



(12)发明专利

(10)授权公告号 CN 104412244 B

(45)授权公告日 2018.02.02

(21)申请号 201380035523.3

(22)申请日 2013.12.26

(65)同一申请的已公布的文献号
申请公布号 CN 104412244 A

(43)申请公布日 2015.03.11

(30)优先权数据
61/748,232 2013.01.02 US

(85)PCT国际申请进入国家阶段日
2014.12.29

(86)PCT国际申请的申请数据
PCT/JP2013/007633 2013.12.26

(87)PCT国际申请的公布数据
W02014/106883 JA 2014.07.10

(73)专利权人 精工爱普生株式会社

地址 日本东京

(72)发明人 高栖和弘 堤浩一郎 池田茂夫

(74)专利代理机构 中科专利商标代理有限责任
公司 11021

代理人 张莉

(51)Int.Cl.
G06F 13/00(2006.01)

(56)对比文件
US 2005046887 A1,2005.03.03,
CN 101136836 A,2008.03.05,
JP 2011124914 A,2011.06.23,

审查员 张文波

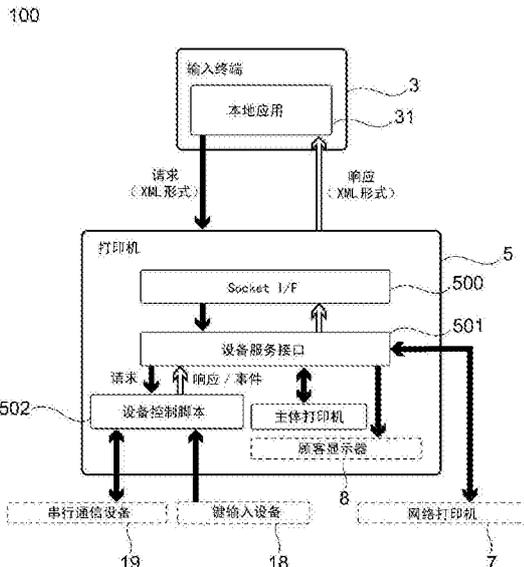
权利要求书3页 说明书33页 附图57页

(54)发明名称

设备控制系统以及设备控制系统的控制方法

(57)摘要

本发明能够通过进行了网络连接的终端对设备进行控制。本发明是一种设备控制系统，具备：终端(3)，其具备应用(31)；和打印机(5)，其对设备进行控制。终端(3)发送应用(31)所生成的用XML记述的请求。打印机(5)接收XML的请求，向终端(3)发送XML的响应，并在终端(3)通过XML的请求而要求了设备的控制的情况下对设备进行控制。



1. 一种设备控制系统,其特征在于,具备:
设备,其进行数据的输入或者输出;
终端,其执行应用,该应用发送标记语言的请求以及接收标记语言的响应;和
设备控制装置,其与所述设备相连接,进行基于所述标记语言的请求的所述设备的控制以及所述标记语言的响应的发送,

所述设备控制装置基于表示所述设备的控制的结果的数据,将所述标记语言的响应发送到所述终端,

所述应用将包含通信建立消息和指定所述设备的设备指定信息的所述请求发送到所述设备控制装置,其中所述通信建立消息是要求所述设备控制装置建立通信路径的消息,

所述设备控制装置在接收到包含所述通信建立消息和所述设备指定信息的所述请求时,将所述响应发送到所述应用,所述响应包含针对包含在所述请求中的所述通信建立消息的响应消息和表示能否对所述应用给予与所述设备指定信息对应的所述设备的排他使用权的信息。

2. 根据权利要求1所述的设备控制系统,其中,

所述应用基于包含在由所述设备控制装置发送的所述响应中的针对所述通信建立消息的响应消息,来建立与所述设备控制装置的所述通信路径。

3. 根据权利要求2所述的设备控制系统,其中,

所述应用在所述通信路径建立之后,将包含打开消息的所述请求发送到所述设备控制装置,其中所述打开消息是打开与连接至所述设备控制装置的所述设备的通信的消息。

4. 根据权利要求3所述的设备控制系统,其中,

所述设备控制装置在接收到包含所述打开消息的所述请求时,将包含针对包含在所述请求中的所述打开消息的响应消息的所述响应发送到所述应用。

5. 根据权利要求4所述的设备控制系统,其中,

所述应用接收由所述设备控制装置发送的包含针对所述打开消息的响应消息的所述响应,并基于针对所述打开消息的响应消息而成为能进行与所述设备的通信的状态。

6. 根据权利要求5所述的设备控制系统,其中,

所述应用在成为能进行与所述设备的通信的状态之后,将包含关闭与所述设备的通信的关闭消息的所述请求发送到所述设备控制装置。

7. 根据权利要求6所述的设备控制系统,其中,

所述设备控制装置接收包含所述关闭消息的所述请求,

将包含与所述关闭消息相对应的响应消息的所述响应发送到所述应用。

8. 根据权利要求7所述的设备控制系统,其中,

所述应用接收由所述设备控制装置发送的包含针对所述关闭消息的响应消息的所述响应,并基于针对所述关闭消息的响应消息来关闭与所述设备的通信。

9. 一种设备控制系统的控制方法,其特征在于,

所述设备控制系统具有进行数据的输入或者输出的设备、执行应用的终端、以及与所述设备相连接的设备控制装置,

在所述控制方法中,

将由所述应用生成的标记语言的请求发送到所述设备控制装置,

由所述设备控制装置接收所述请求，
基于所述标记语言的请求对所述设备进行控制，
生成所述标记语言的响应，
将所生成的所述响应向所述终端发送，
由所述设备控制装置将包含在所述请求中的信息发送到所述设备，
所述设备基于包含在所述请求中的信息来进行处理，
将基于包含在所述请求中的信息进行处理的结果发送到所述设备控制装置，
所述设备控制装置根据由所述设备发送的基于包含在所述请求中的信息进行处理的结果，将所述标记语言的响应发送到所述终端，

由所述应用将包含通信建立消息和指定所述设备的设备指定信息的所述请求发送到所述设备控制装置，其中所述通信建立消息是要求所述设备控制装置建立通信路径的消息，

由所述设备控制装置在接收到包含所述通信建立消息和所述设备指定信息的所述请求时，将所述响应发送到所述应用，所述响应包含针对包含在所述请求中的所述通信建立消息的响应消息和表示能否对所述应用给予与所述设备指定信息对应的所述设备的排他使用权的信息。

10. 根据权利要求9所述的设备控制系统的控制方法，其中，

由所述应用基于包含在由所述设备控制装置发送的所述响应中的针对所述通信建立消息的响应消息，来建立与所述设备控制装置的所述通信路径。

11. 根据权利要求10所述的设备控制系统的控制方法，其中，

在所述通信路径建立之后，由所述应用将包含打开消息的所述请求发送到所述设备控制装置，其中所述打开消息是打开与连接至所述设备控制装置的所述设备的通信的消息。

12. 根据权利要求11所述的设备控制系统的控制方法，其中，

由所述设备控制装置在接收到包含所述打开消息的所述请求时，将包含针对包含在所述请求中的所述打开消息的响应消息的所述响应发送到所述应用。

13. 根据权利要求12所述的设备控制系统的控制方法，其中，

由所述应用接收所述设备控制装置所发送的包含针对所述打开消息的响应消息的所述响应，并且

基于针对所述打开消息的响应消息而变得能进行与所述设备的通信。

14. 根据权利要求13所述的设备控制系统的控制方法，其中，

在变得能进行所述应用与所述设备的通信后使所述应用停止时，由所述应用将包含关闭消息的所述请求发送到所述设备控制装置，其中所述关闭消息是关闭与所述设备的通信的消息。

15. 根据权利要求14所述的设备控制系统的控制方法，其中，

由所述设备控制装置接收包含所述关闭消息的所述请求，
将包含与所述关闭消息相对应的响应消息的所述响应发送到所述应用。

16. 根据权利要求15所述的设备控制系统的控制方法，其中，

由所述应用接收所述设备控制装置所发送的包含针对所述关闭消息的响应消息的所述响应，并且

基于针对所述关闭消息的响应消息来关闭与所述设备的通信。

设备控制系统以及设备控制系统的控制方法

技术领域

[0001] 本发明涉及对设备进行控制的设备控制系统、以及设备控制系统的控制方法。

背景技术

[0002] 一直以来,在通过与网络相连接的终端对键盘、条形码扫描器等设备进行控制的系统中,准备了成为控制器的机器(计算机等)来用于设备的控制。

[0003] 在先技术文献

[0004] 专利文献

[0005] 专利文献1:JP特开2004-152256号公报

发明内容

[0006] 发明要解决的课题

[0007] 但是,存在这样的需求,即想要通过简单的构成来使用上述那样的设备。

[0008] 本发明鉴于上述情况而作,其目的在于提供一种能够通过进行了网络连接的终端对设备进行控制的设备控制系统、终端以及设备控制系统的控制方法。

[0009] 用于解决课题的手段

[0010] 为了达成上述目的,本发明的设备控制系统的特征在于,具备:设备,其进行连2: Connect消息。建立与设备服务接口501的通信路径。

[0011] 据的输入或者输出;终端,其执行应用,该应用发送标记语言的请求以及接收标记语言的响应;和设备控制装置,其与所述设备相连接,进行基于所述标记语言的请求的所述设备的控制以及所述标记语言的响应的发送。

[0012] 根据本发明,能够通过进行了网络连接的终端,来控制与设备控制装置相连接的设备。此外,终端只要是能够执行生成用XML记述的数据的装置的装置,便不受限制。因为XML通用性高,所以能够将各种各样的装置作为终端来使用。此外,终端的装置的开发很容易。

[0013] 此外,响应包含表示所述设备的控制的结果的数据。

[0014] 此外,所述应用将包含通信建立消息的所述请求发送到所述设备控制装置,其中所述通信建立消息是要求所述设备控制装置建立通信路径的消息,所述设备控制装置在接收到包含所述通信建立消息的所述请求时,将包含针对包含在所述请求中的所述通信建立消息的响应消息的所述响应发送到所述应用。

[0015] 此外,所述应用基于包含在由所述设备控制装置发送的所述响应中的针对所述通信建立消息的响应消息,来建立与所述设备控制装置的所述通信路径。

[0016] 此外,所述应用在所述通信路径建立之后,将包含打开消息的所述请求发送到所述设备控制装置,其中所述打开消息是打开与连接至所述设备控制装置的所述设备的通信的消息。

[0017] 此外,所述设备控制装置在接收到包含所述打开消息的所述请求时,将包含针对

包含在所述请求中的所述打开消息的响应消息的所述响应发送到所述应用。

[0018] 此外,所述应用接收由所述设备控制装置发送的包含针对所述打开消息的响应消息的所述响应,并基于针对所述打开消息的响应消息而成为能进行与所述设备的通信的状态。

[0019] 此外,所述应用在成为能进行与所述设备的通信的状态之后,将包含关闭与所述设备的通信的关闭消息的所述请求发送到所述设备控制装置。

[0020] 此外,所述设备控制装置接收包含所述关闭消息的所述请求,将包含与所述关闭消息相对应的响应消息的所述响应发送到所述应用。

[0021] 此外,所述应用接收由所述设备控制装置发送的包含针对所述关闭消息的响应消息的所述响应,并基于针对所述关闭消息的响应消息来关闭与所述设备的通信。

[0022] 此外,本发明的设备控制系统的控制方法的特征在于,所述设备控制系统具有进行数据的输入或者输出的设备、执行应用的终端、以及与所述设备相连接的设备控制装置,在所述控制方法中,将由所述应用生成的标记语言的请求发送到所述设备控制装置,由所述设备控制装置接收所述请求,生成所述标记语言的响应,将所生成的所述响应向所述终端发送。

[0023] 此外,由所述设备控制装置将包含在所述请求中的信息发送到所述设备,所述设备基于包含在所述请求中的信息来进行处理,将基于包含在所述请求中的信息进行处理的结果发送到所述设备控制装置,将包含由所述设备发送的基于包含在所述请求中的信息进行处理的结果的所述响应发送到所述终端。

[0024] 此外,由所述应用将包含通信建立消息的所述请求发送到所述设备控制装置,其中所述通信建立消息是要求所述设备控制装置建立通信路径的消息,由所述设备控制装置在接收到包含所述通信建立消息的所述请求时,将包含针对包含在所述请求中的所述通信建立消息的响应消息的所述响应发送到所述应用。

[0025] 此外,由所述应用基于包含在由所述设备控制装置发送的所述响应中的针对所述通信建立消息的响应消息,来建立与所述设备控制装置的所述通信路径。

[0026] 此外,在所述通信路径建立之后,由所述应用将包含打开消息的所述请求发送到所述设备控制装置,其中所述打开消息是打开与连接至所述设备控制装置的所述设备的通信的消息。

[0027] 此外,由所述设备控制装置在接收到包含所述打开消息的所述请求时,将包含针对包含在所述请求中的所述打开消息的响应消息的所述响应发送到所述应用。

[0028] 此外,由所述应用接收所述设备控制装置所发送的包含针对所述打开消息的响应消息的所述响应,基于针对所述打开消息的响应消息而变得能进行与所述设备的通信。

[0029] 此外,在变得能进行所述应用与所述设备的通信后使所述应用停止时,由所述应用将包含关闭消息的所述请求发送到所述设备控制装置,其中所述关闭消息是关闭与所述设备的通信的消息。

[0030] 此外,由所述设备控制装置接收包含所述关闭消息的所述请求,将包含与所述关闭消息相对应的响应消息的所述响应发送到所述应用。

[0031] 此外,由所述应用接收所述设备控制装置所发送的包含针对所述关闭消息的响应消息的所述响应,基于针对所述关闭消息的响应消息来关闭与所述设备的通信。

附图说明

- [0032] 图1是本发明的实施方式所涉及的设备控制系统的构成图。
- [0033] 图2是表示设备控制系统的功能构成的图。
- [0034] 图3A是表示能够从键盘取得的键码的一览的图。
- [0035] 图3B是表示能够从键盘取得的键码的一览的图。
- [0036] 图4是表示与智能打印机相连接的设备的连接例的图。
- [0037] 图5是智能打印机的外观立体图。
- [0038] 图6是表示连接器面板的图。
- [0039] 图7是表示无线LAN单元的连接例的图。
- [0040] 图8是表示环境构筑的工作流程的图。
- [0041] 图9是表示设备控制脚本的登记步骤的图。
- [0042] 图10是表示受控制的设备的登记步骤的图。
- [0043] 图11是表示所登记的设备的种类的一览的图。
- [0044] 图12是设备的登记的说明图。
- [0045] 图13是设备的登记的说明图。
- [0046] 图14是设备的登记的说明图。
- [0047] 图15是设备的登记的说明图。
- [0048] 图16是设备的登记的说明图。
- [0049] 图17是表示基本的编程顺序的图。
- [0050] 图18是表示所使用的接口的图。
- [0051] 图19A是表示数据的格式的图。
- [0052] 图19B是表示数据的内容的图。
- [0053] 图20A是表示通信数据的例子的图。
- [0054] 图20B是表示通信数据的例子的图。
- [0055] 图21是表示消息的要素的阵列的例子的图。
- [0056] 图22是表示消息的图。
- [0057] 图23是表示每个设备的信息数据的图。图24A是<admin_info>消息的说明图。
- [0058] 图24B是<admin_info>消息的说明图。
- [0059] 图24C是<admin_info>消息的说明图。
- [0060] 图24D是<admin_info>消息的说明图。
- [0061] 图24E是<admin_info>消息的说明图。
- [0062] 图25A是<open_device>消息的说明图。
- [0063] 图25B是<open_device>消息的说明图。
- [0064] 图25C是<open_device>消息的说明图。
- [0065] 图25D是<open_device>消息的说明图。
- [0066] 图25E是<open_device>消息的说明图。
- [0067] 图25F是<open_device>消息的说明图。
- [0068] 图26A是<close_device>消息的说明图。

- [0069] 图26B是<close_device>消息的说明图。
- [0070] 图26C是<close_device>消息的说明图。
- [0071] 图26D是<close_device>消息的说明图。
- [0072] 图26E是<close_device>消息的说明图。
- [0073] 图27A是<device_data>消息的说明图。
- [0074] 图27B是<device_data>消息的说明图。
- [0075] 图27C是<device_data>消息的说明图。
- [0076] 图27D是<device_data>消息的说明图。
- [0077] 图28A是<error>消息的说明图。
- [0078] 图28B是<error>消息的说明图。
- [0079] 图28C是<error>消息的说明图。
- [0080] 图29A是条形码扫描器用消息的说明图。
- [0081] 图29B是条形码扫描器用消息的说明图。
- [0082] 图29C是条形码扫描器用消息的说明图。
- [0083] 图30A是打印机用消息的说明图。
- [0084] 图30B是打印机用消息的说明图。
- [0085] 图31A是打印机用消息的说明图。
- [0086] 图31B是打印机用消息的说明图。
- [0087] 图32A是打印机控制XML文档的说明图。
- [0088] 图32B是打印机控制XML文档的说明图。
- [0089] 图33A是打印机控制XML文档的说明图。
- [0090] 图33B是打印机控制XML文档的说明图。
- [0091] 图33C是打印机控制XML文档的说明图。
- [0092] 图33D是打印机控制XML文档的说明图。
- [0093] 图34A是打印机控制XML文档的说明图。
- [0094] 图34B是打印机控制XML文档的说明图。
- [0095] 图34C是打印机控制XML文档的说明图。
- [0096] 图34D是打印机控制XML文档的说明图。
- [0097] 图34E是打印机控制XML文档的说明图。
- [0098] 图34F是打印机控制XML文档的说明图。
- [0099] 图34G是打印机控制XML文档的说明图。
- [0100] 图34H是打印机控制XML文档的说明图。
- [0101] 图34I是打印机控制XML文档的说明图。
- [0102] 图34J是打印机控制XML文档的说明图。
- [0103] 图34K是打印机控制XML文档的说明图。
- [0104] 图34L是打印机控制XML文档的说明图。
- [0105] 图34M是打印机控制XML文档的说明图。
- [0106] 图34N是打印机控制XML文档的说明图。
- [0107] 图34P是打印机控制XML文档的说明图。

- [0108] 图35是打印机控制XML文档的说明图。
- [0109] 图36A是打印机控制XML文档的说明图。
- [0110] 图36B是打印机控制XML文档的说明图。
- [0111] 图36C是打印机控制XML文档的说明图。
- [0112] 图36D是打印机控制XML文档的说明图。
- [0113] 图37A是打印机控制XML文档的说明图。
- [0114] 图37B是打印机控制XML文档的说明图。
- [0115] 图38A是打印机控制XML文档的说明图。
- [0116] 图38B是打印机控制XML文档的说明图。
- [0117] 图38C是打印机控制XML文档的说明图。
- [0118] 图38D是打印机控制XML文档的说明图。
- [0119] 图38E是打印机控制XML文档的说明图。
- [0120] 图38F是打印机控制XML文档的说明图。
- [0121] 图38G是打印机控制XML文档的说明图。
- [0122] 图38H是打印机控制XML文档的说明图。
- [0123] 图38I是打印机控制XML文档的说明图。
- [0124] 图38J是打印机控制XML文档的说明图。
- [0125] 图39A是打印机控制XML文档的说明图。
- [0126] 图39B是打印机控制XML文档的说明图。
- [0127] 图39C是打印机控制XML文档的说明图。
- [0128] 图39D是打印机控制XML文档的说明图。
- [0129] 图39E是打印机控制XML文档的说明图。
- [0130] 图39F是打印机控制XML文档的说明图。
- [0131] 图39G是打印机控制XML文档的说明图。
- [0132] 图39H是打印机控制XML文档的说明图。
- [0133] 图39I是打印机控制XML文档的说明图。
- [0134] 图39J是打印机控制XML文档的说明图。
- [0135] 图39K是打印机控制XML文档的说明图。
- [0136] 图40A是打印机控制XML文档的说明图。
- [0137] 图40B是打印机控制XML文档的说明图。
- [0138] 图41A是打印机控制XML文档的说明图。
- [0139] 图41B是打印机控制XML文档的说明图。
- [0140] 图42A是打印机控制XML文档的说明图。
- [0141] 图42B是打印机控制XML文档的说明图。
- [0142] 图43A是打印机控制XML文档的说明图。
- [0143] 图43B是打印机控制XML文档的说明图。
- [0144] 图44A是打印机控制XML文档的说明图。
- [0145] 图44B是打印机控制XML文档的说明图。
- [0146] 图45A是打印机控制XML文档的说明图。

