IRP_MJ_QUERY_INFORMATION 或 IRP_MJ SCSI 等;所述特定子请求如:上电、读或写等;

[0068] 步骤 105:获取运行测试程序的测试平台的操作系统信息;

[0069] 具体地,所述操作系统信息如 Windows 2000、Windows XP、Windows Vista 和 Windows 7 等;

[0070] 步骤 106:初始化测试请求包;

[0071] 具体地,所述测试请求包由第一数据、第二数据、测试类型数据、第三数据以及请求长度数据组成,所述初始化测试请求包包括:初始化第一数据为第一预设数据、初始化第二数据为第二预设数据、初始化测试类型数据为第一初始值、初始化第三数据为第三预设数据以及初始化请求长度数据为第二初始值;

[0072] 优选地,在本实施例中,所述测试请求包的格式为:bmRequest bRequest wValue wIndex wlength,其中 bmRequest 表示第一数据,bRequest 表示第二数据,wValue 表示测试类型数据,wIndex 表示第三数据,wlength 表示请求长度数据,第一预设数据为 0x80,第二预设数据为 0x06,第一初始值为 0x0000,第三预设数据为 0x0000,第二初始值为 0x0000;

[0073] 步骤 107:读取当前待执行测试,并根据当前待执行测试的类型设置测试请求包中的测试类型数据;

[0074] 在本实施例中,所述读取当前待执行测试包括:从配置信息的待执行测试中选择一个未被使用过的待执行测试,将选择出的待执行测试作为当前待执行测试;

[0075] 具体地,若当前待执行测试的类型为获取设备描述符则设置测试请求数据包中的测试类型数据为第一预设值,优选地,在本实施例中,设置测试请求包中的 wValue 为 0x0100;

[0076] 若当前待执行测试的类型为获取配置描述符则设置测试请求数据包中的测试类型数据为第二预设值,优选地,在本实施例中,设置测试请求包中的 wValue 为 0x0200;

[0077] 若当前待执行测试的类型为获取字符串描述符则设置测试请求数据包中的测试类型数据为第三预设值,优选地,在本实施例中,设置测试请求包中的 wValue 为 0x0300;

[0078] 若当前待执行测试的类型为获取字符串描述符(厂商信息)则设置测试请求数据包中的测试类型数据为第四预设值,优选地,在本实施例中,设置测试请求包中的 wValue 为 0x0301;

[0079] 若当前待执行测试的类型为获取字符串描述符(产品信息)则设置测试请求数据包中的测试类型数据为第五预设值,优选地,在本实施例中,设置测试请求包中的 wValue 为 0x0302;

[0080] 若当前待执行测试的类型为获取接口描述符则设置测试请求数据包中的测试类型数据为第六预设值,优选地,在本实施例中,设置测试请求包中的 wValue 为 0x0400;

[0081] 若当前待执行测试的类型为获取测试端点【0】描述符则设置测试请求数据包中的测试类型数据为第七预设值,优选地,在本实施例中,设置测试请求包中的 wValue 为 0x0500;

[0082] 若当前待执行测试的类型为获取测试端点【1】描述符则设置测试请求数据包中的测试类型数据为第八预设值,优选地,在本实施例中,设置测试请求包中的 wValue 为 0x0501;

[0083] 若当前待执行测试的类型为获取测试端点【2】描述符则设置测试请求数据包中的测试类型数据为第九预设值,优选地,在本实施例中,设置测试请求包中的 wValue 为 0x0502;

[0084] 若当前待执行测试的类型为获取测试端点【3】描述符则设置测试请求数据包中的测试类型数据为第十预设值,优选地,在本实施例中,设置测试请求包中的 wValue 为 0x0503;

[0085];

[0086] 若当前待执行测试的类型为获取 HID 描述符则设置测试请求数据包中的测试类型数据为第十二预设值,优选地,在本实施例中,设置测试请求包中的 wValue 为 0x2200;

[0087] 步骤 108:判断请求长度是否小于第五预设数据,是则执行步骤 109,否则修改所述请求长度为初始值,执行步骤 121;

[0088] 其中,所述请求长度的初始值为第四预设数据;

[0089] 优选地,在本实施例中,所述第四预设数据为 0,所述第五预设数据为 65535;