- [0147] 图45B是打印机控制XML文档的说明图。
- [0148] 图46是打印机控制XML文档的说明图。
- [0149] 图47A是打印机控制XML文档的说明图。
- [0150] 图47B是打印机控制XML文档的说明图。
- [0151] 图48A是打印机控制XML文档的说明图。
- [0152] 图48B是打印机控制XML文档的说明图。
- [0153] 图49A是打印机控制XML文档的说明图。
- [0154] 图49B是打印机控制XML文档的说明图。
- [0155] 图50A是打印机控制XML文档的说明图。
- [0156] 图50B是打印机控制XML文档的说明图。
- [0157] 图50C是打印机控制XML文档的说明图。
- [0158] 图51A是打印机控制XML文档的说明图。
- [0159] 图51B是打印机控制XML文档的说明图。
- [0160] 图51C是打印机控制XML文档的说明图。
- [0161] 图52A是顾客显示器用消息的说明图。
- [0162] 图52B是顾客显示器用消息的说明图。
- [0163] 图53A是顾客显示器用消息的说明图。
- [0164] 图53B是顾客显示器用消息的说明图。
- [0165] 图54A是顾客显示器控制XML文档的说明图。
- [0166] 图54B是顾客显示器控制XML文档的说明图。
- [0167] 图55A是顾客显示器控制XML文档的说明图。
- [0168] 图55B是顾客显示器控制XML文档的说明图。
- [0169] 图55C是顾客显示器控制XML文档的说明图。
- [0170] 图56是顾客显示器控制XML文档的说明图。
- [0171] 图57A是顾客显示器控制XML文档的说明图。
- [0172] 图57B是顾客显示器控制XML文档的说明图。
- [0173] 图57C是顾客显示器控制XML文档的说明图。
- [0174] 图57D是顾客显示器控制XML文档的说明图。
- [0175] 图58A是顾客显示器控制XML文档的说明图。
- [0176] 图58B是顾客显示器控制XML文档的说明图。
- [0177] 图58C是顾客显示器控制XML文档的说明图。
- [0178] 图59A是顾客显示器控制XML文档的说明图。
- [0179] 图59B是顾客显示器控制XML文档的说明图。
- [0180] 图59C是顾客显示器控制XML文档的说明图。
- [0181] 图60A是顾客显示器控制XML文档的说明图。
- [0182] 图60B是顾客显示器控制XML文档的说明图。
- [0183] 图61A是顾客显示器控制XML文档的说明图。
- [0184] 图61B是顾客显示器控制XML文档的说明图。
- [0185] 图62A是顾客显示器控制XML文档的说明图。

- [0186] 图62B是顾客显示器控制XML文档的说明图。
- [0187] 图62C是顾客显示器控制XML文档的说明图。
- [0188] 图62D是顾客显示器控制XML文档的说明图。
- [0189] 图62E是顾客显示器控制XML文档的说明图。
- [0190] 图62F是顾客显示器控制XML文档的说明图。
- [0191] 图63是顾客显示器控制XML文档的说明图。
- [0192] 图64是顾客显示器控制XML文档的说明图。
- [0193] 图65是顾客显示器控制XML文档的说明图。
- [0194] 图66是顾客显示器控制XML文档的说明图。
- [0195] 图67A是POS键盘用消息的说明图。
- [0196] 图67B是POS键盘用消息的说明图。
- [0197] 图67C是POS键盘用消息的说明图。
- [0198] 图68A是POS键盘用消息的说明图。
- [0199] 图68B是POS键盘用消息的说明图。
- [0200] 图69A是POS键盘用消息的说明图。
- [0201] 图69B是POS键盘用消息的说明图。
- [0202] 图69C是POS键盘用消息的说明图。
- [0203] 图70A是串行通信设备用消息的说明图。
- [0204] 图70B是串行通信设备用消息的说明图。
- [0205] 图71A是串行通信设备用消息的说明图。
- [0206] 图71B是串行通信设备用消息的说明图。
- [0207] 图71C是串行通信设备用消息的说明图。
- [0208] 图72是设备控制脚本的说明图。
- [0209] 图73是设备控制脚本的说明图。
- [0210] 图74是设备控制脚本的说明图。
- [0211] 图75是设备控制脚本的说明图。
- [0212] 图76是设备控制脚本的说明图。
- [0213] 图77是设备控制脚本的说明图。
- [0214] 图78是设备控制脚本的说明图。
- [0215] 图79是设备控制脚本的说明图。
- [0216] 图80是设备控制脚本的说明图。
- [0217] 图81是设备控制脚本的说明图。
- [0218] 图82是表示程序的例子的图。
- [0219] 图83表示程序的环境构筑的流程。
- [0220] 图84A是程序的环境构筑的说明图。
- [0221] 图84B是程序的环境构筑的说明图。

具体实施方式

- [0222] 以下,参照附图对本发明的实施方式进行说明。

[0223] 图1是表示应用了本发明的实施方式的设备控制系统100的构成的图。设备控制系统100具备终端3以及打印机(智能打印机)5。

[0224] 终端3只要是与网络连接、能够进行Socket通信的设备即可,能够使用个人计算机、智能电话、或者平板电脑等。终端3能够进行Socket通信,具有能够处理作为标记语言的XML文档的环境,作为OS(操作系统),例如具备iOS(注册商标)、Android(注册商标)、Windows(注册商标)、Linux(注册商标)、Mac(注册商标)OS等。

[0225] 终端3以及打印机5经由通信网络而连接。在图1的例子中,设备控制系统100具有无线LAN接入点11,无线LAN接入点11被电缆连接至打印机5。通过对终端3与无线LAN接入点11进行无线连接,从而终端3与打印机5进行通信。

[0226] 打印机5作为控制对象的设备而连接网络打印机7、顾客显示器8、以及条形码扫描器9。网络打印机7经由网络与打印机5连接。网络打印机7是与打印机5所具备的本地打印机55(图2)不同的打印机,具备有线LAN或无线LAN的接口。顾客显示器8与后述的USB接口连接。条形码扫描器9通过后述的USB接口而被连接。一般将这些设备称作外围机器(peripheral),而在以下的说明中称作设备。能够与打印机5相连接的设备并不限定于图1所示的机器。例如,包含显示器、以及键盘等的键输入设备。还包含能通过OS标准的HID(Human interface device,人机接口设备)驱动器来控制的设备、能通过OS标准的串行通信驱动器来执行动作的串行通信设备、以及能够与串行通信设备同样地进行控制的USB设备。

[0227] 设备控制系统100的动作如下所述。

[0228] (1) 在终端3配置应用,并进行显示。

[0229] (2) 从应用向打印机5发送请求消息。

[0230] (3) 从打印机5向可控制的设备发送数据。

[0231] (4) 对与打印机5相连接的设备、网络打印机进行控制。

[0232] (5) 打印机5向终端3返回响应。

[0233] 图2是表示设备控制系统100的功能构成的图。

[0234] 设备控制系统100具备用作为标记语言的XML的指令体系来定义控制与打印机5相连接的设备(外围机器)的功能的设备XML。

[0235] 终端3具有应用31。应用31是安装于终端3的本地应用程序。关于应用31,设备应用程序接口(API)33对打印机5,通过Socket通信来发送XML格式的请求消息(Request)。

[0236] 在打印机5中,装入设备服务接口501以及设备控制脚本502。设备服务接口501以及设备控制脚本502对请求消息进行解释,向设备控制脚本502输出请求。设备控制脚本502对键输入设备18以及串行通信设备19进行控制。设备控制脚本502将键输入设备18以及串行通信设备19的事件、控制结果的响应,输出到设备服务接口501。在此,键输入设备18包含图1的键盘14,串行通信设备19包含图1的条形码扫描器9以及现金抽屉13。

[0237] 此外,设备服务接口501与本地打印机55收发数据。本地打印机55是打印机5所具备的打印机。此外,设备服务接口501与网络打印机7以及顾客显示器8收发数据。

[0238] 设备服务接口501基于网络打印机7、顾客显示器8以及本地打印机55的控制结果、设备控制脚本502的响应,将XML格式的响应(Response)输出到应用31。

[0239] 如此这样,打印机5通过设备服务接口501以及设备控制脚本502对设备进行控制。

图2的例子中受控制的设备为网络打印机7、顾客显示器8、键输入设备18、串行通信设备19、以及本地打印机55。

[0240] 按照以上的动作,能够利用设备控制系统100所具备的设备XML,由终端3对打印机5所连接的设备进行控制。

[0241] 设备XML具有以下特征。

[0242] • 无需为了设备的控制而另外准备成为控制器的机器(计算机等)。

[0243] • 通过打印机5的OS所具备的标准驱动器来执行动作的设备,不用安装驱动器就能够通过设备控制脚本502来使用。

[0244] • 若对设备进行访问,则设备被排他锁定。即使从多个终端3同时进行访问,控制也不会混在一起。若对设备进行了控制的终端3释放设备,则释放出的设备能够从其他的终端3进行控制。

[0245] 此外,设备API33具有以下特征。

[0246] • 印刷格线的指令在打印机5的本地打印机55或网络打印机7支持“格线指令”的情况下能够使用。

[0247] • 在打印机5或网络打印机7连接有蜂鸣器的情况下能够使用蜂鸣器的功能。

[0248] • 能够从键盘14取得的键码有限制。能够从键盘14取得的键码如图3A、图3B所示。

[0249] 对打印机5的构成进行说明。

[0250] 打印机5具备CPU、RAM、快闪ROM、NV(非易失性存储器)、视频控制器、辅助存储装置(SSD:solidstatedrive,固态硬盘)、接口、以及本地打印机(印刷部)。打印机5也可以具备扬声器。本地打印机是在80mm宽或58mm宽的卷筒纸上进行印刷的热敏打印机。

[0251] 在打印机5可以安装POS(point of sales)系统。打印机5的OS(operating system)例如是基于Windows(注册商标)的OS,存储在辅助存储装置中。

[0252] 打印机5具备由终端3对打印机5所连接的设备进行控制的软件即设备控制程序。因此,终端3不需要驱动程序。

[0253] 打印机5具备对打印机5所具备的本地打印机以及设备进行控制的软件即Windows(注册商标)标准的设备驱动程序(APD)、UPOS驱动器、OPOS驱动器等。

[0254] 在打印机5中,能够安装网络应用。由此,如图3所示,可以将打印机5作为应用服务器来利用。网络应用能够是例如PHP以及Perl的服务器端脚本、SQLite数据库访问(服务器端脚本)。

[0255] 图4是表示与打印机5相连接的设备连接例的图。

[0256] 在打印机5,能够连接上述的网络打印机7、顾客显示器8以及条形码扫描器9。能够将显示器12、现金抽屉13以及键盘14与打印机5连接。

[0257] 图5是打印机5的外观立体图。

[0258] 在打印机5的上部,设置卷筒纸盖51。若按下盖打开按钮52则卷筒纸盖打开,能够取放卷筒纸。在排出印刷后的卷筒纸的排出口,具有通过手动来剪切卷筒纸的手动剪切器53、以及剪切器盖54。剪切器盖54在打印机5的本地打印机发生了卡纸的情况下以及卷筒纸盖51打不开的情况下被打开。若打开剪切器盖54则手动剪切器53的刀片返回到基准位置。打印机5具备电源开关56B、复位按钮56A、LED显示部57、以及控制面板58。LED显示部57具备表示辅助存储装置的访问的磁盘访问LED、以及状态LED。状态LED用于通知OS的动作状态、

OS的待机模式、电源断开状态、OS的起动顺序中、CPU的高温警告。

[0259] 控制面板58具备电源LED、错误LED、卷筒纸LED、以及进给按钮。电源LED在电源正在被供给的情况下点亮。错误LED在通常动作时熄灭,在进行了复位时、以及检测到卷筒纸的末端从而印刷停止时点亮。纸张LED在卷筒纸的余量充足时熄灭,若余量变少则点亮,并在自诊断功能的执行时闪烁。若按下进给按钮则1行1行地或连续地输送卷筒纸。

[0260] 在打印机5的后部设置连接器盖59。若取下连接器盖59,则打印机5 的背面的连接器面板(connection panel)露出来。

[0261] 图6是表示连接器面板60的图。连接器面板60具有抽屉弹出连接器(drawer kick connector)61、以太网连接器62、USB连接器63、VGA连接器64、COM连接器65、线路输出66、以及DC输入67。连接器面板60相当于本发明的连接部。其中,在打印机5的设备中,包含打印机5主体所具备的本地打印机55。因此,连接部包含连接器面板60、以及打印机5内部的接口。

[0262] 在抽屉弹出连接器61,连接现金抽屉13或可选的蜂鸣器。以太网连接器62与网络连接。USB连接器63具有6个USB端口。在USB连接器63,连接顾客显示器8、条形码扫描器9、键盘14以及其他设备。在VGA连接器64连接显示器12。在COM连接器65连接进行串行通信的设备的串行接口。线路输出66与外部的扬声器连接。

[0263] 图7是表示无线LAN单元17的连接例的图。无线LAN单元17插入到USB延长线15中,USB延长线15插入到USB连接器63中。由此,能够对打印机5进行无线LAN连接。

[0264] 对设备服务接口501的环境构筑进行说明。

[0265] 图8表示环境构筑的工作流程。

[0266] 1.网络设定

[0267] 进行系统以及打印机5的网络设定。打印机5的网络设定能够通过以下任意一个方法来设定。

[0268] • 在打印机5,进行Windows(注册商标)的网络设定。

[0269] • 在外部机器(Windows的计算机),安装打印机5的设定应用即TMNet WinConfig并进行设定。

[0270] 2.将设备连接至打印机5

[0271] 将设备连接至打印机5。如上所述,可连接的设备为顾客显示器8、条形码扫描器9、显示器12、现金抽屉13、键盘14等。还包含能通过OS标准的HID驱动器来控制的设备、能通过OS标准的串行通信驱动器来执行动作的串行通信设备、以及能够与串行通信设备同样地进行控制的USB设备。在图6的连接器面板60上,COM连接器65仅有1个,但只要有了串行-USB转换线、以及串行-USB转换所对应的驱动程序,就能够连接多个串行通信设备。

[0272] 3.设备控制脚本的登记

[0273] 在通过打印机5来控制打印机5所对应的产品以外的设备的情况下,登记用户所准备的设备控制脚本。登记从网络浏览器进行。

[0274] 4.设备的登记

[0275] 将与打印机5相连接的设备登记到打印机5的软件中。登记从网络浏览器进行。

[0276] 在步骤1~5中所使用的网络浏览器包含在终端3运行的网络浏览器。

[0277] 参照图9来说明登记用户预先准备的设备控制脚本的方法。

[0278] 登记步骤如下所示。

- [0279] (1) 准备进行了自定义 (customize) 的设备控制脚本。
- [0280] (2) 接通打印机5的电源。
- [0281] (3) 起动网络浏览器,输入URL (http://打印机5的IP地址/WebConfig/)。
- [0282] (4) TMNetWebConfig起动。点击图9的画面的[Web服务设定]-[登记与删除]。
- [0283] (5) 显示“控制脚本”画面。
- [0284] 点击[进行登记的控制脚本]的[参照],选择进行登记的设备控制脚本。
- [0285] (6) 点击[登记]。
- [0286] 设备控制脚本被登记,并被一览显示于画面下的[已登记控制脚本]。
- [0287] 登记通过设备XML来控制的设备。如图10所示,通过网络浏览器显示TMNetWebConfig来登记设备。
- [0288] 登记步骤如下所述。
- [0289] (1) 起动网络浏览器,输入URL (http://打印机5的IP地址/WebConfig/)。
- [0290] (2) TMNetWebConfig起动。从图10的画面的[Web服务设定]-[设备登记]按照所登记的设备的种类进行点击。在图10中设备被显示于[设备登记]。点击进行登记的设备的种类。如图11所示,在本实施方式中设备的种类从打印机、显示器、键输入设备、串行通信设备、其他这5个种类中选择。设备的种类的详情如图11所示。
- [0291] (3) 显示各设备的设定画面。将所连接的设备登记到打印机5中。
- [0292] (3-1) 登记打印机的情况
- [0293] 通过以下步骤来确认打印机5的主体所具备的本地打印机已被登记。
- [0294] (3-1-1) 确认在图10的画面中在[已登记打印机]的设备ID“local_printer”中已经登记有“打印机5”。
- [0295] (3-1-2) 点击[测试打印]。确认通过本地打印机55打印出“TEST_PRINT”。
- [0296] (3-1-3) 登记所连接的网络打印机7。
- [0297] 通过以下步骤来登记。
- [0298] (3-1-3-1) 设定图12所示的项目(设备ID、类型、型号、IP地址、重试间隔),点击[登记]。
- [0299] (3-1-3-2) 若确认已被追加到[已登记打印机]中,则点击[测试打印]。确认通过所登记的网络打印机7打印出“TEST_PRINT”。
- [0300] (3-2) 登记显示器(顾客显示器8)的情况。
- [0301] 顾客显示器8的设备ID固定为“local_display”。通过以下步骤来登记所连接的顾客显示器8。
- [0302] (3-2-1) 在图10的画面中选择[使用]。
- [0303] (3-2-2) 设定图13的项目(通信设定、亮度设定),点击[登记]。
- [0304] (3-2-3) 点击[测试显示]。确认字符被显示于顾客显示器8。
- [0305] (3-3) 登记键输入设备18的情况
- [0306] 通过以下步骤来登记所连接的键输入设备18。
- [0307] (3-3-1) 设定图14的项目(设备ID、设备名、控制脚本),点击[登记]。
- [0308] (3-3-2) 若确认已被追加到[已登记键输入设备]中,则点击[动作测试]。
- [0309] (3-3-3) 通过网络浏览器来显示画面。操作键输入设备18,确认与操作相符的结果

被显示。

[0310] (3-4) 登记串行通信设备19的情况

[0311] 通过以下步骤来登记所连接的串行通信设备19。

[0312] (3-4-1) 对于所连接的串行通信设备19,设定图15的项目(设备ID、设备名、控制脚本、通信速度、数据比特、奇偶(parity)、停止位、自由控制),点击[登记]。

[0313] (3-4-2) 确认已被追加到[已登记串行通信设备]中。

[0314] (3-5) 登记其他的设备的情况

[0315] 通过以下步骤来登记所连接的设备。

[0316] (3-5-1) 对于所连接的设备,设定图16的项目(设备ID、控制脚本),点击[登记]。

[0317] (3-5-2) 确认已被追加到[已登记其他设备]中。

[0318] 在此,对使用了设备XML的应用开发的编程方法进行说明。

[0319] 设备XML的基本编程顺序如图17所示。

[0320] 图17的“应用”相当于应用31,“设备控制”相当于设备控制功能(设备服务接口501),“设备”表示受控制的设备。

[0321] 1:TCP连接的建立。从应用31对打印机5的Socket I/F进行TCP连接。

[0322] 2:Connect消息。建立与设备服务接口501的通信路径。

[0323] 3:应用31发送open_device消息(type_printer)。

[0324] 3.1:通过设备服务接口501向设备发送“Cmd()”指令。

[0325] 4:设备返回“Deviceopen”。

[0326] 4.1:设备服务接口501返回响应。

[0327] 6:应用31发送device_data_message。使用device_data_message来控制设备。

[0328] 6.1:向设备发送“Cmd()”指令。

[0329] 6.1.1:设备执行。

[0330] 在此,对Device_data_message进行响应。

[0331] 7.:设备返回执行结果。

[0332] 7.1:设备服务接口501返回device_data_message或error message。

[0333] 8:应用31发送close_device_message。切断与设备服务接口501的连接以及Socket I/F通信。

[0334] 8.1:通过设备服务接口501向设备发送“Cmd()”指令。

[0335] 9:设备关闭。

[0336] 9.1:设备服务接口501向应用31返回响应。

[0337] 在进行基于设备XML的控制的情况下,使用图18的接口。

[0338] 各数据项的格式如图19A所示,各数据的内容在图19B中示出。

[0339] 图20A、图20B中示出应用31与设备服务接口501之间的通信数据的例子。在图20A、图20B中,用“\0”来表示NULL字符。在图20A、图20B中,示出连接的建立、管理者信息的取得、设备的打开、设备的控制、设备的关闭的数据例。

[0340] 对<device_data>消息的<data>要素的阵列方法进行说明。

[0341] 在<data>标签内记述阵列时的编程方法如下所示。

[0342] • 在设为阵列的要素的开头指定属性array="true"。

- [0343] • 之后以同名的要素名来指定值。
- [0344] 图21中示出表现keycodes=[49、50、51、52];的情况的例子。
- [0345] 对设备XML进行说明。示出ePOS-设备XML的一览。
- [0346] 在ePOS-设备XML中,准备如下:
- [0347] • 消息(图22)
- [0348] 消息是从应用31对打印机5进行请求的消息、以及从打印机5向应用31的响应消息的XML。
- [0349] • 每个设备的消息数据(图23)
- [0350] 其是所控制的每个设备的保存请求以及响应的数据的XML。该形式成为<device_data>消息的<data>要素的子要素。
- [0351] 对每个设备的消息数据进行说明。
- [0352] 每个设备的消息数据为<device_data>消息的<data>要素的子要素,指定对设备进行控制的数据。
- [0353] 消息数据的构成要素根据设备的种类而不同。
- [0354] 在指定数据之前,用<type>要素来指定消息数据的种类,然后指定子要素的数据。每个设备种类能够使用的<type>要素如图23所示。
- [0355] 以下,对于消息进行说明。
- [0356] <connect>
- [0357] 其是包含在[响应]中的消息。
- [0358] 将连接已被建立的情况返回给应用。
- [0359] 应用31以本消息的接收为触发,进行之后说明的消息发送。
- [0360] [例]
- [0361] <connect></connect>
- [0362] <admin_info>
- [0363] 返回在打印机5中设定的管理者信息。是通过TMNet WebConfig而设定的信息。
- [0364] 其是包含在[请求]中的消息。
- [0365] 向打印机5要求管理者信息。
- [0366] [例]
- [0367] <admin_info></admin_info>
- [0368] 其是包含在[响应]中的消息。
- [0369] 将管理者信息返回给应用31。图24A中示出子要素、数据类型以及说明。
- [0370] • code
- [0371] 图24B中示出code要素。
- [0372] • data
- [0373] 图24C中示出data要素。
- [0374] 对于接收数据的字符串进行说明。图24D所示的字符串被进行转义(escape)处理。
- [例]图24E中示出正常时以及异常时的响应例。
- [0375] <open_device>
- [0376] 使与设备ID建立关联的设备能够使用。对于进行了请求的应用31,给予所指定的

设备的排他使用权。

[0377] 其是包含在[请求]中的消息。

[0378] 指定所要打开的设备。图25A中示出子要素、数据类型以及说明。

[0379] • data

[0380] 图25B中示出data要素。

[0381] [例]图25C中示出请求例。

[0382] 其是包含在[响应]中的消息。

[0383] 将设备的打开结果返回到应用31。图25D中示出子要素、数据类型以及说明。

[0384] • code

[0385] 图25E中示出code要素。

[0386] [例]图25F中示出正常时以及异常时的响应例。

[0387] <close_device>

[0388] 关闭已打开的设备。

[0389] 其是包含在[请求]中的消息。

[0390] 指定所要关闭的设备。图26A中示出子要素、数据类型以及说明。

[0391] [例]图26B中示出请求例。

[0392] 是包含在[响应]中的消息。

[0393] 将设备的关闭结果返回到应用31。图26C中示出子要素、数据类型以及说明。

[0394] • code

[0395] 图26D中示出code要素。

[0396] [例]图26E中示出正常时以及异常时的响应例。

[0397] <device_data>

[0398] 其是包含在[请求]中的消息。

[0399] 向设备发送数据。包含设备控制的命令、印刷/显示数据。图27A中示出子要素、数据类型以及说明。

[0400] [例]图27B中示出请求例。

[0401] 其是包含在[响应]中的消息。

[0402] 返回来自设备的数据。包含设备控制的结果、在设备中发生的事件、来自设备的输入数据。图27C中示出子要素、数据类型以及说明。

[0403] [例]图27D中示出响应例。

[0404] <error>

[0405] 其是包含在[响应]中的消息。

[0406] 返回设备控制命令、服务控制命令中的错误、其他共同错误。图28A中示出子要素、数据类型以及说明。

[0407] • code

[0408] 图28B中示出code要素。

[0409] [例]图28C中示出响应例。

[0410] 对于条形码扫描器的消息数据进行说明。

[0411] <type>ondata</type>

- [0412] 其是包含在[响应]中的消息。
- [0413] 将来自条形码扫描器的扫描数据返回给应用。图29A中示出子要素、数据类型以及说明。
- [0414] 对于接收数据的字符串进行说明。图29B的字符串被进行转义处理。
- [0415] [例]图29C中示出响应例。
- [0416] 对于打印机的消息数据进行说明。
- [0417] 关于印刷数据,在后述的“打印机控制用XML”中进行说明。
- [0418] `<type>print</type>`
- [0419] 其是包含在[请求]中的消息。
- [0420] 向打印机发送印刷数据与设定数据。图30A中示出子要素、数据类型以及说明。
- [0421] [例]图30B中示出请求例。
- [0422] `<type>onxmlresult</type>`
- [0423] 其是包含在[响应]中的消息。
- [0424] 将印刷结果返回给应用。图31A中示出子要素、数据类型以及说明。
- [0425] [例]图31B中示出响应例。
- [0426] 对控制打印机的XML文档即打印机控制用XML进行说明。
- [0427] 关于打印机的消息数据,在后述的“打印机的消息数据”中进行说明。
- [0428] `<epos-print>`
- [0429] 其是从应用发送给打印机的XML文档。向打印机请求所指定的功能的执行。`<epos-print>`具有要素。
- [0430] [属性]
- [0431] • xmlns
- [0432] 声明epos-print的名称。
- [0433] [子要素]图32A中示出子要素。
- [0434] [例]图32B中示出XML文档的例子。
- [0435] `<response>`
- [0436] 其是从打印机返回给应用的XML文档。
- [0437] [属性]
- [0438] • success
- [0439] 取得显示结果。图33A中示出属性值。
- [0440] • code
- [0441] 取得错误代码。图32B中示出属性值。
- [0442] • status
- [0443] 根据打印机的状态,通过10进制表达式来取得下列值的逻辑和。图32C中示出属性值。
- [0444] [例]图33D中示出XML文档例。
- [0445] `<text>`
- [0446] 指定进行打印的字符串。此外,也进行字符的装饰、或打印位置、换行量等与字符串相关的设定。

- [0447] 在文本打印后打印文本以外的情况下,执行换行或送纸。
- [0448] 在页面模式下,以字符的基准点(baseline dot)为基准,在当前的打印位置展开字符。
- [0449] [实体参照]
- [0450] 打印机控制所需的横向跳格(horizontal tab)、换行、以及下列记号使用图34A的实体参照来记述。
- [0451] [属性]
- [0452] • lang
- [0453] 指定对象语言。图34B中示出属性值。
- [0454] 能够打印的字符码取决于各打印机的规格。
- [0455] • font
- [0456] 指定字符字体。图34C中示出属性值。
- [0457] 若指定日语,则能够打印移位JIS码系统的字符。
- [0458] • smooth
- [0459] 设定平滑。若使平滑有效,则字符的打印品质提高。图34D中示出属性值。
- [0460] • dw
- [0461] 设定横倍角。图34E中示出属性值。
- [0462] 在一个要素中同时指定了dw属性与width属性的情况下,优先width属性的倍率指定。
- [0463] • dh
- [0464] 设定纵倍角。图34F中示出属性值。
- [0465] 在一个要素中同时指定了dh属性与height属性的情况下,优先height属性的倍率指定。
- [0466] • width
- [0467] 指定横向倍率。图34G中示出属性值。
- [0468] 在一个要素中同时指定了dw属性与width属性的情况下,优先width属性的倍率指定。
- [0469] • height
- [0470] 指定纵向倍率。图34H中示出属性值。
- [0471] 在一个要素中同时指定了dh属性与height属性的情况下,优先height属性的倍率指定。
- [0472] • reverse
- [0473] 设定黑白反转字符。图34I中示出属性值。
- [0474] • ul
- [0475] 设定下划线。图34J中示出属性值。
- [0476] • em
- [0477] 设定强调打印。图34K中示出属性值。
- [0478] • color
- [0479] 指定字符颜色。图34L中示出属性值。

- [0480] • x
- [0481] (初始值：“0”)
- [0482] 以点为单位来指定字符的打印开始位置。
- [0483] 在页面模式下,与position要素的横向的打印位置指定相同,而横向的打印位置会移动。
- [0484] • align
- [0485] 指定位置对齐。图34M中示出属性值。
- [0486] 在标准模式的情况下,align属性的指定在“行的开头的状态”下进行。
- [0487] 本要素中设定的align属性也被应用于image、logo、barcode、symbol要素的align属性。
- [0488] • rotate
- [0489] 设定倒立打印。图34N中示出属性值。
- [0490] 在页面模式下,rotate的指定无效。
- [0491] 在页面模式下用direction将打印方向设定为右→左(right_to_left)而进行180度旋转打印。
- [0492] 本要素中设定的rotate属性也被应用于barcode、symbol要素的rotate属性。
- [0493] • linespc
- [0494] (初始值：“30”)
- [0495] 以点为单位来指定每行的送纸量。
- [0496] [例]图34P中示出打印字符串的设定例。
- [0497] <feed>
- [0498] 指定送纸量。以点为单位或、以行为单位来指定送纸量。在没有送纸量的指定情况下,进行1行的送纸(换行)。此外,设定每行的换行量。
- [0499] [属性]
- [0500] • unit
- [0501] 以点为单位来指定送纸量。
- [0502] • line
- [0503] 以行为单位来指定送纸量。
- [0504] • linespc
- [0505] (初始值：“30”)
- [0506] 以点为单位来指定每行的送纸量。
- [0507] 在每行的送纸量为30点的情况下,最大能够指定到240行为止。
- [0508] 在标准模式与页面模式下,分别保持换行量。若指定linespc属性,则会影响到后续的<text>、<feed>的linespc属性。
- [0509] [例]图35中示出XML文档例。
- [0510] <image>
- [0511] 指定光栅形式的位图数据。(数据类型xs:base64Binary)
- [0512] 所谓的光栅形式就是以图像的左上为原点沿水平方向对像素进行扫描而得到的数据。

[0513] 在数据为2级灰度的情况下每个像素为1比特,在数据为16级灰度的情况下每个像素为4比特,从字节数据的上位比特起不断填入。此外,用0来进行填充,以便1行的扫描数据成为字节单位。

[0514] 在高速打印光栅图像的情况下,将align属性指定为left,将width属性的值指定为不超过打印机的纸张宽度的8的倍数。

[0515] 在页面模式下,以光栅图像的左下点为基准,在当前的打印位置展开光栅图像。打印位置不移动。

[0516] 在页面模式下,对打印位置进行设定,以使得光栅图像不会超出打印区域。

[0517] 因为存在以多级灰度光栅图像来印刷的条形码/二维码的读取品质的影响,所以以2级灰度来进行印刷。

[0518] 光栅形式的位图通过ePOS-Print XML生成工具来作成,或通过用户应用来作成。通过用户应用来作成的情况下要考虑以下几点。

[0519] • 2级灰度的情况:将图像的宽度设为8的倍数,或者将不足的bit设为0。

[0520] • 16级灰度的情况:将图像的宽度设为2的倍数,或者将不足的bit设为0。

[0521] [属性]

[0522] • width

[0523] 以点为单位来指定图像的宽度。

[0524] • height

[0525] 以点为单位来指定图像的高度。

[0526] • color

[0527] 指定字符颜色。图36A中示出属性值。

[0528] • align

[0529] 指定印刷位置。图36B中示出属性值。

[0530] 本要素中设定的align属性也被应用于text、logo、barcode、symbol要素的align属性。

[0531] • mode

[0532] (可选(option))

[0533] 指定彩色模式。图36C中示出属性值。

[0534] [例]图36D中示出XML文档例。

[0535] <logo>

[0536] 指定登记在打印机的NV存储器中的标识。

[0537] 对于标识,使用机型专用实用程序或标识登记实用程序(TMFLogo),预先将标识登记到打印机中。

[0538] 在页面模式下,以标识的左下点为基准,在当前的打印位置展开标识。

[0539] [属性]

[0540] • key1

[0541] 指定在NV标识登记时所设定的键码1的值。

[0542] • key2

[0543] 指定在NV标识登记时所设定的键码2的值。必须指定。

- [0544] • align
- [0545] 指定印刷位置。图37A中示出属性值。
- [0546] 本要素中设定的align属性也被应用于text、image、barcode、symbol要素的align属性。
- [0547] [例]图37B中示出XML文档例。
- [0548] <barcode>
- [0549] 用字符串来指定条形码数据。
- [0550] 在进行了未遵循条形码的标准的设定的情况下、或者条形码比打印机的打印区域大的情况下,不打印条形码。
- [0551] 在页面模式下,以条形码的左下点(除了HRI以外)为基准,在当前的打印位置展开条形码。
- [0552] [条形码的种类]
- [0553] 能够指定图38A、图38B、图38C所示的条形码。
- [0554] 在指定无法用字符串表现的二进制数据的情况下,用图38D所示的转义序列来进行指定。
- [0555] [属性]
- [0556] • type
- [0557] 指定条形码的种类。图38E中示出属性值。
- [0558] • hri
- [0559] 指定HRI的位置。图38F中示出属性值。
- [0560] • font
- [0561] 指定HRI字体。图38G中示出属性值。
- [0562] • width
- [0563] (初始值:“3”)
- [0564] 以点为单位来指定模块的宽度。用2~6的整数値来指定。
- [0565] • height
- [0566] (初始值:“162”)
- [0567] 以点为单位来指定模块的高度。
- [0568] • align
- [0569] 指定印刷位置。图38H中示出属性值。
- [0570] 本要素中设定的align属性也被应用于text、image、logo、symbol要素的align属性。
- [0571] • rotate
- [0572] 指定倒立打印。图38I中示出属性值。
- [0573] 本要素中设定的rotate属性也被应用于text、symbol要素的rotate属性。
- [0574] [例]图38J中示出XML文档例。
- [0575] <symbol>
- [0576] 对二维码进行印刷。
- [0577] 用字符串来指定二维码。

- [0578] 在进行了未遵循二维码的标准的设定的情况下、或者二维码比打印机的打印区域大的情况下,不打印二维码。
- [0579] 在页面模式下,以二维码的左上点为基准,在当前的打印位置展开二维码。
- [0580] [二维码的种类]
- [0581] 能够指定图39A、图39B所示的二维码。
- [0582] 在指定无法用字符串表现的二进制数据的情况下,用图39C所示的转义序列来进行指定。
- [0583] [属性]
- [0584] • type
- [0585] 指定二维码的种类。图39D中示出属性值。
- [0586] • level
- [0587] 指定纠错等级。图39E中示出属性值。
- [0588] 等级按照二维码的种类来进行选择。
- [0589] 在MaxiCode、二维GS1DataBar的情况下,选择default。
- [0590] • width
- [0591] 以点为单位来指定1个模块的宽度。详细如图39F所示。
- [0592] • height
- [0593] 以点为单位来指定1个模块的高度。详细如图39G所示。
- [0594] • size
- [0595] 详细如图39H所示。
- [0596] • align
- [0597] 指定印刷位置。图39I中示出属性值。
- [0598] 本要素中设定的align属性也被应用于text、image、logo、barcode要素的align属性。
- [0599] • rotate
- [0600] 指定倒立打印。图39J中示出属性值。
- [0601] 本要素中设定的rotate属性也被应用于text、barcode要素的rotate属性。
- [0602] [例]图39K中示出XML文档例。
- [0603] <hline>
- [0604] 指定横格线。
- [0605] [属性]
- [0606] • x1
- [0607] 指定横格线的描绘开始位置(以点为单位)。
- [0608] • x2
- [0609] 指定横格线的描绘结束位置(以点为单位)。
- [0610] • style
- [0611] (可选)
- [0612] 指定格线的种类。图40A中示出属性值。
- [0613] [例]图40B中示出XML文档例。

- [0614] <vline-begin>
- [0615] 开始纵格线的描绘。
- [0616] 在页面模式下不能使用。
- [0617] 纵格线直到由后述的<vline-end>指定结束为止进行描绘。本要素与vline-end一起使用。
- [0618] [属性]
- [0619] • x
- [0620] 指定纵格线的描绘开始位置(以点为单位)。
- [0621] • style
- [0622] (可选)
- [0623] 指定格线的种类。图41A中示出属性值的例子。
- [0624] [例]图41B中示出XML文档例。
- [0625] <vline-end>
- [0626] 结束纵格线的描绘。
- [0627] 本要素与上述的vline-begin一起使用。
- [0628] [属性]
- [0629] • x
- [0630] 指定纵格线的描绘结束位置(以点为单位)。
- [0631] • style
- [0632] (可选)
- [0633] 指定格线的种类。图42A中示出属性值的例子。
- [0634] [例]图42B中示出XML文档例。
- [0635] <page>
- [0636] 从标准模式向页面模式切换。
- [0637] [<page>的要素]
- [0638] 在<page>中能够使用图43A所示的要素。
- [0639] [例]图43B中示出XML文档例。
- [0640] <area>
- [0641] 指定页面模式的印刷区域。
- [0642] 以绝对原点为基准,指定原点、宽度、高度,来设定打印区域。如图44A所示,绝对原点为可打印区域的左上的点。
- [0643] 按照打印内容来指定打印区域。在打印数据超出了打印区域的情况下,会成为打印数据在中途发生了中断的打印结果。
- [0644] 本要素在page要素内使用。
- [0645] [属性]
- [0646] • x
- [0647] (初始值:“0”)
- [0648] 以点为单位来指定横向的原点。
- [0649] • y

- [0650] (初始值：“0”)
- [0651] 以点为单位来指定纵向的原点。
- [0652] • width
- [0653] (初始值：根据机型而不同)
- [0654] 以点为单位来指定打印区域的宽度。
- [0655] • height
- [0656] (初始值：根据机型而不同)
- [0657] 以点为单位来指定打印区域的高度。
- [0658] 打印区域的宽度与高度配合打印方向的设定来确定。存在打印数据发生中断的情况。
- [0659] [例]图44B中示出XML文档例。
- [0660] <direction>
- [0661] 指定页面模式的打印方向。
- [0662] 指定打印方向,来进行打印区域的旋转。与打印区域的旋转相配合,打印区域的起点发生移动。
- [0663] 本要素在page要素内使用。
- [0664] [属性]
- [0665] • dir
- [0666] 指定旋转方向。图45A中示出属性值。
- [0667] [例]图45B中示出XML文档例。
- [0668] <position>
- [0669] 指定页面模式的打印位置。
- [0670] 以打印区域的起点为基准来指定打印位置。打印区域的起点配合打印区域的旋转而发生移动。
- [0671] 本要素在page要素内使用。
- [0672] [属性]
- [0673] • x
- [0674] (初始值：“0”)
- [0675] 以点为单位来指定横向位置。
- [0676] • y
- [0677] (初始值：“21”)
- [0678] 以点为单位来指定纵向位置。
- [0679] 印字开始位置(坐标)按照打印内容来指定。将以下作为参考。
- [0680] • 打印字符串的情况
- [0681] 指定最初的字符的基线左端。
- [0682] 在以标准大小且左对齐的方式进行打印的情况下能够可选。在印刷高度是2倍的字符的情况下,将y指定为“42”以上。
- [0683] • 打印条形码的情况
- [0684] 指定符号(symbol)的左下。将条形码的高度指定为y。