[0090] 步骤 109:更新所述请求长度,根据更新后的请求长度设置测试请求数据包中的请求长度数据;

[0091] 具体地,更新所述请求长度为其当前值加上预设步长,设置测试请求数据包中的请求长度数据为所述更新后的请求长度对应的十六进制数;

[0092] 优选地,在本实施例中,更新所述请求长度为其当前值加 1;例如,若更新后的请求长度为 1,则设置测试请求数据包中的 wlength 为 0x0001,若更新后的请求长度为 65535,则设置测试请求数据包中的 wlength 为 0xFFFF;

[0093] 步骤 110:判断是否满足发送测试请求包条件,是则执行步骤 111,否则待满足发送测试请求包条件后执行步骤 111;

[0094] 具体地,在本实施例中,所述配置信息中还可包含延时信息;

[0095] 当步骤 103 中判断结果为否时,步骤 110 具体为:检查配置信息中是否包含延时信息,是则不满足发送测试请求包条件,延时相应时间后满足发送测试请求包条件,否则满足发送测试请求包条件,其中,相应时间等于延时信息的值;

[0096] 当步骤 103 中判断结果为是时,步骤 110 具体为:检查是否接收到过滤驱动发送的测试消息,是则满足发送测试请求包条件,否则不满足发送测试请求包条件,待接收到过滤驱动发送的测试消息后满足发送测试请求包的条件;

[0097] 当步骤 103 中判断结果为是时,步骤 110 还可以具体为:检查是否接收到过滤驱动发送的测试消息,若接收到则检查配置信息中是否包含延时信息,是则不满足发送测试请求包条件,延时相应时间后满足发送测试请求包条件,否则满足发送测试请求包条件;若未接收到则不满足测试请求条件,接收到过滤驱动发送的测试消息后(配置信息中不包含延时信息)或者待接收到过滤驱动发送的测试消息时再延时相应时间后(配置信息中包含延时信息)满足发送测试请求包条件,其中,相应时间等于延时信息的值;

[0098] 优选地,步骤 110 中可设置超时机制,若在预设时间内仍然没有满足发送测试请求包条件则结束;

[0099] 步骤 111:调用预设函数向待测试设备所在的 Hub 驱动端点发送测试请求包;

[0100] 具体地,所述预设函数为 Windows 操作系统的“DeviceIoControl”函数,所述指定端点为待测试设备所在的端点;

[0101] 步骤 112:根据预设函数返回的数据判断函数调用是否成功,是则执行步骤 113,否则执行步骤 115;

[0102] 具体地,预设函数返回的数据的格式为 $ret=fun(a,b)$, ret 表示函数调用结果, a 表示设备返回数据的内容, b 表示设备返回数据的长度,当函数调用不成功时, a 和 b 的内容为空;

[0103] 优选地,在本实施例中,检查预设函数返回的数据中的 ret ,若 ret 为 TRUE 则函数调用成功,若 ret 为 FALSE 函数调用不成功;

[0104] 步骤 113:获取预设函数返回的数据中的设备返回数据的内容和长度;

[0105] 步骤 114:判断所述设备返回的数据是否合法,是则返回执行步骤 108,否则生成第一错误码,执行步骤 117;

[0106] 具体地,当设备返回的数据的长度不大于所述请求长度时,设备返回的数据的长度合法;当设备返回的数据的内容为与测试请求包中的测试类型数据相对应的描述符时,设备返回的数据的内容合法,当设备返回的数据的长度和内容均合法时,设备返回的数据合法,否则设备返回的数据不合法;

[0107] 所述第一错误码为自定义错误码,例如,在本实施例中,所述第一错误码包括 -1 和 -2,当设备返回数据的内容不合法时设置第一错误码为 -1,当设备返回数据的长度不合法时设置第一错误码为 -2;

[0108] 步骤 115:获取系统错误码,根据获取到的系统错误码判断函数调用不成功的原因是否为由设备处理不当产生的错误,是则执行步骤 116,否则执行步骤 119;

[0109] 具体地,通过系统调用 GetLastError 获取系统错误码,当系统错误码为第一预设错误码时表明函数调用不成功的原因不是由设备处理不当产生的错误,否则表明函数调用不成功的原因是由设备处理不当产生的错误;