- [0685] • 打印图形/标识的情况
- [0686] 指定图形数据的左下。将图形数据的高度指定为y。
- [0687] • 打印二维码的情况
- [0688] 指定符号的左上。
- [0689] [例]图46中示出XML文档例。
- [0690] <line>
- [0691] 在页面模式下描绘直线。
- [0692] 不能描绘斜线。
- [0693] 本要素在page要素内使用。
- [0694] [属性]
- [0695] • x1
- [0696] 指定横格线的描绘开始位置(以点为单位)。
- [0697] • y1
- [0698] 指定纵格线的描绘开始位置(以点为单位)。
- [0699] • x2
- [0700] 指定横格线的描绘结束位置(以点为单位)。
- [0701] • y2
- [0702] 指定纵格线的描绘结束位置(以点为单位)。指定格线的种类。
- [0703] • style
- [0704] (可选)
- [0705] 指定格线的种类。图47A中示出属性值。
- [0706] [例]图47B中示出XML文档例。
- [0707] <rectangle>
- [0708] 在页面模式下描绘长方形。
- [0709] 本要素在page要素内使用。
- [0710] 在标准模式下使用hline要素、vline-begin要素、vline-end要素。
- [0711] [属性]
- [0712] • x1
- [0713] 指定横格线的描绘开始位置(以点为单位)。
- [0714] • y1
- [0715] 指定纵格线的描绘开始位置(以点为单位)。
- [0716] • x2
- [0717] 指定横格线的描绘结束位置(以点为单位)。
- [0718] • y2
- [0719] 指定纵格线的描绘结束位置(以点为单位)。
- [0720] • style
- [0721] (可选)
- [0722] 指定格线的种类。图48A中示出属性值的例子。
- [0723] [例]图48B中示出XML文档例。

- [0724] <cut>
- [0725] 指定纸张剪切。
- [0726] 在标准模式下使用。
- [0727] 在“行的开头的状态”下执行。若对纸张进行剪切则成为“行的开头的状态”。
- [0728] [属性]
- [0729] • type
- [0730] (可选)
- [0731] 指定剪切的种类。图49A中示出属性值的例子。
- [0732] [例]图49B中示出XML文档例。
- [0733] <pulse>
- [0734] 指定向抽屉弹出连接器输出信号。能够根据机型使蜂鸣器鸣响。
- [0735] [属性]
- [0736] • drawer
- [0737] (可选)
- [0738] 指定抽屉弹出连接器。图50A中示出属性值。
- [0739] • time
- [0740] (可选)
- [0741] 指定抽屉弹出信号的有效(ON)时间。图50B中示出属性值。
- [0742] [例]图中示出XML文档例。
- [0743] <sound>
- [0744] 使蜂鸣器鸣响。
- [0745] [属性]
- [0746] • pattern
- [0747] (可选)
- [0748] 指定蜂鸣器的音色。图51A中示出属性值。
- [0749] • repeat
- [0750] (可选,初始值:“1”)
- [0751] 指定重复次数。图51B中示出属性值。
- [0752] [例]图中示出XML文档例。
- [0753] 对于顾客显示器的消息数据进行说明。
- [0754] 关于显示数据,在后述的“顾客显示器控制用XML”中进行说明。
- [0755] <type>display</type>
- [0756] [请求]
- [0757] 向顾客显示器发送显示数据与设定数据。图52A中示出子要素。
- [0758] [例]图52B中示出请求例。
- [0759] <type>onxmlresult</type>
- [0760] [响应]
- [0761] 将发送给顾客显示器后的结果返回给应用。图53A中示出子要素。
- [0762] [例]图53B中示出响应例。

- [0763] 对于控制顾客显示器的XML文档即顾客显示器控制用XML进行说明。
- [0764] 关于顾客显示器的消息数据,如上述的“顾客显示器的消息数据”所说明的那样。
- [0765] <epos-display>
- [0766] 其是从应用向顾客显示器发送的XML文档。向顾客显示器请求所指定的功能的执行。<epos-display>具有顾客显示器的控制所需要的子要素。
- [0767] [属性]
- [0768] • xmlns
- [0769] 声明epos-display的名称。图54A中示出子要素。
- [0770] [例]图54B中示出XML文档例。
- [0771] <response>
- [0772] 其是从顾客显示器返回给应用的XML文档。
- [0773] [属性]
- [0774] • success
- [0775] 取得显示结果。图55A中示出属性值。
- [0776] • code
- [0777] 取得错误代码。图55B中示出属性值。
- [0778] • status
- [0779] 始终被设为“0”。
- [0780] [例]图55C中示出XML文档例。
- [0781] <window>
- [0782] 控制窗口的作成、删除、当前窗口的移动。
- [0783] • 在指定了属性number、x、y、width、height的情况下,进行指定的窗口的作成。在定义窗口的情况下按照不会与已定义窗口重叠的方式来定义。此外,指定容纳在显示器的尺寸(20位×2行)内的大小。
- [0784] • 在指定了属性number destroy="true"的情况下,删除指定的窗口。
- [0785] • 在并未指定与窗口的作成、删除相符的属性而指定了number的情况下,使当前窗口移动至指定的窗口。
- [0786] [属性]
- [0787] • number
- [0788] 指定对象的窗口编号。图56中示出属性值。
- [0789] • x
- [0790] 用整数值来指定窗口作成时的x坐标(1~20)。
- [0791] • y
- [0792] 用整数值来指定窗口作成时的y坐标(1~2)。
- [0793] • width
- [0794] 用整数值来指定窗口作成时的宽度(1~20)。
- [0795] • height
- [0796] 用整数值来指定窗口作成时的高度(1~2)。
- [0797] • scrollmode

- [0798] 指定窗口作成时作成窗口的滚动方法。图57A中示出属性值。
- [0799] • destroy
- [0800] 指定是否删除通过number指定的窗口。图57B中示出属性值。
- [0801] [错误条件]
- [0802] 在错误的情况下,会返回图57C的值。
- [0803] [例]图57D中示出XML文档例。
- [0804] <text>
- [0805] 控制字符串的显示。
- [0806] • 在指定了属性x以及y的情况下,从所指定的光标位置起进行显示。
- [0807] • 在未指定属性x以及y的情况下,从当前的光标位置起进行显示。
- [0808] [属性]
- [0809] • x
- [0810] 用整数值来指定文本显示的x坐标(1~20)。
- [0811] • y
- [0812] 用整数值来指定文本显示的y坐标(1~2)。
- [0813] • reverse
- [0814] 指定显示字符的明暗反转。图58A中示出属性值。
- [0815] • lang
- [0816] 指定文本显示的语言。省略时维持当前设定。图58B中示出属性值。
- [0817] [例]图58C中示出XML文档例。
- [0818] <cursor>
- [0819] 控制光标位置以及显示设定。
- [0820] • 通过x、y的指定来指定在显示器内的光标坐标。
- [0821] • 通过moveto的指定来指定在当前窗口内的光标位置。
- [0822] • 在同时指定了x、y、moveto的情况下,哪个都不实施。
- [0823] [属性]
- [0824] • x
- [0825] 用整数值指定光标的x坐标(1~20)。
- [0826] • y
- [0827] 用整数值指定光标的y坐标(1~2)。
- [0828] • moveto
- [0829] 指定当前窗口内的光标位置。图59A中示出属性值。
- [0830] • type
- [0831] 指定光标的显示方法。省略时维持当前设定。图59B中示出属性值。
- [0832] [例]图59C中示出XML文档例。
- [0833] <blink>
- [0834] 控制画面的闪烁。以在interval中指定的间隔反复闪烁。
- [0835] 实际的闪烁间隔以50ms为单位进行进位(round up)。
- [0836] 例如interval在1~50的情况下成为50ms,在51~100的情况下成为100ms。

- [0837] [属性]
- [0838] • interval
- [0839] 用整数值来指定闪烁的间隔。图60A中示出属性值。
- [0840] [例] 图60B中示出XML文档例。
- [0841] <brightness>
- [0842] 控制显示器的亮度。
- [0843] [属性]
- [0844] • value
- [0845] 用百分比来指定亮度。图61A中示出属性值。
- [0846] [例] 图61B中示出XML文档例。
- [0847] <marquee>
- [0848] 控制字符串的滚动显示 (marquee display)。
- [0849] 以在uwait中指定的间隔1个字符1个字符地显示所指定的字符串。到末尾为止显示完成后,待机由rwait指定的时间,再次从开头进行显示。
- [0850] [属性]
- [0851] • format
- [0852] 指定滚动显示模式。图62A中示出属性值。
- [0853] • repeat
- [0854] 指定显示的反复次数。图62B中示出属性值。
- [0855] • uwait
- [0856] 用毫秒来指定1字符显示的待机间隔。图62C中示出属性值。
- [0857] • rwait
- [0858] 用毫秒来指定显示字符串的末尾之后的待机间隔。图62D中示出属性值。
- [0859] • lang
- [0860] 指定显示语言。省略时维持当前设定。图62E中示出属性值。
- [0861] [例] 图62F中示出XML文档例。
- [0862] <clock>
- [0863] 在显示器画面的最下段右端显示时刻。
- [0864] 显示时刻是打印机5的OS所管理的本地时间。
- [0865] [例] 图63中示出XML文档例。
- [0866] <clear>
- [0867] 清除当前窗口的显示。
- [0868] 在窗口不存在的情况下,清除显示器整体的显示。
- [0869] [例] 图64中示出XML文档例。
- [0870] <reset>
- [0871] 将显示器的显示初始化。通过初始化而成为以下状态。
- [0872] • 显示字符全部被清除,已登记的所有窗口都被废弃。
- [0873] • 光标返回到默认状态,移动至显示器原点。
- [0874] • 显示闪烁、显示亮度返回到默认状态。

[0875] [例]图65中示出XML文档例。

[0876] <command>

[0877] 执行任意的ESC/POS指令。

[0878] 所指定的指令通过16进制字符串来指定。

[0879] [例]图66中示出XML文档例。

[0880] 对于POS键盘的消息数据进行说明。

[0881] <type>onkeypress</type>

[0882] [响应]

[0883] 将来自POS键盘的输入数据返回给应用。图67A中示出子要素。

[0884] (关于接收数据的字符串)

[0885] 图67B所示的字符串被进行转义处理。

[0886] [例]图67C中示出响应例。

[0887] <type>setprefix</type>

[0888] 为了检测来自POS键盘的输入,指定判断为字符串的开始的键码。在本消息所指定的键码被输入了的情况下,将从该处起到Enter键被按下为止的字符串作为onstring消息来进行通知。例如,在从POS键盘进行条形码的代替输入的情况下使用。在将onstring消息停止的情况下,发送无 keycode的setprefix消息。

[0889] [请求]

[0890] 指定用于检测来自POS键盘的输入的、判断为字符串的开始的键码。图68A中示出子要素。

[0891] [例]图68B中示出请求例。

[0892] <type>onstring</type>

[0893] [响应]

[0894] 将来自POS键盘的输入数据返回给应用。

[0895] 通知检测到从由setprefix指定的键码中的任意一个到Enter为止的输入。在自变量中除了所输入的一系列字符串之外,还取得检测到的键码的信息。图69A中示出子要素。

[0896] (关于接收数据的字符串)

[0897] 图69B的字符串被进行转义处理。

[0898] [例]图69C中示出响应例。

[0899] 对于串行设备的信息数据进行说明。

[0900] <type>sendcommand</type>

[0901] [请求]

[0902] 将指令发送到串行设备。图70A中示出子要素。

[0903] [例]图70B中示出请求例。

[0904] <type>oncommandreply</type>

[0905] [响应]

[0906] 将来自串行设备的数据返回给应用。图71A中示出子要素。

[0907] • status

[0908] 图71B中示出属性值。

- [0909] [例]图71C中示出响应例。
- [0910] 接下来,对设备控制脚本502进行说明。
- [0911] 首先,对编程进行说明。
- [0912] 图72是关于设备控制脚本的使用的说明图。
- [0913] 通过使用由本发明的设备控制功能提供的设备控制脚本用API,能够进行设备的数据处理的自定义,或者从应用31来使用新设备。
- [0914] 如图72所示,设备控制脚本502具有设备连接503和客户端连接504。在该图72中将高功能打印机5的设备总称为设备20来表示。
- [0915] 若包含上述的open_device消息在内的XML数据(请求)被发送,则打印机5上的Socket I/F500接收XML数据。Socket I/F500将所接收到的数据传递到设备服务接口501。设备服务接口501生成对象,使得能够使用与open_device消息所要求的设备相对应的设备控制脚本。通过所生成的对象,能够控制设备20。
- [0916] [设备控制脚本的对象]
- [0917] 从设备服务接口501将图73所示的对象传递给设备控制脚本502。通过使用这些客户端连接(ClientConnection)对象、以及设备连接(DeviceConnection)对象,从而设备控制脚本502能够与应用31以及设备20进行通信。
- [0918] 对于使用了设备控制脚本的对象的功能进行说明。
- [0919] 使用设备控制脚本API能够使用以下的功能。
- [0920] • 能够调用应用31侧的设备对象的任意的事件。
- [0921] • 能够向设备发送数据。
- [0922] • 能够接收从设备产生的数据。
- [0923] 对于设备控制脚本的构成进行说明。
- [0924] 以包含以下条件的方式对设备控制脚本编写代码。
- [0925] • 设备控制所需要的代码汇总在1个文件中记述。
- [0926] (在使用TMNetWebConfig来进行设备的设定时,只能登记1个文件)
- [0927] • 使文件名的第一个点"."为止的名称与构造器名称相同。
- [0928] [例]文件名:Keyboard_Generic.ver1.0.js→构造器的名称:Keyboard_Generic
- [0929] • 为了构造器的外部参照,进行exports声明。
- [0930] [例]exports.Keyboard_Generic=Keyboard_Generic
- [0931] • 使设备控制脚本具有图74以及图75所示的属性。在构造器中设定适当的名称。
- [0932] DEVICE_TYPE属性(对象种类:String)
- [0933] 图74中示出设定值的一览。
- [0934] DEVICE_GROUP属性(对象种类:String)
- [0935] 图75中示出设定值的一览。
- [0936] • 为了接收设备所产生的数据而准备onDeviceData方法。详情在后述的“设备控制脚本名对象”进行说明。
- [0937] • 准备与在应用31侧执行动作的设备对象的方法相对应的方法。详情在后述的“任意事件”进行说明。
- [0938] 图76中示出设备控制脚本502的构成例。

- [0939] 示出设备控制脚本API的一览。
- [0940] 在设备控制脚本API中,准备有下面的对象。
- [0941] • ClientConnection对象(图77中示出API的一览。)
- [0942] • DeviceConnection对象(图78中示出API的一览。)
- [0943] • 设备控制脚本名对象(图79中示出API的一览。)
- [0944] 对于ClientConnection对象进行说明。
- [0945] 该对象是传递至设备控制脚本502的构造器的第1参数的对象。
- [0946] • send
- [0947] 向在浏览器中执行动作的设备对象发送数据。
- [0948] [语法]
- [0949] send(event,data);
- [0950] [参数]
- [0951] • event:对象种类:(String)
- [0952] 设定设备对象的事件名。
- [0953] • data:对象种类:(Object)
- [0954] 指定向设备对象的事件传递的数据。
- [0955] 图80中示出例子。在该例中,设备对象的onkeypress事件被调用,使用onkeypress事件的data参数,从data.keycode取得49,从data.ascii取得'1'。
- [0956] 对于DeviceConnection对象进行说明。
- [0957] 该对象是向设备控制脚本502的构造器的第2参数传递的对象。
- [0958] • send
- [0959] 向串行通信设备发送数据。
- [0960] [语法]
- [0961] send(data);
- [0962] [参数]
- [0963] • data:数据种类:(Buffer)
- [0964] 指定发送给设备的数据。
- [0965] 对于设备控制脚本名对象进行说明。
- [0966] • onDeviceData事件(能够通过HID驱动器来执行动作的输入设备)
- [0967] 该事件是接收从键输入设备检测到的数据的事件。
- [0968] 在键输入设备用的设备控制脚本502的情况下,以这种形式来记述事件。
- [0969] [语法]
- [0970] onDeviceData(event,keycode,ascii);
- [0971] [参数]
- [0972] • event:对象种类:(Number)
- [0973] 接收键操作的方向。图81中示出值的一览。
- [0974] • keycode:对象种类:(Number)
- [0975] 接收键码。图3A、图3B的键码一览中示出键码的例子。
- [0976] • ascii:对象种类:(String)

- [0977] 接收与所操作的键相对应的字符。
- [0978] 在不存在与键码相对应的字符的情况下 (F1键等),成为undefined。
- [0979] • onDeviceData事件 (串行通信设备)
- [0980] 该事件是从串行通信设备接收数据的事件。
- [0981] 在串行通信用的设备控制脚本502的情况下,以这种形式来记述事件。
- [0982] [语法]
- [0983] onDeviceData (data) ;
- [0984] [参数]
- [0985] • data:对象种类: (Buffer)
- [0986] 接收从串行通信设备接收到的数据。
- [0987] • 任意事件
- [0988] 其是接收在浏览器中执行动作的设备对象的API执行结果的事件。
- [0989] [语法]
- [0990] 由callEvent指定的名称 (data) ;
- [0991] [参数]
- [0992] • data:对象种类: (Object)
- [0993] 接收设备对象的由callEvent参数指定的对象。
- [0994] 对于程序进行说明。
- [0995] [概要]
- [0996] 如图82所示,程序提供利用条形码扫描器读取数据并进行印刷的功能。
- [0997] [环境]
- [0998] 设备
- [0999] • 打印机5
- [1000] • 条形码扫描器9
- [1001] 执行环境
- [1002] 在此,用下面的环境进行解说。
- [1003] 图83A中示出程序的环境构筑的流程。
- [1004] 1. 设定打印机5的网络。
- [1005] 进行系统以及打印机5的网络设定。打印机5的网络设定能够通过以下的任何一种方法来进行设定。
- [1006] • 在打印机5中,设定Windows的网络。
- [1007] • 在外部机器 (Windows的计算机) 中,安装TMNet WinConfig并进行设定。
- [1008] 2. 将条形码扫描器9与打印机5连接。
- [1009] 将条形码扫描器9与打印机5连接。
- [1010] 3. 条形码扫描器的设定。
- [1011] 将条形码扫描器9登记到打印机5中。登记从网络浏览器进行。
- [1012] [条形码扫描器的设定]
- [1013] 将通过设备XML来控制的条形码扫描器9登记到打印机5中。使用TMNet WebConfig来进行登记以及设定。

[1014] 图84A中示出TMNet WebConfig的画面例。

[1015] 1. 确认条形码扫描器9是否已被连接至打印机5,并接通打印机5的电源。

[1016] 2. 在设定用的计算机起动网络浏览器,并输入URL (http://打印机5的IP地址/webconfig/)。

[1017] 3. TMNet WebConfig被起动。选择图84A中(3)所示的[Web服务设定]-[设备管理]-[键输入设备]。

[1018] 4. “键输入设备”画面被显示。在图84A的(4)所示的输入栏中,设定图84B所示的项目,点击[登记]。

[1019] 5. 登记了的条形码扫描器被显示于图84A的(5)的[已登记键输入设备]。

[1020] 如上所述,应用了本发明的实施方式的设备控制系统100经由网络将执行应用31的终端3与打印机5连接,其中该打印机5具备用于连接设备的连接器面板60。终端3将应用31所生成的用XML记述的数据发送到打印机5。打印机5通过Socket I/F500来接收从终端3发送出的XML的数据。打印机5利用设备服务接口501以及设备控制脚本502的功能,在终端3用数据要求了管理者信息的情况下将打印机5所具有的管理者信息发送给终端3。此外,打印机5在终端3用数据要求了设备的控制的情况下对设备进行控制。因此,通过进行了网络连接的终端3,能够对与打印机5相连接的设备进行控制。此外,终端3只要是能够执行生成用XML记述的数据的应用的装置,便不受限制。因为XML通用性高,所以能够将各种各样的装置作为终端3来使用。此外,终端3的应用的开发很容易。

[1021] 此外,打印机5在终端3用包含<admin_info>消息在内的数据要求了管理者信息的情况下,将包含打印机5所具有的管理者名以及/或者位置在内的XML数据返回到终端3。

[1022] 此外,打印机5在终端3用数据要求了设备的打开的情况下,使设备成为能控制的状态,在终端3用数据要求了设备的关闭的情况下,使能控制的设备转变成不能控制的状态。

[1023] 此外,应用31所生成的数据包含连接至打印机5的各设备间各不相同的消息数据、或设备间共同的消息数据。

[1024] 此外,打印机5在对用数据要求的设备进行控制的情况下,生成向设备发送数据的DeviceConnection对象和向终端3发送数据的ClientConnection对象。

[1025] 另外,上述实施方式终究是表示本发明的一个方式,在本发明的范围内可以任意地进行变形以及应用。

[1026] 工业实用性

[1027] 如上所述,本发明对于对设备进行控制的设备控制系统非常有用,能够应用于在POS系统中使用的设备的控制或印刷装置。

[1028] 符号说明

[1029] 2 应用服务器

[1030] 3 终端

[1031] 5 打印机

[1032] 7 网络打印机

[1033] 8 顾客显示器

[1034] 9 条形码扫描器

- [1035] 11 无线LAN接入点
- [1036] 12 显示器
- [1037] 13 现金抽屉
- [1038] 14 键盘
- [1039] 15 USB延长线
- [1040] 17 无线LAN单元
- [1041] 18 键输入设备
- [1042] 19 串行通信设备
- [1043] 20 设备
- [1044] 31 应用
- [1045] 51 卷筒纸盖
- [1046] 52 盖打开按钮
- [1047] 53 手动剪切器
- [1048] 54 剪切器盖
- [1049] 55 本地打印机
- [1050] 56A 复位按钮
- [1051] 56B 电源开关
- [1052] 57 LED显示部
- [1053] 58 控制面板
- [1054] 59 连接器盖
- [1055] 60 连接器面板
- [1056] 61 抽屉弹出连接器
- [1057] 62 以太网连接器
- [1058] 63 USB连接器
- [1059] 64 VGA连接器
- [1060] 65 COM连接器
- [1061] 100 设备控制系统
- [1062] 500 Socket I/F
- [1063] 501 设备服务接口
- [1064] 502 设备控制脚本
- [1065] 503 设备连接
- [1066] 504 客户端连接

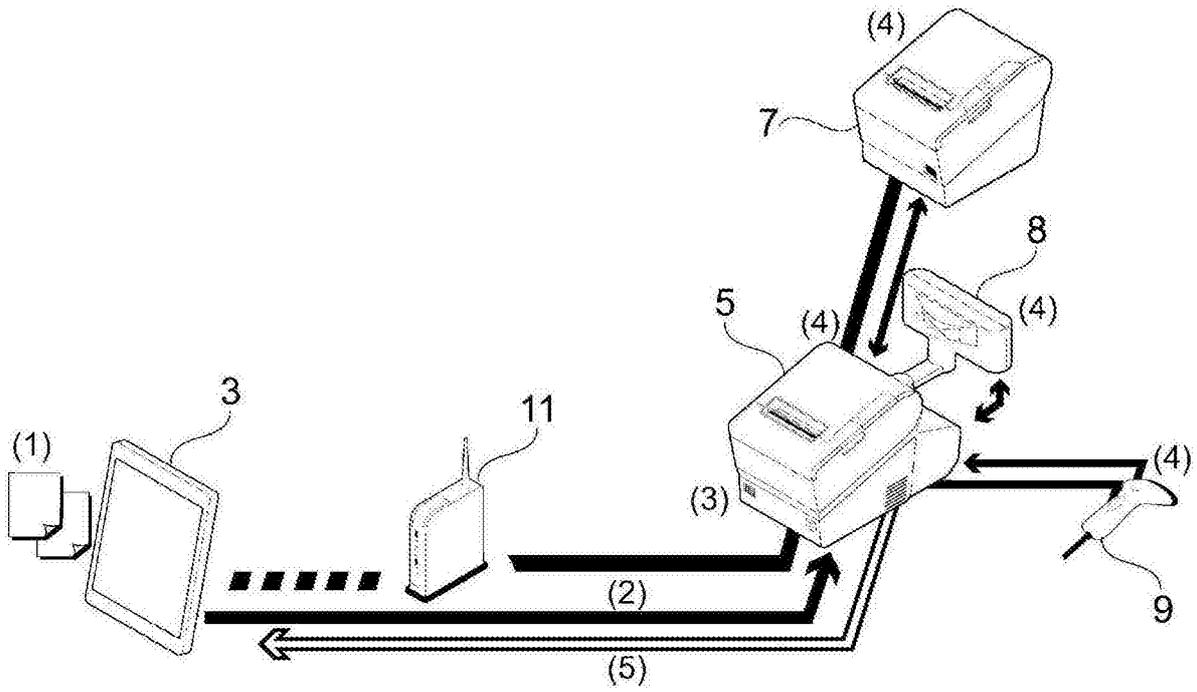


图1

100

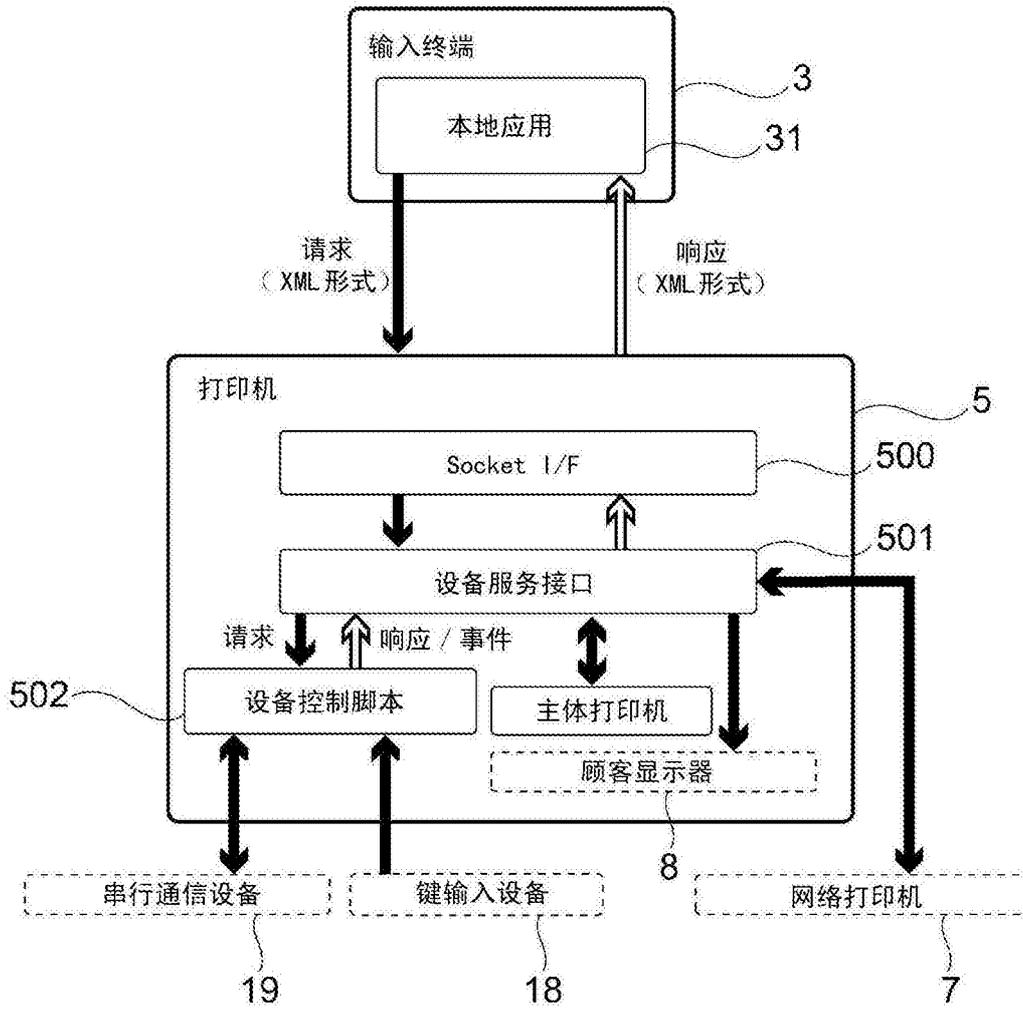


图2

键	键码 (10进制数)	键	键码 (10进制数)
Backspace	8	键盘上的数字	
Tab	9	5	53
Enter	13	6	54
Shift	16	7	55
Ctrl	17	8	56
Alt	18	9	57
Esc	27	键盘上的字母	
空格	32	A	65
PgUp	33	B	66
PgDn	34	C	67
End	35	D	68
Home	36	E	69
!	37	F	70
#	38	G	71
i	39	H	72
\$	40	I	73
Insert	45	J	74
Delete	46	K	75
键盘上的数字		L	76
0	48	M	77
1	49	N	78
2	50	O	79
3	51	P	80
4	52	Q	81

图3A

键	键码 (10进制数)	键	键码 (10进制数)
键盘上的字母		功能键	
R	82	F8	119
S	83	F9	120
T	84	F10	121
U	85	F11	122
V	86	F12	123
W	87	:	186
X	88	;	187
Y	89	'	188
Z	90	=	189
数字键		.	190
*	106	/	191
+	107	@	192
/	109	[219
功能键		\	220
F1	112]	221
F2	113	^	222
F3	114	Caps Lock	240
F4	115		
F5	116		
F6	117		
F7	118		

图3B

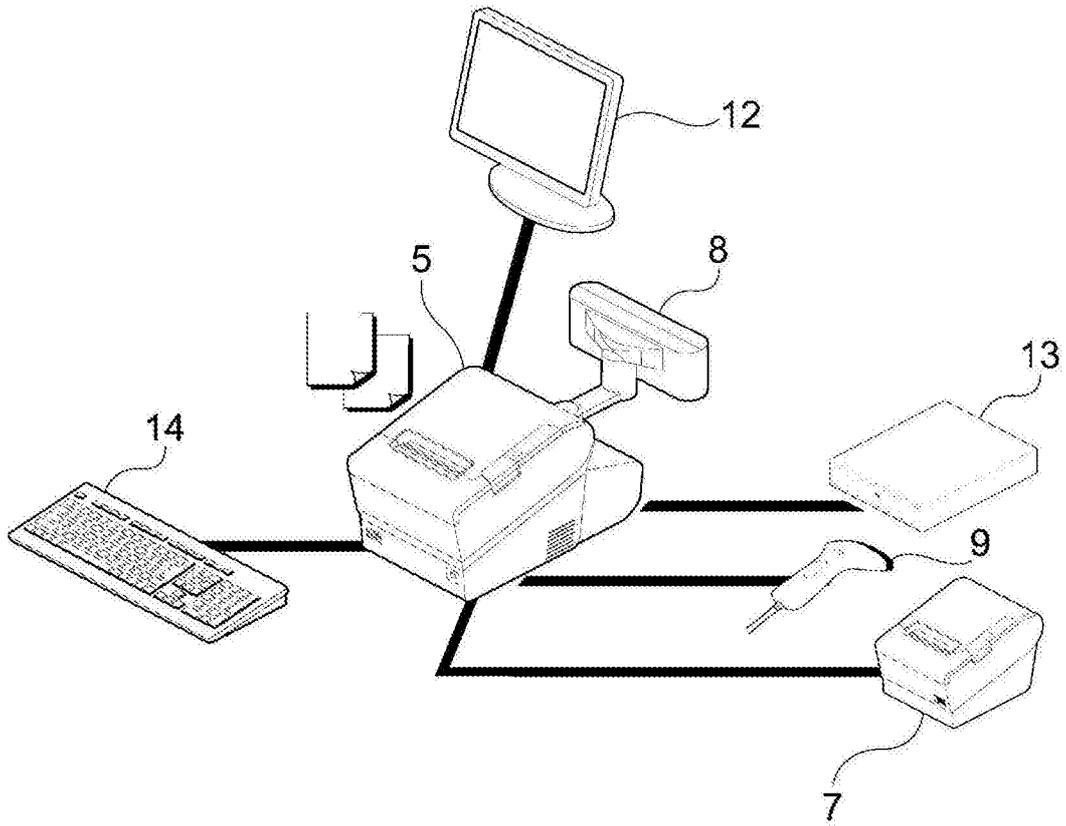


图4

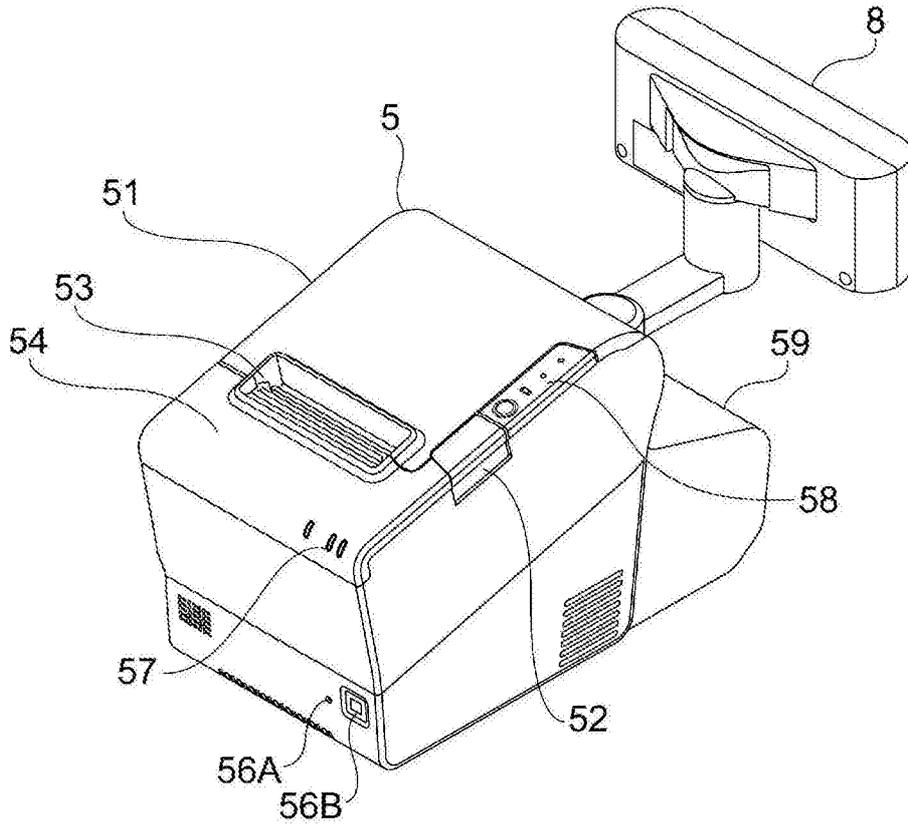


图5

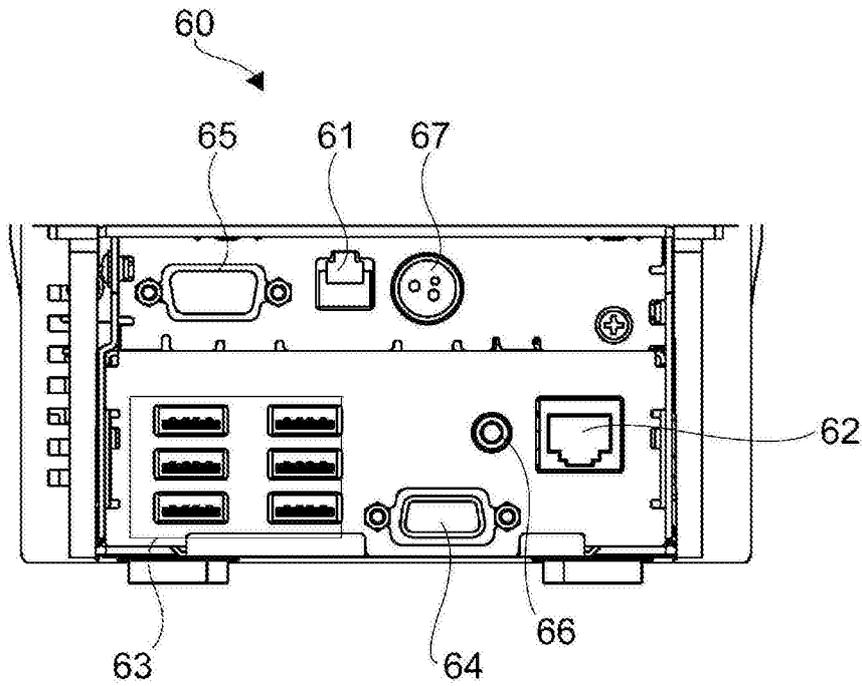


图6

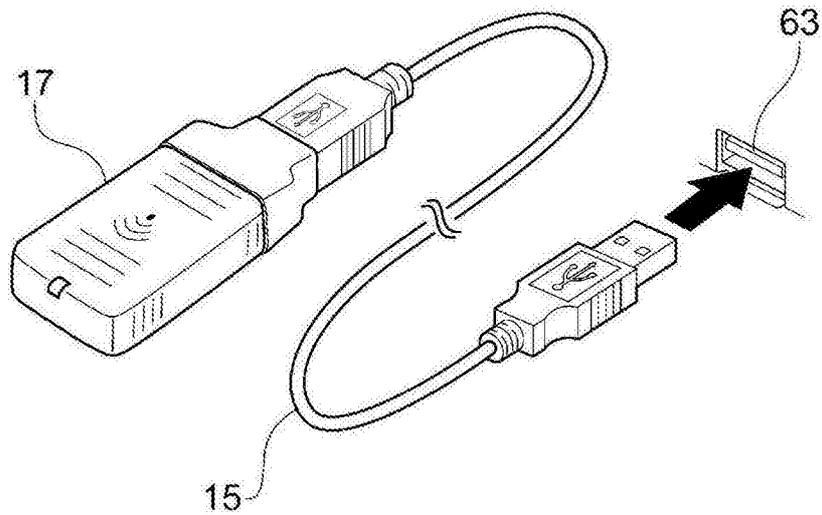


图7

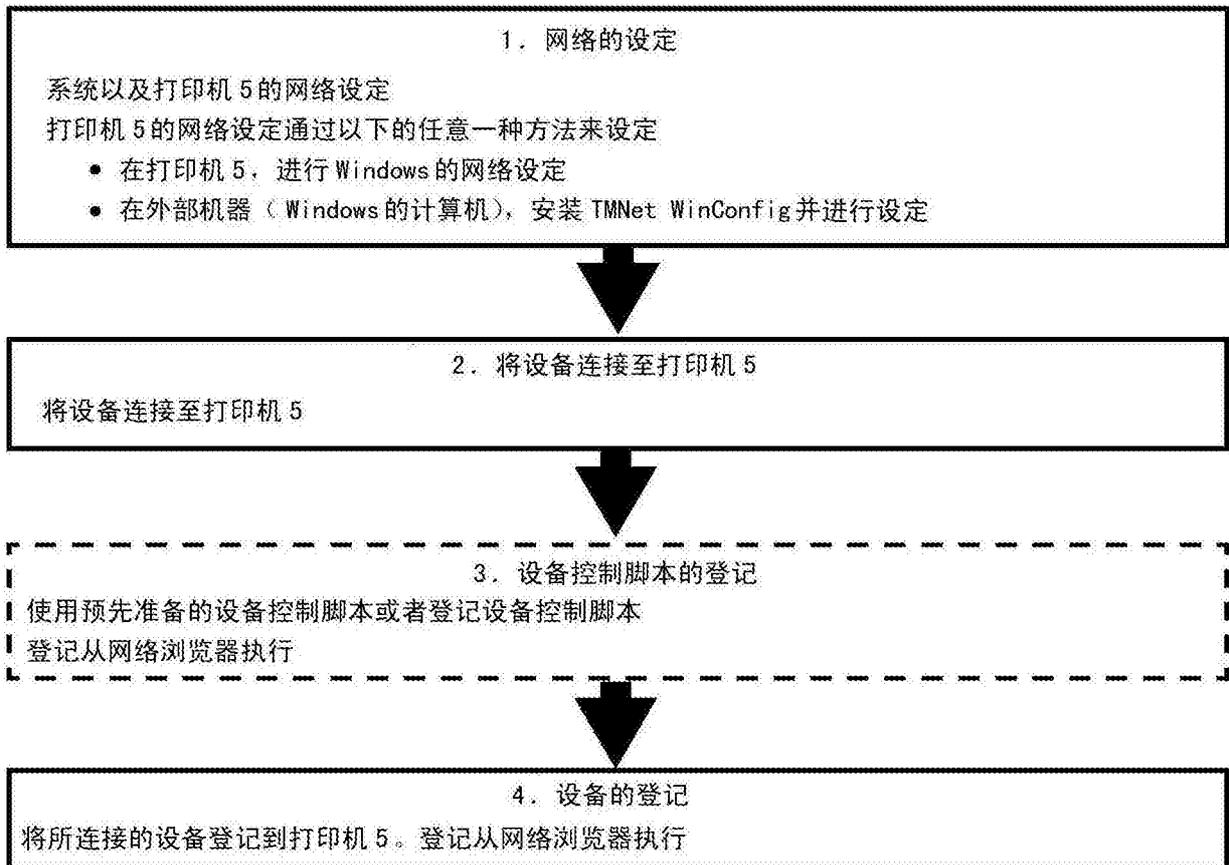


图8



图9



图10

项目	说明
打印机	由打印机 5 控制的打印机的设定
显示器	由打印机 5 控制的顾客显示器的设定
键输入设备	由打印机 5 控制的键输入设备的设定
串行通信	由打印机 5 控制的串行通信设备的设定
其他	上述以外的由打印机 5 控制的设备的设定