- [0110] 优选地,在本实施例中,所述第一预设错误码为 1 或 87;
- [0111] 步骤 116:判断该由设备处理不当产生的错误是否可以忽略,是则执行步骤 119,否则生成第二错误码,执行步骤 117;
- [0112] 具体地,所述配置信息中包括可以忽略的由设备处理不当产生的错误,例如,可设置系统错误码为 31 或 995 时表示由设备处理不当产生的错误可以忽略;
- [0113] 所述第二错误码为系统错误码;
- [0114] 步骤 117:根据相应错误码设置相应错误描述;
- [0115] 步骤 118:向日志文件中写入操作系统信息、错误码、错误描述以及测试请求包;
- [0116] 具体地,本步骤中的错误码包括设备错误码和系统错误码;
- [0117] 进一步地,本步骤还可以包括向日志文件中写入待测试设备路径以及该路径上各设备的信息;
- [0118] 步骤 119:调用预设函数向所述待测试设备所在的 Hub 驱动端点发送标准的获取设备描述符命令;
- [0119] 步骤 120:判断是否接收到待测试设备返回的应答信息,是则返回执行步骤 108,否则执行步骤 122;
- [0120] 步骤 121:判断是否已执行完全部待执行测试,是则执行步骤 122,否则更新当前待执行测试,返回执行步骤 107;
- [0121] 具体地,在本实施例中,所述更新当前待执行测试包括:从配置信息的待执行测试中选择一个未被使用过的待执行测试,将选择出的待执行测试作为当前待执行测试;
- [0122] 步骤 122:将日志文件写入测试记录文件,测试结束;
- [0123] 进一步地,步骤 116 中,当函数调用不成功的原因是由设备处理不当产生的错误时还可以包括:判断系统错误码是否为第二预设错误码,是则执行步骤 A,否则执行步骤 118;其中,所述第二预设错误码为 6;
- [0124] 步骤 A:重新打开待测试设备,调用预设函数重新向 Hub 驱动指定端点发送测试请求数据包;
- [0125] 步骤 B:根据预设函数返回的数据判断函数调用是否成功,是则执行步骤 C,否则执行步骤 E;
- [0126] 步骤 C:获取预设函数返回的数据中的设备返回数据的内容以及设备返回数据的长度;
- [0127] 步骤 D:判断所述设备返回数据是否合法,是则返回执行步骤 108,否则生成第一错误码,执行步骤 G;
- [0128] 具体地,当设备返回的数据的长度不大于所述请求长度时,设备返回的数据的长度合法;当设备返回的数据的内容为与测试请求包中的测试类型数据相对应的描述符时,设备返回的数据的内容合法,当设备返回的数据的长度和内容均合法时,设备返回的数据合法,否则设备返回的数据不合法;
- [0129] 步骤 E:获取系统错误码,根据获取到的系统错误码判断函数调用不成功的原因是否为由设备处理不当产生的错误,是则执行步骤 F,否则执行步骤 I;
- [0130] 步骤 F:判断该由设备处理不当产生的错误是否可以忽略,是则返回执行步骤 108,否则生成第二错误码,执行步骤 G;

- [0131] 步骤 G:根据相应系统错误码设置相应错误描述;
- [0132] 步骤 H:向日志文件中写入操作系统信息、错误码、错误描述以及测试请求包;
- [0133] 步骤 I:调用预设函数向所述待测试设备所在的 Hub 驱动端点发送标准的获取设备描述符命令;
- [0134] 步骤 J:判断是否接收到待测试设备返回的应答信息,是则返回执行步骤 108,否则执行步骤 K;
- [0135] 步骤 K:将日志文件写入测试记录文件,结束。
- [0136] 需要说明的两点是:一、在本实施例中,请求长度是从小到大变化,容易想到的是还可以从大到小变化;二、在本实施例中,当配置信息中有过滤驱动的信息时,测试程序接收到过滤驱动发送的测试消息后执行一个待执行测试,容易想到的是每次接收到过滤驱动发送的测试信息后发送全部(或若干个)待执行测试,在此不再赘述。
- [0137] 实施例二,
- [0138] 本发明实施例提供一种非标准 USB 协议兼容性检测方法,参见图 3,所述方法由检测工具运行检测程序实现,具体包括如下步骤:
- [0139] 步骤 201:查找待测试设备,判断是否查找到待测试设备,是则记录待测试设备所在的 Hub 驱动端点,执行步骤 202,否则提示没有发现待测试设备,结束;
- [0140] 具体地,在本实施例中,查找待测试设备的过程与实施例一中的相同,在此不再赘述;
- [0141] 优选地,查找到待测试设备后还包括:记录待测试设备路径以及该路径上各设备的信息;
- [0142] 步骤 202:读取配置信息;
- [0143] 具体地,所述配置信息中还包括测试请求包;
- [0144] 步骤 203:判断配置信息中是否包含过滤驱动的信息,是则执行步骤 204,否则执行步骤 205;
- [0145] 步骤 204:加载过滤驱动,执行步骤 205;
- [0146] 具体的,在本实施例中,过滤驱动对上层驱动发送给底层驱动的 IRP (I/O 请求包)进行拦截,并对拦截到的 IRP 进行识别,若为特定 IRP 的特定子请求则发送测试消息给测试程序并将拦截到的 IRP 转发给底层驱动程序,若不为特定 IRP 的特定子请求则将拦截到的 IRP 转发给底层驱动程序;其中,所述特定 IRP 如:IRP_MJ_WRITE、IRP_MJ_READ、IRP_MJ_DEVICE_CONTROL、IRP_INTERNAL_DEVICE_CONTROL、IRP_MJ_QUERY_INFORMATION 或 IRP_MJ_SCSI 等;所述特定子请求如:上电、读或写等;
- [0147] 步骤 205:获取运行测试程序的测试平台的操作系统信息;
- [0148] 具体地,所述操作系统信息包括操作系统名称,如 Windows 2000、Windows XP、Windows Vista 和 Windows 7 等;
- [0149] 步骤 206:判断是否满足发送测试请求包条件,是则执行步骤 207,否则待满足发送测试请求包条件后执行步骤 207;
- [0150] 具体地,在本实施例中,所述配置信息中还可包含延时信息;
- [0151] 当步骤 203 中判断结果为否时,步骤 206 具体为:检查配置信息中是否包含延时信息,是则不满足发送测试请求包条件,延时相应时间后满足发送测试请求包条件,否则满足

发送测试请求包条件,其中,相应时间等于延时信息的值;

[0152] 当步骤 203 中判断结果为是时,步骤 206 具体为:检查是否接收到过滤驱动发送的测试消息,是则满足发送测试请求包条件,否则不满足发送测试请求包条件,待接收到过滤驱动发送的测试消息后满足发送测试请求包的条件;

[0153] 当步骤 203 中判断结果为是时,步骤 206 还可以具体为:检查是否接收到过滤驱动发送的测试消息,若接收到则检查配置信息中是否包含延时信息,是则不满足发送测试请求包条件,延时相应时间后满足发送测试请求包条件,否则满足发送测试请求包条件;若未接收到则不满足测试请求条件,接收到过滤驱动发送的测试消息后(配置信息中不包含延时信息)或者待接收到过滤驱动发送的测试消息时再延时相应时间后(配置信息中包含延时信息)满足发送测试请求包条件,其中,相应时间等于延时信息的值;

[0154] 优选地,步骤 206 中可设置超时机制,若在预设时间内仍然没有满足发送测试请求包条件则结束;

[0155] 步骤 207:调用预设函数向待测试设备所在的 Hub 驱动端点发送测试请求数据包;

[0156] 具体地,所述预设函数为 Windows 操作系统的“DeviceIoControl”函数,所述指定端点为待测试设备所在的端点;

[0157] 步骤 208:根据预设函数返回的数据判断函数调用是否成功,是则执行步骤 209,否则执行步骤 211;

[0158] 具体地,预设函数返回的数据的格式为 $ret=fun(a,b)$, ret 表示函数调用结果, a 表示设备返回数据的内容, b 表示设备返回数据的长度,当函数调用不成功时, a 和 b 的内容为空;

[0159] 优选地,在本实施例中,检查预设函数返回的数据中的 ret ,若 ret 为 TRUE 则函数调用成功,若 ret 为 FALSE 函数调用不成功;

[0160] 步骤 209:获取预设函数返回的数据中的设备返回的数据的内容和长度;