图11

项目	说明
设备 ID	输入所控制的打印机的 ID (任意的字符串)
类型	选择“网络打印机”
型号	选择所控制的打印机的型号
IP 地址	设定每个设备 ID 的打印机的 IP 地址
重试间隔	设定超时的重试间隔

图12

项目	说明
通信设定	设定通信速度、数据比特、奇偶
亮度	设定顾客显示器的亮度

图13

项目	说明
设备 ID	输入设备 ID (任意的字符串)
设备名	选择键输入设备的设备名
控制脚本	选择在键输入设备使用的设备控制脚本

图14

项目	说明
设备 ID	输入设备 ID (任意的字符串)
设备名	选择串行通信设备的设备名 从产品名、端口中进行选择
控制脚本	选择在串行通信设备使用的设备控制脚本
通信速度	设定设备的通信速度
数据比特	设定数据比特
奇偶	设定奇偶
停止位	设定停止位
流量控制	设定流量控制

图15

项目	说明
设备 ID	输入设备 ID (任意的字符串)
控制脚本	选择在所连接的设备使用的设备控制脚本

图16

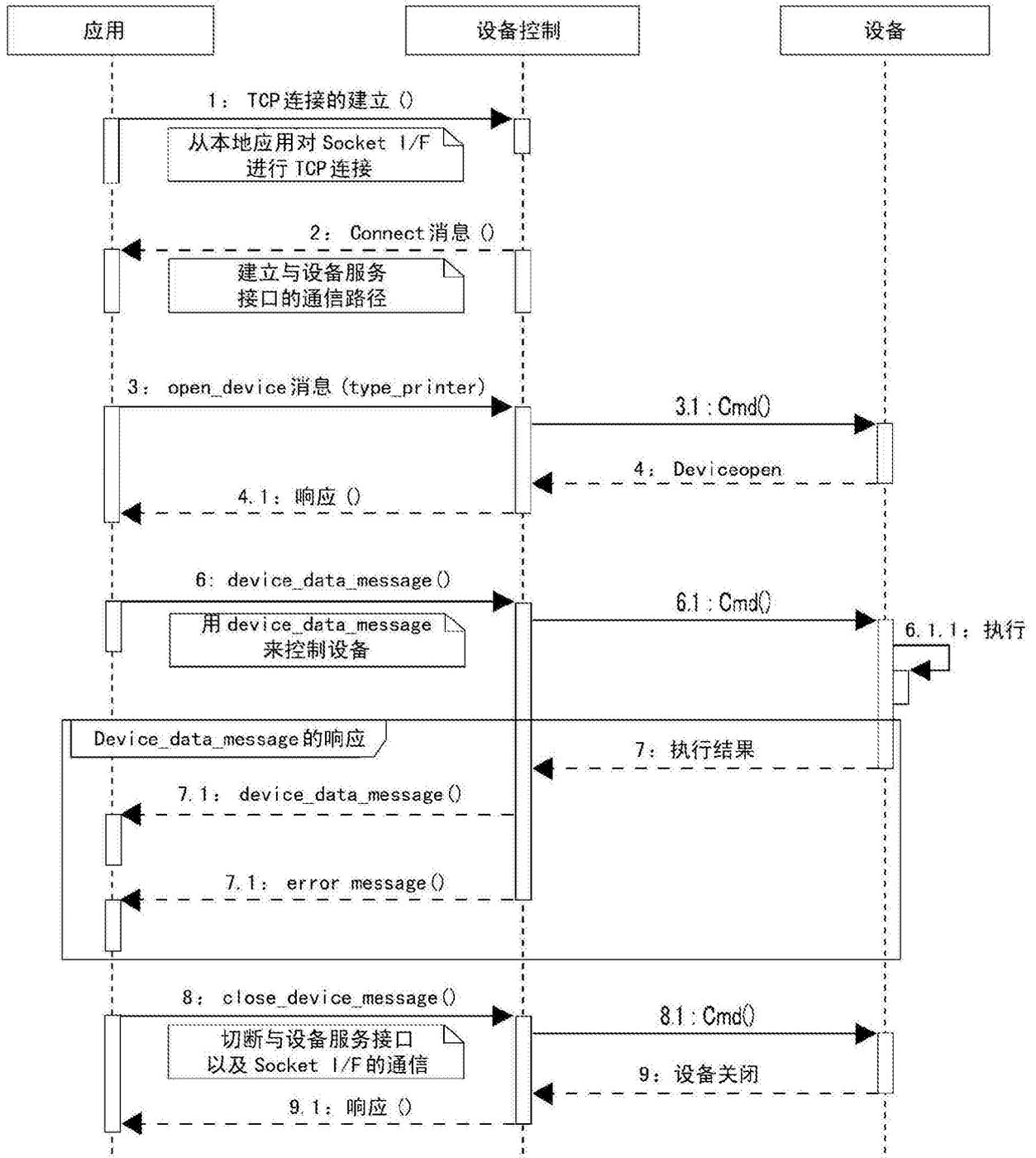


图17

传输层	端口号
TCP	8009

图18

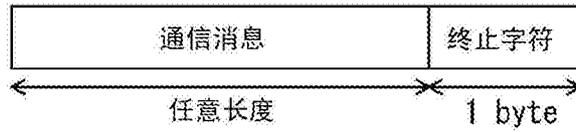


图19A

数据名	内容
通信消息	「UTF-8」的XML形式的字符串 任意长度
终止字符	NULL字符("¥0") 1 byte

图19B

应用	方向	设备	说明
TCP/IP连接的建立	➡		连接的建立
	⬅	<connect> </connect>"\0"	
<admin_info></admin_info>"\0"	➡		管理者信息的取得
	⬅	<admin_info> <code>OK</code> <data> <admin_name>ep-admin</admin_name> <location>Counter01</location> </data> </admin_info>"\0"	
<open_device> <device_id>keyboard01</device_id> <data> <type>type_keyboard</type> <buffer>>true</buffer> </data> </open_device>"\0"	➡		设备的打开
	⬅	<open_device> <device_id>keyboard01</device_id> <code>OK</code> <data_id>1</data_id>	
<device_data> <device_id>keyboard01</device_id> <data> <type>setprefix</type> <keycode>49</keycode> <keycode>50</keycode> <keycode>51</keycode> </data> </device_data>"\0"	➡		设备控制
	⬅	<device_data> <device_id>keyboard01</device_id> <data> <type>onkeypress</type> <keycode>49</keycode> <ascii>a</ascii> </data> <data_id>2</data_id> </device_data>"\0" <device_data> <device_id>keyboard01</device_id> <data> <type>onstring</type> <input>11223344556677</input> <prefix>a</prefix> </data> </device_data>"\0"	设备控制

图20A

应用	方向	设备	说明
<code><close_device> <device_id>keyboard01</device_id> </close_device>"\0"</code>	➔		设备的关闭
	➜	<code><close_device> <device_id>keyboard01</device_id> <code>OR</code> </close_device>"\0"</code>	
连接的切断			

图20B

```

<data>
  <type>setprefix</type>
  <keycodes array="true">49</keycodes>
  <keycodes>50</keycodes>
  <keycodes>51</keycodes>
  <keycodes>52</keycodes>
</data>
    
```

图21

消息		说明
通信路径	<code><connect></code>	建立与 ePOS- 设备服务接口的通信路径
管理信息	<code><admin_info></code>	管理者、设置场所信息的取得
设备的 打开 / 关闭	<code><open_device></code>	打开与设备的通信
	<code><close_device></code>	关闭与设备的通信
设备控制 数据的发送	<code><device_data></code>	请求设备控制
	<code><device_data></code>	设备控制的响应
错误通知	<code><error></code>	错误发生时的错误内容的通知

图22

设备的种类	<type>要素的值	说明
条形码扫描器	ondata	来自条形码扫描器的扫描数据 <响应>
打印机	print	给打印机的印刷数据与设定数据 <请求>
	onxmresult	来自打印机的数据 <响应>
顾客显示器	display	给顾客显示器的显示数据与设定数据 <请求>
	onxmresult	来自顾客显示器的数据 <响应>
POS 键盘	onkeypress	来自 POS 键盘的输入字符串 <响应>
	setprefix	为了检测来自 POS 键盘的输入，指定判断为字符串的开始的键码 <请求>
	onstring	来自 POS 键盘的输入字符串 <响应>
串行设备	sendcommand	向串行设备的指令的发送 <请求>
	oncommandreply	来自串行设备的数据 <响应>

图23

子要素	数据类型	说明
code	string	管理者信息的取得结果 详细参照下述 code
data	string	管理者信息 在取得成功的情况下回复 详细参照 data

图24A

要素的值	说明
"OK"	取得成功
"PARAM_ERROR"	参数错误发生
"SYSTEM_ERROR"	系统错误发生

图24B

子要素	数据类型	说明
admin_name	string	管理者信息
location	string	设置场所信息

图24C

替换前字符串	替换后
&(与号)	&
>(大于号)	>
<(小于号)	<
"(双引号)	"
'(单引号)	'

图24D

正常时

```
<admin_info>
  <code>OK</code>
  <data>
    <admin_name>管理者A</admin_name>
    <location> 柜台 1&amp; 柜台 2</location>
  </data>
</admin_info>
```

异常时

```
<admin_info>
  <code>PARAM_ERROR</code>
</admin_info>
```

图24E

子要素	数据类型	说明
device_id	string	通过 TMNet WebConfig 指定完成的设备 ID 顾客显示器为“local_display”
data	string	详细参照 data

图25A

要素的值	说明
"type_scanner"	条形码扫描器
"type_keyboard"	POS 键盘
"type_printer"	打印机
"type_display"	顾客显示器
"type_simple_serial"	串行通信设备

图25B

```

<open_device>
  <device_id>scanner_001</device_id>
  <data>
    <type>type_scanner</type>
  </data>
</open_device>

```

图25C

子要素	数据类型	说明
device_id	string	在请求中指定的设备 ID
code	string	处理结果 详细参照 code

图25D

要素的值	说明
"OK"	处理成功
"DEVICE_NOT_FOUND"	所指定的设备不存在
"DEVICE_IN_USE"	所指定的设备处于使用中
"DEVICE_OPEN_ERROR"	所指定的设备的打开失败
"DRIVER_ERROR"	在设备控制脚本发生错误
"SYSTEM_ERROR"	系统错误发生
"DEVICE_CATEGORY_INVALID"	所指定的设备的种类不正
"COMMAND_PARAM_ILLEGAL"	参数错误发生

图25E

正常时

```
<open_device>
  <code>OK</code>
  <data_id>scanner_001</device_id>
</open_device>
```

异常时

```
<open_device>
  <code>DEVICE_NOT_FOUND</code>
  <device_id>scanner_001</device_id>
</open_device>
```

图25F

子要素	数据类型	说明
device_id	string	通过 TMNet WebConfig 指定完成的设备 ID 顾客显示器为 "local_display"

图26A

```
<close_device>
  <device_id>local_display</device_id>
</close_device>
```

图26B

子要素	数据类型	说明
device_id	string	在请求中指定的设备 ID
code	string	处理结果 详细参照 code

图26C

要素的值	说明
"OK"	处理成功
"DEVICE_NOT_FOUND"	所指定的设备不存在
"DEVICE_NOT_OPEN"	所指定的设备 ID 未被打开
"DEVICE_CLOSE_ERROR"	所指定的设备的关闭失败
"SYSTEM_ERROR"	系统错误发生
"COMMAND_PARAM_ILLEGAL"	参数错误发生

图26D

正常时

```

<close_device>
  <device_id>local_display</device_id>
  <code>OK</code>
</close_device>

```

异常时

```

<close_device>
  <device_id>scanner_001</device_id>
  <code>DEVICE_NOT_FOUND</code>
</close_device>

```

图26E

子要素	数据类型	说明
sequence	string	在发送错误时，用于确定请求消息的任意的数值 若在请求消息中进行指定，则在错误发生时设定到 error message 的 sequence 中
device_id	string	已经打开的设备 ID
data	string	设备控制的数据 内容按照每个设备的种类来进行定义

图27A

```

<device_data>
  <sequence>123</sequence>
  <device_id>poskeyboard001</device_id>
  <data>
    <type>setprefix</type>
    <keycodes array="true">49</keycodes>
    <keycodes>50</keycodes>
    <keycodes>51</keycodes>
    <keycodes>52</keycodes>
  </data>
</device_data>

```

图27B

子要素	数据类型	说明
sequence	string	0 固定
device_id	string	产生了数据的设备 ID
data	string	产生的数据 内容按照每个设备的种类来进行定义

图27C

```

<device_data>
  <sequence>0</sequence>
  <device_id>poskeyboard001</device_id>
  <data>
    <type>onstring</type>
    <input>&#2398749238429</input>
    <prefix>49</prefix>
  </data>
</device_data>

```

图27D

子要素	数据类型	说明
sequence	string	用于确定成为错误原因的消息的 ID 与在请求中指定的 sequence 相同的值
device_id	string	发生了错误的设备 ID 在设备控制消息中的错误的情况下指定
code	string	用代码来表示所发生的错误
data	string	DRIVER_ERROR 发生时，指定与驱动器相应的错误的详情

图28A

● device_data 中的错误时

要素值	说明
"DEVICE_NOT_FOUND"	所指定的设备 ID 不存在
"DEVICE_NOT_OPEN"	所指定的设备没有打开
"SEND_ERROR"	在向符合的物理设备的数据发送中发生错误
"DRIVER_ERROR"	在驱动器检测出错误
"PARAM_ERROR"	参数错误发生
"FUNCTION_NOT_FOUND"	所指定的函数不存在于驱动器中

● 其他错误时

要素值	说明
"SYSTEM_ERROR"	系统错误发生

图28B

```

<error>
  <sequence>123</sequence>
  <device_id>scanner002</device_id>
  <code>DEVICE_NOT_OPEN</code>
  <data></data>
</error>
    
```

图28C

子要素	数据类型	说明
input	string	所接收到的扫描数据的字符串

图29A

替换前字符串	替换后
& (与号)	&
> (大于号)	>
< (小于号)	<
" (双引号)	"
' (单引号)	'

图29B

```

<data>
  <type>ondata</type>
  <input>F12345678909123</input>
</data>
    
```

图29C

子要素	数据类型	说明
timeout	int	以毫秒指定请求的超时时间 1000~60000 (整数)
printdata	string	指定印刷的数据

图30A

```

<data>
  <type>print</type>
  <timeout>10000</timeout>
  <printdata>
    <!-- 打印机控制用 XML -->
    <epos-print xmlns=
      .
      .
    </epos-print>
  </printdata>
</data>
    
```

图30B

子要素	数据类型	说明
resultdata	string	所发送的数据的结果

图31A

```

<data>
  <type>onxmlresult</type>
  <resultdata>
    <response success=    />
  </resultdata>
</data>

```

图31B

子要素	功能	打印机的印刷模式	
		标准模式	页面模式
<text>	字符的打印	○	○
<feed>	送纸	○	○
<image>	光栅图像的打印	○	○
<logo>	NV标识的打印	○	○
<barcode>	条形码的打印	○	○
<symbol>	二维码的打印	○	○
<hline>	横格线的打印	○	△
<vline-begin>	纵格线的开始	○	△
<vline-end>	纵格线的结束	○	△
<page>	页面模式	○	△
<area>	页面模式打印区域的设定	△	○
<direction>	页面模式打印方向的设定	△	○
<position>	页面模式打印位置的设定	△	○
<line>	页面模式直线的描绘	△	○
<rectangle>	页面模式长方形的描绘	△	○
<cut>	纸张剪切	○	△
<pulse>	抽屉弹出	○	△
<sound>	蜂鸣器的鸣动	○	△
<command>	指令的插入	○	○

图32A

```

<!--作为子要素而追加了 text 标签的例子 -->
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <text>Hello world!!</text>
</epos-print>

```

图32B

属性值	说明
"true" 或者 "1"	表示成功
"false" 或者 "0"	表示失败

图33A

属性值	说明
"EPTR_AUTOMATIC"	自动恢复错误发生
"EPTR_COVER_OPEN"	盖打开错误发生
"EPTR_CUTTER"	自动剪切器错误发生
"EPTR_MECHANICAL"	机械错误发生
"EPTR_REC_EMPTY"	卷筒纸用尽检测器中无纸
"EPTR_UNRECOVERABLE"	无法恢复错误发生
"SchemaError"	请求文档的语法中有错误
"DeviceNotFound"	由设备 ID 指定的打印机不存在
"PrintSystemError"	印刷系统中错误发生
"EX_BADPORT"	在通信端口检测出异常
"EX_TIMEOUT"	印刷超时发生

图33B

属性值	说明
"0x00000001"	TM打印机无响应
"0x00000002"	印刷结束
"0x00000004"	抽屉弹出连接器第3管脚的状态 = "H"
"0x00000008"	离线状态
"0x00000020"	盖正打开着
"0x00000040"	通过送纸开关进行的送纸中
"0x00000100"	在线恢复等待中
"0x00000200"	送纸开关按下中
"0x00000400"	机械错误发生
"0x00000800"	自动剪切器错误发生
"0x00002000"	无法恢复错误发生
"0x00004000"	自动恢复错误发生
"0x00020000"	卷筒纸即将用尽检测器中无纸
"0x00080000"	卷筒纸用尽检测器中无纸
"0x01000000"	蜂鸣器鸣动中 (仅限支持机器)
"0x80000000"	打印程序的停止 (未使用)