[0161] 步骤 210:判断所述设备返回的数据是否合法,是则执行步骤 217,否则生成第一错误码,执行步骤 213;

[0162] 具体地,当设备返回的数据的长度不大于所述请求长度时,设备返回数据的长度合法;当设备返回的数据的内容为与测试请求包中的测试类型数据相对应的描述符时,设备返回的数据的内容合法,当设备返回的数据的长度和内容均合法时,设备返回的数据合法,否则设备返回的数据不合法;

[0163] 步骤 211:获取系统错误码,根据获取到的系统错误码判断函数调用不成功的原因是否为由设备处理不当产生的错误,是则执行步骤 212,否则执行步骤 215;

[0164] 具体地,通过系统调用 `GetLastError` 获取系统错误码,当错误码为 `0x01` 和 `0x57` 是表明函数调用不成功的原因不是由设备处理不当产生的错误,否则表明函数调用不成功的原因是由设备处理不当产生的错误;

[0165] 步骤 212:判断该由设备处理不当产生的错误是否可以忽略,是则执行步骤 215,否则生成第二错误码,执行步骤 213;

[0166] 具体地,所述配置信息中包括可以忽略的由设备处理不当产生的错误,例如,可设置系统错误码为 `0x1F` 和 `0x3E3` 的由设备处理不当产生的错误可以忽略;

- [0167] 步骤 213 :根据相应系统错误码设置相应错误描述 ;
- [0168] 步骤 214 :向日志文件中写入操作系统信息、错误码、错误描述以及测试请求包 ;
- [0169] 具体地,本步骤中的错误码包括设备错误码和系统错误码 ;
- [0170] 进一步地,本步骤还可以包括向日志文件中写入待测试设备路径以及该路径上各设备的信息 ;
- [0171] 步骤 215 :调用预设函数向所述待测试设备所在的 Hub 驱动端点发送标准的获取设备描述符命令 ;
- [0172] 步骤 216 :判断是否接收到待测试设备返回的应答信息,是则执行步骤 217,否则执行步骤 218 ;
- [0173] 步骤 217 :判断是否还有未发送的待测试请求包,是则返回执行步骤 206,否则执行步骤 218 ;
- [0174] 步骤 218 :将日志文件写入测试记录文件,测试结束 ;
- [0175] 进一步地,步骤 211 中,当函数调用不成功的原因是由设备处理不当产生的错误还可以包括 :判断系统错误码是否为第二预设错误码,是则执行步骤 A' ,否则执行步骤 213 ;其中,所述第二预设错误码为 6 ;
- [0176] 步骤 A' :重新打开待测试设备,调用预设函数重新向 Hub 驱动指定端点发送测试请求数据包 ;
- [0177] 步骤 B' :根据预设函数返回的数据判断函数调用是否成功,是则执行步骤 C' ,否则执行步骤 E' ;
- [0178] 步骤 C' :获取预设函数返回的数据中的设备返回数据的内容和长度 ;
- [0179] 步骤 D' :判断所述设备返回的数据是否合法,是则执行步骤 217,否则生成第一错误码,执行步骤 G' ;
- [0180] 具体地,当设备返回的数据的长度不大于所述请求长度时,设备返回数据的长度合法 ;当设备返回的数据的内容为与测试请求包中的测试类型数据相对应的描述符时,设备返回的数据的内容合法,当设备返回的数据的长度和内容均合法时,设备返回的数据合法,否则设备返回的数据不合法 ;
- [0181] 步骤 E' :获取系统错误码,根据获取到的系统错误码判断函数调用不成功的原因是否为由设备处理不当产生的错误,是则执行步骤 F' ,否则执行步骤 I' ;
- [0182] 步骤 F' :判断该由设备处理不当产生的错误是否可以忽略,是则执行步骤 I' ,否则生成第二错误码,执行步骤 G' ;
- [0183] 步骤 G' :根据相应系统错误码设置相应错误描述 ;
- [0184] 步骤 H' :向日志文件中写入操作系统信息、错误码、错误描述以及测试请求包 ;
- [0185] 步骤 I' :调用预设函数向所述待测试设备所在的 Hub 驱动端点发送标准的获取设备描述符命令 ;
- [0186] 步骤 J' :判断是否接收到待测试设备返回的应答信息,是则执行步骤 217,否则执行步骤 K' ;
- [0187] 步骤 K' :将日志文件写入测试记录文件,结束。
- [0188] 需要说明的两点是 :一、在本实施例中,请求长度是从小到大变化,容易想到的是还可以从大到小变化 ;二、在本实施例中,当配置信息中有过滤驱动的信息时,测试程序接

收到过滤驱动发送的测试消息后执行一个待执行测试,容易想到的是每次接收到过滤驱动发送的测试信息后发送全部(或若干个)待执行测试,在此不再赘述。

[0189] 以上所述的实施例只是本发明较优选的具体实施方式,本领域的技术人员在本发明技术方案范围内进行的通常变化和替换都应包含在本发明的保护范围内。

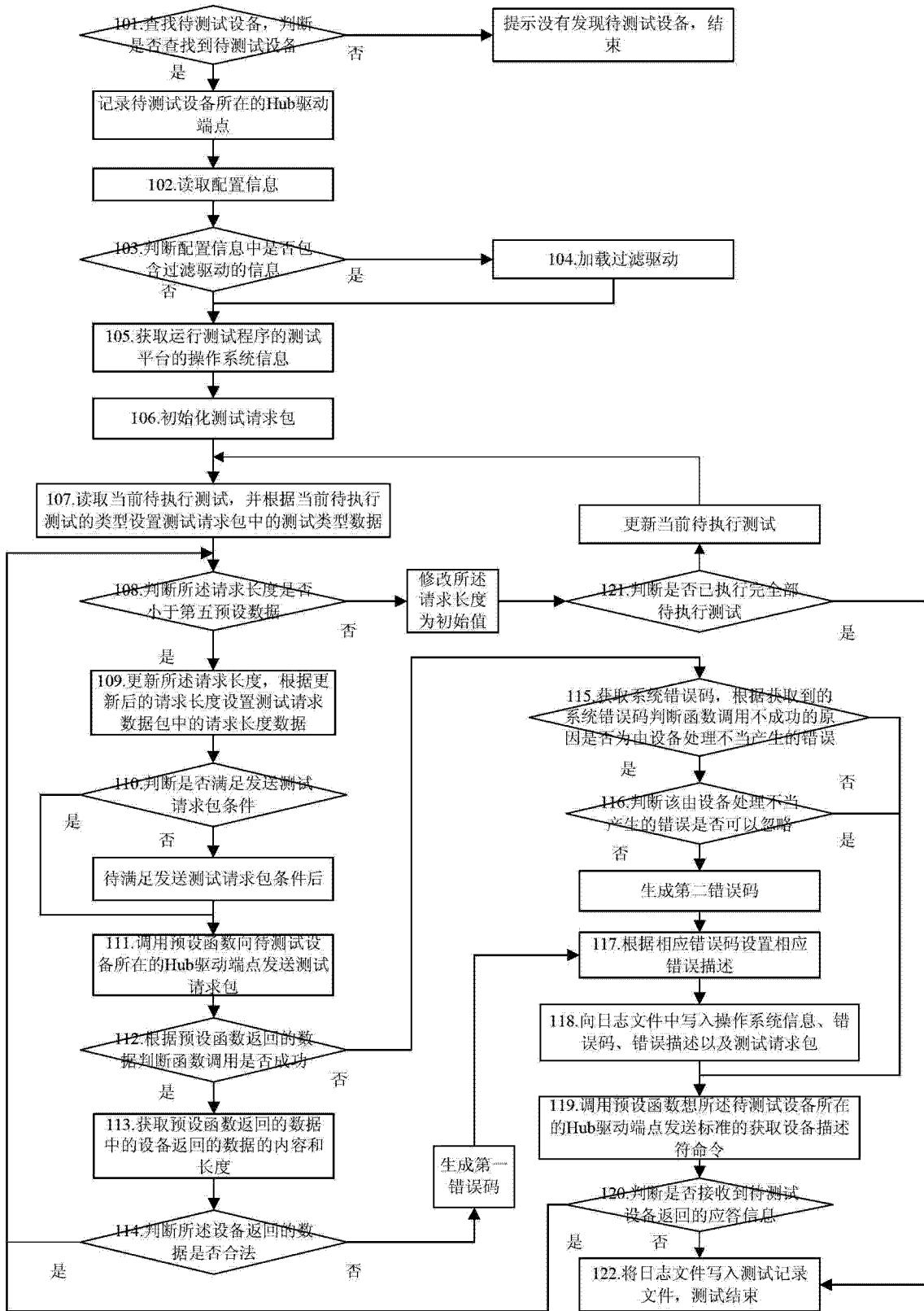


图 1

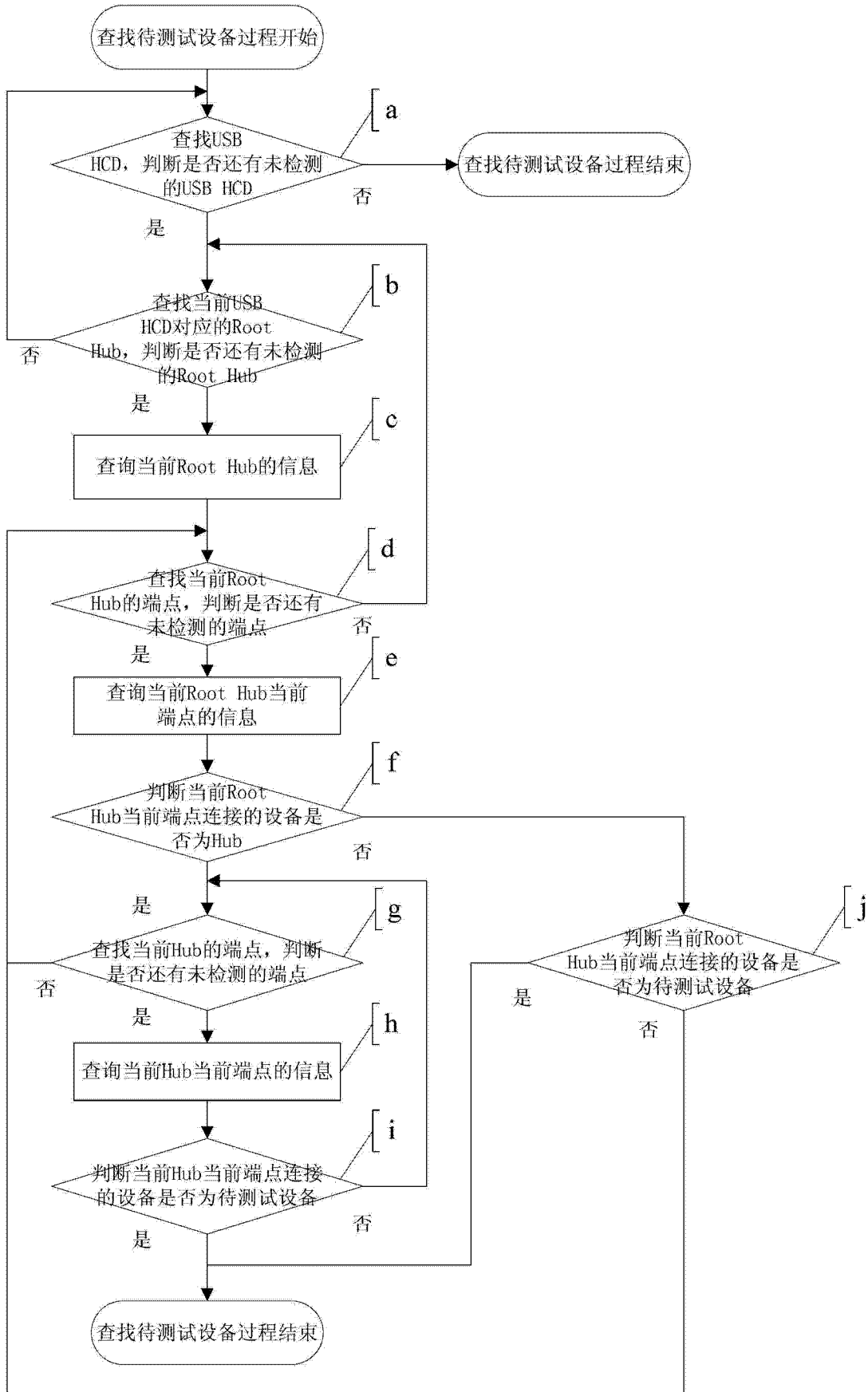


图 2