图33C

```

<!-- Print 的打印机的状态      -->
<!-- 成功时 -->
  <response success="true" code="" status="" />
<!-- 打印中纸张用尽、印刷失败的情况 -->
<response xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print"
  success="false" code="EPTR_REC_EMPTY" status="252641308" />
    
```

图33D

功能 (符号)	实体参照
&	&
'	'
>	>
<	<
"	"
水平跳格 (HT)		 或者 	
换行 (LF)	
 或者

图34A

属性值	说明
"en" (初始值)	英语
"ja"	日语

图34B

属性值	说明
"font_a" (初始值)	字体 A
"font_b"	字体 B
"font_c"	字体 C

图34C

属性值	说明
"true" 或者 "1"	指定平滑
"false" 或者 "0" (初始值)	解除平滑

图34D

属性值	说明
"true" 或者 "1"	指定横倍角
"false" 或者 "0" (初始值)	解除横倍角

图34E

属性值	说明
"true" 或者 "1"	指定纵倍角
"false" 或者 "0" (初始值)	解除纵倍角

图34F

属性值	说明
"1" ~ "8" (初始值: "1")	横向倍率

图34G

属性值	说明
"1" ~ "8" (初始值: "1")	纵向倍率

图34H

属性值	说明
"true" 或者 "1"	指定黑白反转字符
"false" 或者 "0" (初始值)	解除黑白反转字符

图34I

属性值	说明
"true" 或者 "1"	指定下划线
"false" 或者 "0" (初始值)	解除下划线

图34J

属性值	说明
"true" 或者 "1"	指定强调打印
"false" 或者 "0" (初始值)	解除强调打印

图34K

属性值	说明
"none"	不打印
"color_1" (初始值)	第 1 颜色
"color_2"	第 2 颜色
"color_3"	第 3 颜色
"color_4"	第 4 颜色

图34L

属性值	说明
"left" (初始值)	左对齐
"center"	中央对齐
"right"	右对齐

图34M

属性值	说明
"true"或者"1"	指定倒立打印
"false"或者"0" (初始值)	解除倒立打印

图34N

项目	设定值
语言	日语
平滑	有效
位置对齐	中央对齐
字体	字体 A
倍角	倍角
下划线	有效

```

<text lang="ja" />
<text smooth="true" />
<text align="center" />
<text font="font_a" />
<text dw="true" dh="true" />
<text ul="true" />
<text>Hello, World!&#10;</text>
    
```

图34P

```

<!-- 进行3行送纸 -->
<feed line="3" />
    
```

```

<!-- 打印字符串并换行 -->
<text>Hello</text><feed />
<text>World</text><feed />
    
```

图35

属性值	说明
"none"	非打印
"color_1" (初始值)	第 1 颜色
"color_2"	第 2 颜色
"color_3"	第 3 颜色
"color_4"	第 4 颜色

图36A

属性值	说明
"left" (初始值)	左对齐
"center"	中央对齐
"right"	右对齐

图36B

属性值	说明
"mono" (初始值)	单色 (2级灰度)
"gray16"	多级灰度 (16级灰度)

图36C

```

<!-- 打印以 8 点宽、8 点高填充的光栅图像 -->
<image width="8" height="8">//////////8=</image>
    
```

图36D

属性值	说明
"left" (初始值)	左对齐
"center"	中央对齐
"right"	右对齐

图37A

```

<!-- 对键码 1 被登记为 48、键码 2 被登记为 48 的 NV 标识进行打印 -->
<logo key1="48" key2="48" />
    
```

图37B

种类	条形码数据的指定方法
UPC-A	在指定了 11 位数字的情况下，附加校验位 在指定了 12 位数字的情况下，将第 12 位作为校验位处理
UPC-E	在第 1 位中指定 0 在第 2~6 位中指定制造商代码 在第 7~11 位中以向右对齐的方式指定项目代码 在指定了 11 位数字的情况下，附加校验位 在指定了 12 位数字的情况下，将第 12 位作为校验位处理
EAN13	在指定了 12 位数字的情况下，附加校验位
JAN13	在指定了 13 位数字的情况下，将第 13 位作为校验位处理
EAN8	在指定了 7 位数字的情况下，附加校验位
JAN8	在指定了 8 位数字的情况下，将第 8 位作为校验位处理
CODE39	在开头的字符为 * 的情况下，将该字符作为起始字符处理 除此以外的情况下，附加起始字符
ITF	附加起始代码以及终止代码
CODABAR	指定起始字符 (A~D、a~d) 指定终止字符 (A~D、a~d)

图38A

种类	条形码数据的指定方法
CODE93	附加起始字符以及终止字符
CODE128	指定起始字符 (CODE A, CODE B, CODE C) 附加终止字符 计算并附加校验位 为了对以下的字符进行编码, 指定以字符 [开始的 2 个字符 FNC1 : {1 FNC2 : {2 FNC3 : {3 FNC4 : {4 CODE A : {A CODE B : {B CODE C : {C SHIFT : {S { : {{
GS1-128	附加起始字符、FNC1、校验位、终止字符 为了计算并附加应用标识符 (AI) 以及接在其后的数据的校验位, 在校验位的位置指定字符 * 能够用括号将应用标识符 (AI) 括起来 括号作为 HRI 的打印字符使用 能够在应用标识符 (AI) 和数据之间插入空白 空白作为 HRI 的打印字符使用 为了对以下的字符进行编码, 指定以字符 [开始的 2 个字符 字符指定 字符 FNC1 {1 FNC3 {3 { {{) {} * {* { {{
GS1 DataBar Omnidirectional,	指定应用标识符 (AI) 和除了校验位以外的 13 位的商品识别编号 (GTIN)
GS1 DataBar Truncated	
GS1 DataBar Limited	

图38B

种类	条形码数据的指定方法
BARCODE_GS1_DataBar Expanded	能够用括号将应用标识符 (AI) 括起来 括号作为 HRI 的打印字符使用 为了对以下的字符进行编码, 指定以字符 [开始的 2 个字符 FNC1 : {1 { {{) {}

图38C

字符串	说明
\xnn	控制代码
\\	反斜线

图38D

属性值	说明
"upc_a"	UPC-A
"upc_e"	UPC-E
"ean13"	EAN13
"jan13"	JAN13
"ean8"	EAN8
"jan8"	JAN8
"code39"	CODE39
"itf"	ITF
"codabar"	CODABAR
"code93"	CODE93
"code128"	CODE128
"gs1_128"	GS1-128
"gs1_databar_omnidirectional"	GS1 DataBar Omnidirectional
"gs1_databar_truncated"	GS1 DataBar Truncated
"gs1_databar_limited"	GS1 DataBar Limited
"gs1_databar_expanded"	GS1 Databar Expanded

图38E

属性值	说明
"none" (初始值)	不打印
"above"	条形码之上
"below"	条形码之下
"both"	条形码之上和下两方

图38F

属性值	说明
"font_a" (初始值)	字体 A
"font_b"	字体 B
"font_c"	字体 C

图38G

属性值	说明
"left" (初始值)	左对齐
"center"	中央对齐
"right"	右对齐

图38H

属性值	说明
"true" 或者 "1"	指定倒立打印
"false" 或者 "0" (初始值)	解除倒立打印

图38I

```

<!-- 指定各种条形码 -->
<barcode type="upc_a" width="2" height="64" hri="below">01234567890</barcode>
<barcode type="upc_e">01234500005</barcode>
<barcode type="ean13">201234567890</barcode>
<barcode type="jan13">201234567890</barcode>
<barcode type="ean8">2012345</barcode>
<barcode type="jan8">2012345</barcode>
<barcode type="code39">ABCDE</barcode>
<barcode type="itf">012345</barcode>
<barcode type="codabar">A012345A</barcode>
<barcode type="code93">ABCDE</barcode>
<barcode type="code128">{Babcde</barcode>
<barcode type="gs1_128">(01)201234567890*</barcode>
<barcode type="gs1_databar_omnidirectional">0201234567890</barcode>
<barcode type="gs1_databar_truncated">0201234567890</barcode>
<barcode type="gs1_databar_limited">0201234567890</barcode>
<barcode type="gs1_databar_expanded">(01)2012345678903</barcode>
    
```

图38J

种类	说明
PDF417	将字符串转换成 UTF-8 后，进行转义序列的处理，编码数据区域的最大代码字数为 928 个，每一级的数据区域的最大代码字数为 30 个，最大级数为 90 级
QR Code	将字符串转换成 Shift-JIS 后，进行转义序列的处理，从以下中选择数据的种类来进行编码 数字数据 : 0 ~ 9 字母数字数据 : 0 ~ 9, A ~ Z, スペース, \$, %, *, +, -, ., /, : 汉字数据 : Shift-JIS 值 8 位字节数据 : 0x00 ~ 0xff

图39A

种类	说明
MaxiCode	<p>将字符串转换成 UTF-8 后，进行转义序列的处理，进行编码</p> <p>在编码模式 2 以及模式 3 的情况下，在最开始的数据为 []>\x1e01\x1dyx (yy 是 2 位数字) 的情况下，将其作为消息头进行处理，从后面的数据起作为主要消息来处理</p> <p>上述以外的情况下，从最初的数据起作为主要消息来处理</p> <p>通过以下的形式来指定主要消息</p> <ul style="list-style-type: none"> ● 模式 2 的情况 <p>邮政编码: (1 ~ 9 位数字)</p> <p>GS: (\x1d)</p> <p>ISO 国名代码: (1 ~ 3 位数字)</p> <p>GS: (\x1d)</p> <p>服务类代码 (service class code)</p> <p>(1 ~ 3 位数字)</p> <ul style="list-style-type: none"> ● 模式 3 的情况 <p>邮政编码: (能够用 1 ~ 6 个代码集 A 变换的数据)</p> <p>GS: (\x1d)</p> <p>ISO 国名代码: (1 ~ 3 位数字)</p> <p>GS: (\x1d)</p> <p>服务类代码</p> <p>(1 ~ 3 位数字)</p>
GS1 DataBar Stacked	将字符串转换成 UTF-8 后，进行转义序列的处理，进行编码
GS1 DataBar Stacked Omnidirectional	指定应用标识符 (AI) 和除了校验位以外的 13 位的商品识别编号 (GTIN)
GS1 DataBar Expanded Stacked	<p>将字符串转换成 UTF-8 后，进行转义序列的处理，进行编码</p> <p>能够用括号将应用标识符 (AI) 括起来</p> <p>括号作为 HRI 的打印字符使用</p> <p>为了对以下的字符进行编码，指定以字符 { 开始的 2 个字符</p> <p>FNC1 : {</p> <p>(: {{</p> <p>) : {}</p>

图 39B

字符串	说明
\xnn	控制代码
//	反斜线

图 39C

属性值	种类
"pdf417_standard"	Standard PDF417
"pdf417_truncated"	Truncated PDF417
"qrcode_model_1"	QR Code Model 1
"qrcode_model_2"	QR Code Model 2
"maxicode_mode_2"	MaxiCode Mode 2
"maxicode_mode_3"	MaxiCode Mode 3
"maxicode_mode_4"	MaxiCode Mode 4
"maxicode_mode_5"	MaxiCode Mode 5
"maxicode_mode_6"	MaxiCode Mode 6
"gs1_databar_stacked"	GS1 DataBar Stacked
"gs1_databar_stacked_omnidirectional"	GS1 DataBar Stacked Omnidirectional
"gs1_databar_expanded_stacked"	GS1 DataBar Expanded Stacked

图39D

属性值	说明	
"level_0"	PDF417 纠错等级	0
"level_1" (初始值)	PDF417 纠错等级	1
"level_2"	PDF417 纠错等级	2
"level_3"	PDF417 纠错等级	3
"level_4"	PDF417 纠错等级	4
"level_5"	PDF417 纠错等级	5
"level_6"	PDF417 纠错等级	6
"level_7"	PDF417 纠错等级	7
"level_8"	PDF417 纠错等级	8
"level_l"	QR Code 纠错等级	L
"level_m"	QR Code 纠错等级	M
"level_q"	QR Code 纠错等级	Q
"level_h"	QR Code 纠错等级	H
"default"	保持默认的设置	

图39E

二维码	有效值的范围	初始值
PDF417	"2" ~ "8"	width="3"
QR Code	"1" ~ "16"	width="3"
MaxiCode	无视	
GS1 DataBar	"2" ~ "8"	width="2"

图39F

二维码的种类	有效值的范围	初始值
PDF417	"2" ~ "8"	height="3"
QR Code	无视	
MaxiCode		
二维 GS1 Databar		

图39G

二维码的种类	初始值	说明
PDF417	size="0"	指定每一级的代码字数
QR Code	无视	
MaxiCode		
GS1 Databar	根据机型而不同	指定条形码的最大宽度

图39H

属性值	说明
"left" (初始值)	左对齐
"center"	中央对齐
"right"	右对齐

图39I

属性值	说明
"true" 或者 "1"	指定倒立打印
"false" 或者 "0" (初始值)	解除倒立打印

图39J

```

<!-- 指定各种二维码 -->
<symbol type="pdf417_standard">ABCDE</symbol>
<symbol type="qrcode_model_2" level="level_q">ABCDE</symbol>
<symbol type="maxicode_mode_2">908063840\x1d850\x1d001\x1d\x04</symbol>
<symbol type="gs1_databar_stacked">0201234567890</symbol>
<symbol type="gs1_databar_stacked_omnidirectional">0201234567890</symbol>
<symbol type="gs1_databar_expanded_stacked">(01)02012345678903</symbol>
    
```

图39K

属性值	说明
"thin"	实线：细
"medium"	实线：中粗
"thick"	实线：粗
"thin_double"	双划线：细
"medium_double"	双划线：中粗
"thick_double"	双划线：粗

图40A

<!-- 以双划线从左端 100 点起描绘 200 点、从左端 400 点起描绘 500 点的横格线 -->
 <hline x1="100" x2="200" style="thin_double" />
 <hline x1="400" x2="500" style="thin_double" />

图40B

属性值	说明
"thin"	实线：细
"medium"	实线：中粗
"thick"	实线：粗
"thin_double"	双划线：细
"medium_double"	双划线：中粗
"thick_double"	双划线：粗

图41A

<!-- 以细线从左端 100 点和 200 点的位置开始描绘 -->
 <vline-begin x="100" />
 <vline-begin x="200" />
 <feed unit="100" />
 <vline-end x="100" />
 <vline-end x="200" />

图41B

属性值	说明
"thin"	实线：细
"medium"	实线：中粗
"thick"	实线：粗
"thin_double"	双划线：细
"medium_double"	双划线：中粗
"thick_double"	双划线：粗

图42A

```

<!-- 以细线描绘到左端 100点和 200点的位置 -->
<vline-begin x="100" />
<vline-begin x="200" />
<feed unit="100" />
<vline-end x="100" />
<vline-end x="200" />

```

图42B

要素	说明
<text>	字符的打印
<feed>	送纸
<image>	光栅图像的打印
<logo>	NV标识的打印
<barcode>	条形码的打印
<symbol>	二维码的打印
<area>	打印区域的设定
<direction>	打印方向的设定
<position>	打印位置的设定
<line>	直线的描绘
<rectangle>	长方形的描绘
<command>	指令的插入

图43A

```

<!-- 以页面模式打印字符 "ABCDE" -->
<page>
<text>ABCDE</text>
</page>

```

图43B

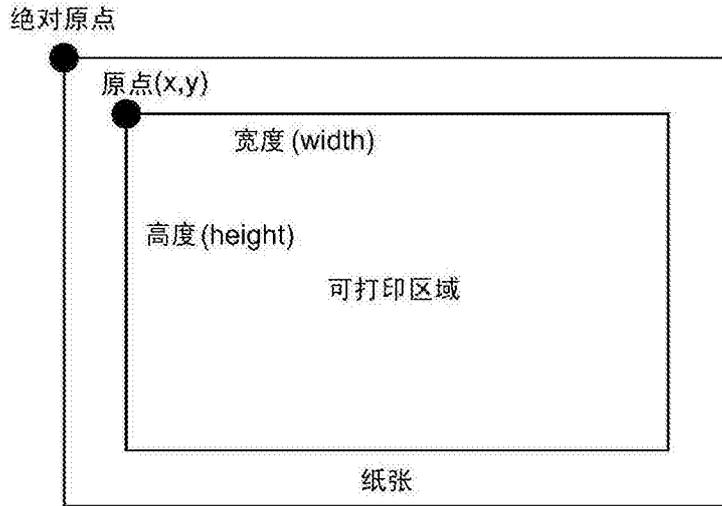


图44A

```

<!-- 指定原点 ( 100,50 )、200点宽、30点高的打印区域来打印字符 "ABCDE" -->
<page>
<area x="100" y="50" width="200" height="30" />
<text>ABCDE</text>
</page>
    
```

图44B

属性值	说明
"left_to_right" (初始值)	左→右 (不旋转。以左上为始点向右打印)
"bottom_to_top"	下→上 (逆时针旋转 90 度。以左下为始点向上打印)
"right_to_left"	右→左 (旋转 180 度。以右下为始点向左打印)
"top_to_bottom"	上→下 (顺时针旋转 90 度。以右上为始点向下打印)

图45A

```

<!-- 顺时针旋转 90 度打印字符 "ABCDE" -->
<page>
<direction dir="top_to_bottom" />
<text>ABCDE</text>
</page>

```

图45B

```

<!-- 将在 area 要素中指定的区域内的打印开始位置指定为 ( 50, 30 ) 来打印字符 "ABCDE" -->
<page>
<area x="100" y="50" width="200" height="100" />
<position x="50" y="30" />
<text>ABCDE</text>
</page>

```

图46

属性值	说明
"thin"	实线：细
"medium"	实线：中粗
"thick"	实线：粗
"thin_double"	双划线：细
"medium_double"	双划线：中粗
"thick_double"	双划线：粗

图47A

```

<!-- 用细实线描绘以开始位置 ( 100, 0 )、结束位置 ( 500, 0 ) 为原点的直线 -->
<page>
<line x1="100" y1="0" x2="500" y2="0" style="thin" />
</page>

```

图47B

属性值	说明
"thin"	实线：细
"medium"	实线：中粗
"thick"	实线：粗
"thin_double"	双划线：细
"medium_double"	双划线：中粗
"thick_double"	双划线：粗

图48A

```
<!-- 用细双划线描绘以开始位置 ( 100, 0 ), 结束位置 ( 500, 200 ) 为顶点的四角形 -->
<page>
<rectangle x1="100" y1="0" x2="500" y2="200" style="thin_double" />
</page>
```

图48B

属性值	说明
"no_feed"	无送纸剪切 (不执行送纸而剪切)
"feed"	送纸剪切 (执行送纸到剪切位置为止并剪切)
"reserve"	预约剪切 (执行后续的打印并在到达剪切位置后剪切)

图49A

```
<!-- 进行送纸剪切 -->
<cut type="feed" />
```

图49B

属性值	说明
"drawer_1" (初始值)	抽屉弹出连接器第 2 管脚
"drawer_2"	抽屉弹出连接器第 5 管脚

图50A

属性值	说明
"pulse_100" (初始值)	100 毫秒的信号
"pulse_200"	200 毫秒的信号
"pulse_300"	300 毫秒的信号
"pulse_400"	400 毫秒的信号
"pulse_500"	500 毫秒的信号

图50B

```

<!-- 向抽屉弹出连接器第 2 管脚输出 100 毫秒的脉冲信号 -->
<pulse drawer="drawer_1" time="pulse_100" />

```

图50C

属性值	说明
"none"	停止
"pattern_a" (初始值)	模式 A
"pattern_b"	模式 B
"pattern_c"	模式 C
"pattern_d"	模式 D
"pattern_e"	模式 E
"error"	错误鸣动模式
"paper_end"	无纸鸣动模式

图51A

属性值	说明
"0"	无限
"1" ~ "255"	重复指定次数

图51B

```

<!-- 将模式 A 鸣响 3 次 -->
<sound pattern="pattern_a" repeat="3" />

```

图51C

子要素	数据类型	说明
timeout	int	以毫秒指定请求超时时间 1000 ~ 60000 (整数)
displaydata	string	指定顾客显示器控制用 XML

图52A

```

<data>
  <type>display</type>
  <timeout>10000</timeout>
  <displaydata>
    <!-- 顾客显示器控制用 XML -->
    <epos-display xmlns=
      .
      .
    </epos-display>
  </displaydata>
</data>
    
```

图52B

子要素	数据类型	说明
resultdata	string	接收发送给顾客显示器的数据的响应

图53A

```

<data>
  <type>onxmlresult</type>
  <resultdata>
    <response success="true" code="" status="" />
  </resultdata>
</data>
    
```

图53B

子要素	功能
<window>	窗口的设定
<text>	字符的显示
<cursor>	光标的显示
<blink>	设定画面的闪烁
<brightness>	设定显示器的亮度
<marquee>	设定字符串的滚动显示
<clock>	时刻的显示
<clear>	当前窗口的显示的清除
<reset>	显示器显示的初始化
<command>	ESC/POS指令的执行

图54A

```

<!-- 作为子要素追加了 text 标签的示例 -->
<epos-display xmlns="http://www.epson-pos.com/schemas/2012/09/epos-display">
  <text>Hello world!!</text>
</epos-display>

```

图54B

属性值	说明
"true" 或者 "1"	显示成功
"false" 或者 "0"	显示失败

图55A

属性值	说明
"EDSP_NOT_FOUND"	设备找不到
"EDSP_NOT_OPEN"	设备的打开失败
"EDSP_INVALID_WINDOW"	指定未登记的窗口
"EDSP_SYSTEM_ERROR"	未预期的错误发生
"EDSP_SCHEMA_ERROR"	XML 结构有误
"EX_BADPORT"	与设备的内部通信错误发生
"EX_TIMEOUT"	与设备的通信中超时错误发生
"EX_INVALID_VALUE"	检测出参数不正确

图55B

```

<!-- 成功时 -->
<response success="true" code="" status=" " />
<!-- 失败时 -->
<response success="false" code="EDSP_NOT_FOUND" status=" " />
    
```

图55C

属性值	说明
"1"	窗口 1
"2"	窗口 2
"3"	窗口 3
"4"	窗口 4

图56

属性值	说明
"overwrite" (初始值)	若显示位置位于上段右端位置时进行字符显示, 则使显示位置向下段左端移动
"v_scroll"	若位于下段右端时进行字符显示, 则使显示位置向上段左端移动 若显示位置位于上段右端位置时进行字符显示, 则使显示位置向下段左端移动
"h_scroll"	若位于下段右端时进行字符显示, 则使已经在显示的下段的显示字符滚动到上段, 清除下段的显示 若显示位置位于右端位置时进行字符显示, 则使在光标所在的行已经在显示的所有的显示字符向左滚动 1 个字符, 将新的显示字符显示于右端

图57A

属性值	说明
"true" 或者 "1"	实施删除
"false" 或者 "0" (初始值)	不实施删除

图57B

错误值	说明
"EX_INVALID_VALUE"	窗口作成时, x、y、width、height 涉及到已经定义完成的窗口的范围的情况
"EDSP_INVALID_WINDOW"	窗口移动时, 所指定的窗口不存在的情况

图57C

```

<!-- 定义从左上起宽度 10、高度 2、纵向滚动的窗口 1 -->
<window number="1" x="1" y="1" width="10" height="2" scrollmode="v_scroll"/>
<!-- 定义从中央起宽度 10、高度 2、纵向滚动的窗口 2 -->
<window number="2" x="11" y="1" width="10" height="2" scrollmode="v_scroll"/>
<!-- 使当前窗口移动至窗口 1 -->
<window number="1"/>
<!-- 删除窗口 2 -->
<window number="2" destroy="true"/>
    
```

图57D

属性值	说明
"true" 或者 "1"	明暗反转
"false" 或者 "0" (初始值)	不使明暗反转

图58A

属性值	说明
"en" (初始值)	英语
"ja"	日语

图58B

```

<!-- 从左上起显示字符串 -->
<text x="1" y="1" lang="ja"> 胡萝卜 \100</text>
<!-- 从现在的光标位置起以明暗反转状态显示字符串 -->
<text reverse="true">Welcome!!</text>

<!-- 将当前窗口切换成窗口 2 进行显示 -->
<window number="2"/>
<text x="1" y="1" lang="ja"> 胡萝卜 $1</text>
    
```

图58C

属性值	说明
"top_left"	向左上移动
"top_right"	向右上移动
"bottom_left"	向左下移动
"bottom_right"	向右下移动

图59A

属性值	说明
"none"	不显示光标
"underline"	以下划线形状来显示

图59B

```

<!-- 使光标向显示器左上移动 -->
<cursor x="1" y="1"/>
<!-- 使光标向当前窗口内的左上移动 -->
<cursor moveto="top_left" type="underline"/>

```

图59C

属性值	说明
"0"	结束闪烁成为连续点亮状态
"1" ~ "12700"	闪烁的间隔的毫秒
"12701" ~ "12750"	以保持了显示内容的状态熄灭

图60A

```

<!-- 将画面的闪烁间隔设为 500ms -->
<blink interval="500"/>
<!-- 解除画面的闪烁 -->
<blink interval="0"/>
<!-- 使画面熄灭。维持显示内容 -->
<blink interval="12750"/>
<cursor moveto="top_left" type="underline"/>

```

图60B

属性值	说明
"20"	亮度 20%
"40"	亮度 40%
"60"	亮度 60%
"100"	亮度 100%

图61A

```

<!-- 将亮度设定为 20%      -->
<brightness value="20"/>
<!-- 将亮度设定为 40%      -->
<brightness value="40"/>
<!-- 将亮度设定为 60%      -->
<brightness value="60"/>
<!-- 将亮度设定为 100%     -->
<brightness value="100"/>

```

图61B

属性值	说明
"walk" (初始值)	从窗口右端起显示字符串
"place"	从窗口左端起显示字符串

图62A

属性值	说明
"0" (初始值)	无限重复
"1" ~ "127" (整数)	重复指定次数

图62B

属性值	说明
"0" ~ "2000" (整数)	每显示 1 个字符的待机间隔

图62C

属性值	说明
"100" ~ "2000" (整数)	显示到字符串末尾后的待机时间

图62D

属性值	说明
"en" (初始值)	英语
"ja"	日语

图62E

```
<!-- 从画面右边起以 200ms/字符进行滚动显示, 以 1000ms 间隔重复 -->
<marquee uwait="200" rwait="1000"> 欢迎 </marquee>
<!-- 从画面左边起以 200ms/字符进行滚动显示, 以 1000ms 间隔重复 -->
<marquee uwait="200" rwait="1000" format="place">Welcome!!</marquee>
```

图62F

```
<clock/>
```

图63

```
<clear/>
```

图64

```
<reset/>
```

图65

```
<command>01025AEF3B405C</command>
```

图66

子要素	数据类型	说明
keycode	int	接收在 POS 键盘输入的键码的 10 进制数
ascii	string	接收根据键码生成的字符 设定变换成基于 Shift 按下状态的大写字符、符号等的字符

图67A

替换前字符串	替换后
& (与号)	&
> (大于号)	>
< (小于号)	<
" (双引号)	"
' (单引号)	'

图67B

```
<data>
  <type>onkeypress</type>
  <keycode>49</keycode>
  <ascii>a</ascii>
</data>
```

图67C

子要素	数据类型	说明
keycode	int	指定判断为字符串的开始的键码的 10 进制数

图68A

```
<data>
  <type>setprefix</type>
  <keycodes array="true">49</keycodes>
  <keycodes>50</keycodes>
  <keycodes>51</keycodes>
  <keycodes>52</keycodes>
</data>
```

图68B

子要素	数据类型	说明
input	string	接收输入字符串 不包含用于判断字符串的开始的键和 Enter 键
prefix	int	接收判断为字符串的开始的键码

图69A

替换前字符串	替换后
& (与号)	&
> (大于号)	>
< (小于号)	<
" (双引号)	"
' (单引号)	'

图69B

```
<data>
  <type>onstring</type>
  <input>&amp;2398749238429</input>
  <prefix>49</prefix>
</data>
```

图69C

子要素	数据类型	说明
command	string	指令列的 16 进制表现字符串

图70A

```
<data>
  <type>sendcommand</type>
  <command>1B123344FF1F5D3C</command>
</data>
```

图70B

子要素	数据类型	说明
status	string	数据接收的状态
data	string	数据列的 16 进制表现字符串

图71A

要素值	说明
"SUCCESS"	数据接收成功

图71B

```

<data>
  <type>oncommandreply</type>
  <status>SUCCESS</status>
  <data>1B123344FF1F5D3C</data>
</data>

```

图71C

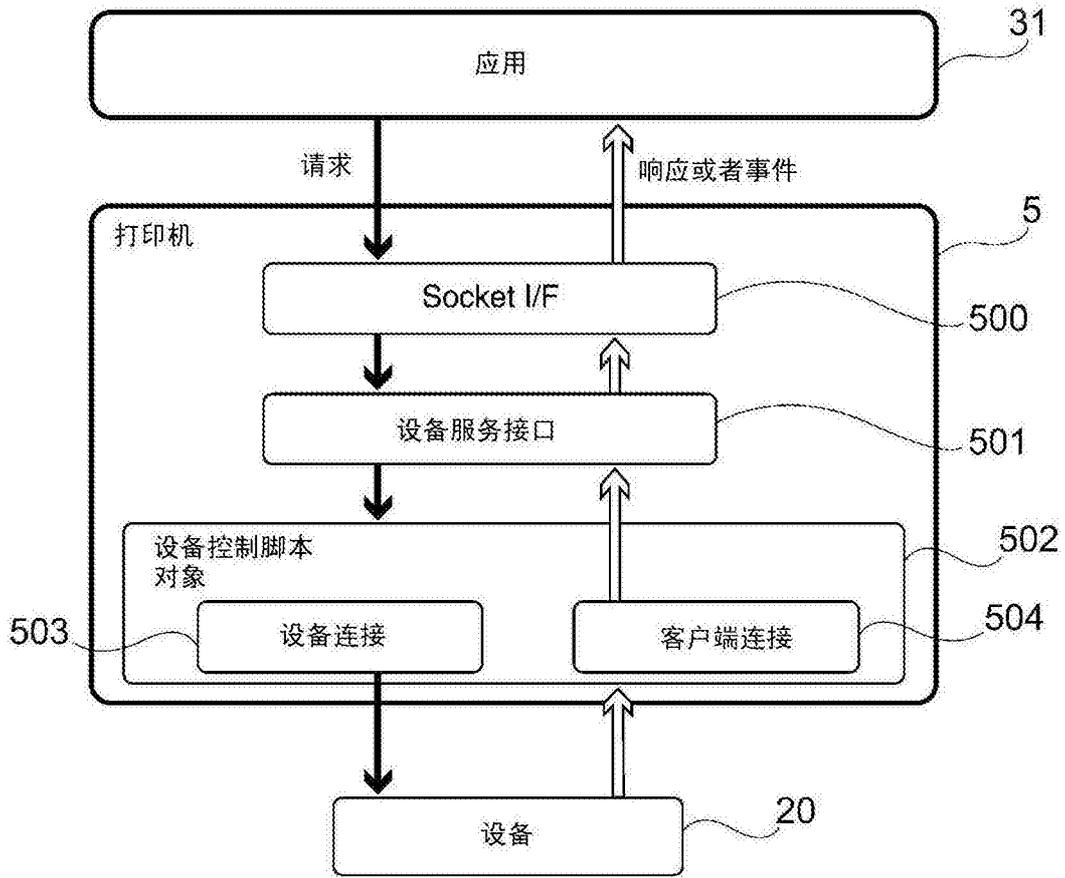


图72

对象	说明
客户端连接	用于向应用侧的设备对象进行数据发送的对象
设备连接	进行与设备的数据收发的对象

图73

设定值	说明
type_keyboard	在使用键盘设备的情况下指定
type_scanner	在使用条形码扫描器的情况下指定
type_simple_serial	在进行单纯的串行通信的情况下指定

图74

设定值	说明
group_hid	在使用能够通过 HID 驱动器执行动作的输入设备的情况下指定
group_serial	在使用串行通信设备的情况下指定
group_other	在使用其他设备的情况下指定

图75

```

exports.Keyboard_Generic = Keyboard_Generic;           // exports 声明

function Keyboard_Generic(clientConn, deviceConn){     // 具有 2 个变量,
                                                        // 与文件名同名
    this.DEVICE_TYPE = 'type_keyboard';               // 具有 DEVICE_TYPE 属性
    this.DEVICE_GROUP = 'group_hid';                  // 具有 DEVICE_GROUP 属性

    this.clientConn = clientConn;
    this.deviceConn = deviceConn;
    .....
    .....
}

Keyboard_Generic.prototype = {
    onDeviceData : function(event, keycode, ascii){...}, // 具有 onDeviceData 方法
    setprefix : function(data){...}                     // 有设备对象对应的方法
}
    
```

图76

要素	API	说明
方法		
	send	向在浏览器中执行动作的设备对象发送数据

图77

要素	API	说明
方法		
	send	向串行通信设备发送数据

图78

要素	API	说明
事件	onDeviceData	键输入接收事件
	onDeviceData	从串行通信设备接收数据事件
	任意事件	在浏览器中执行动作的设备对象的API 执行结果接收事件

图79

```
data = {'keycode' : 49, 'ascii' : '1'};
clientConn.send('onkeypress', data)
```

图80

值	说明
1	键按下
2	键弹起

图81

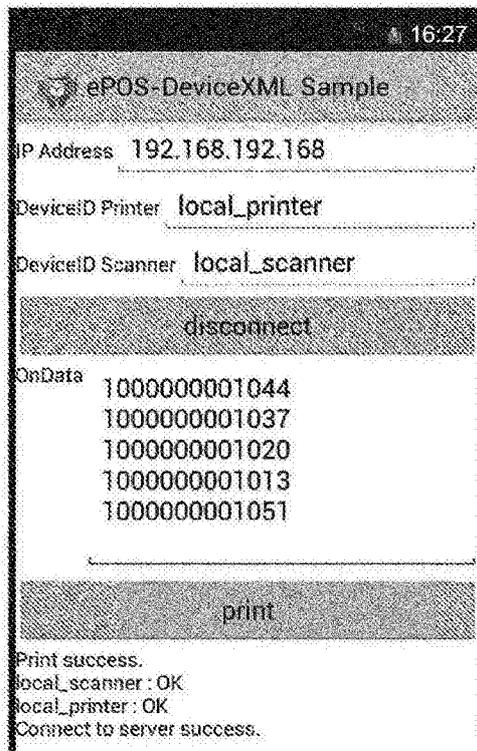


图82

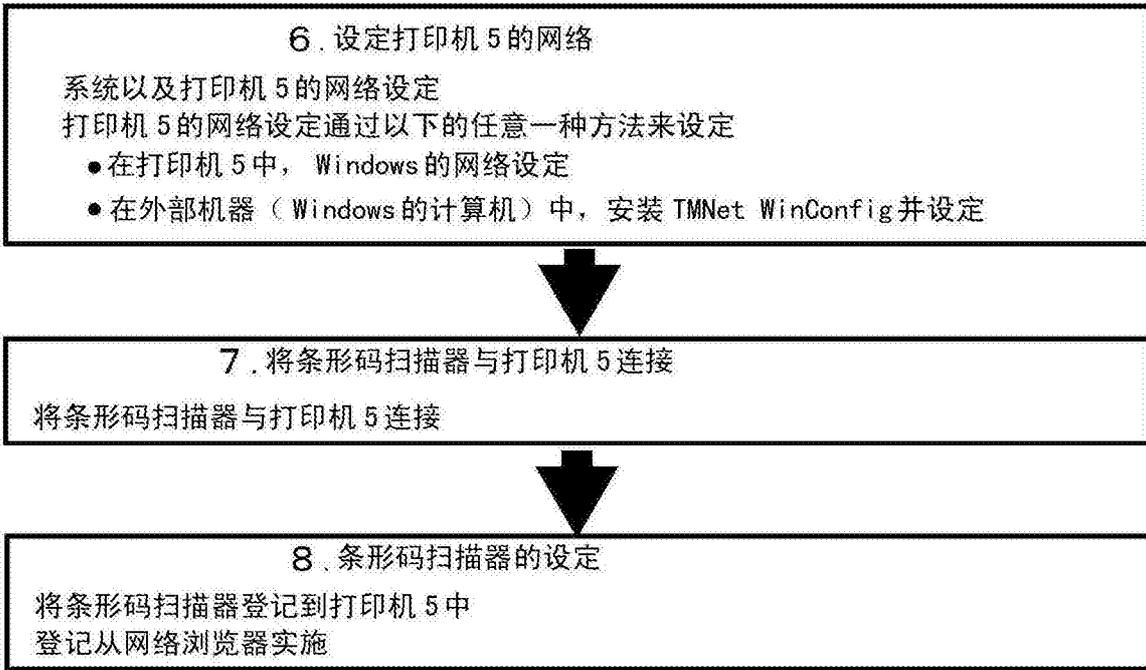


图83

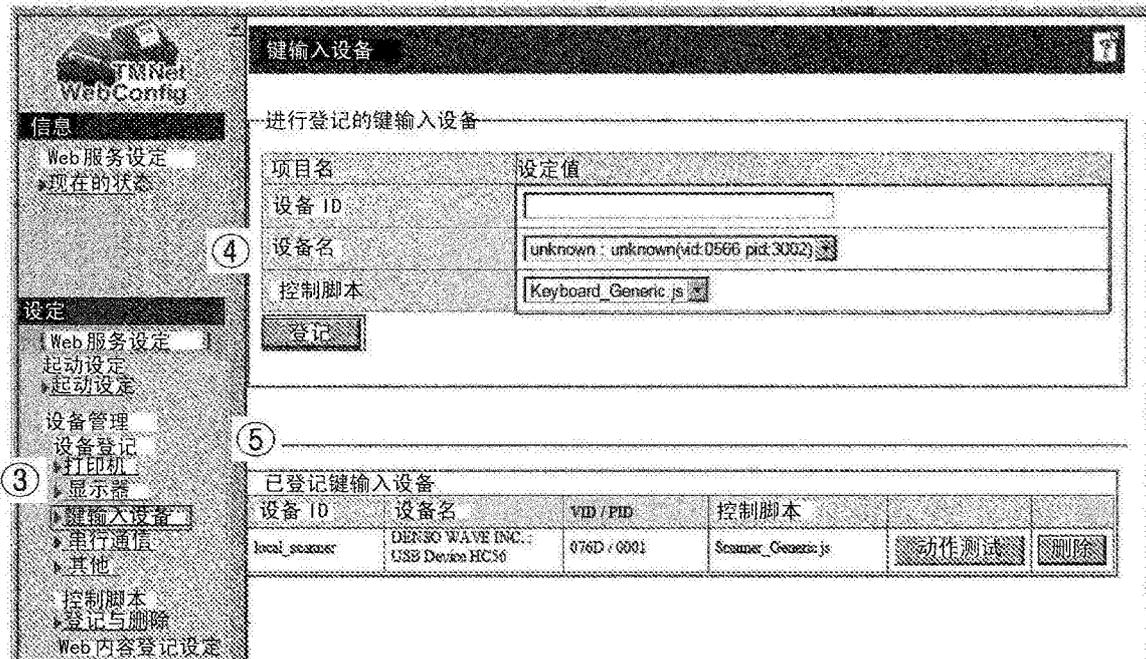


图84A

项目名	设定值
设备 ID	local_scanner
设备名	从列表中选择相符的设备
控制脚本	Scanner_Generic.js

图84B