# 电源管理后台设计说明书

**变更记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本** | **修订时间** | **修订人** | **修订类型** | **修订章节** | **修订内容** |
| 1.0 | 2021-01-20 | 唐杰 | A | ALL | 建立 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

注1：修订类型分为A-ADDED，M-MODIFIED，D–DELETED

注2：对该文件内容增加、删除或修改均需填写此记录，详细记载变更信息，以保证其可追溯性

目录

[电源管理后台设计说明书 1](#_Toc29344)

[一、 概述 3](#_Toc24783)

[1.1. 电源模式 3](#_Toc1936)

[1.1.1. 计算机 3](#_Toc21902)

[1.1.2. 显示器 3](#_Toc8771)

[二、 总体设计 3](#_Toc24521)

[三、 模块设计 4](#_Toc9528)

[3.1. 电源管理 4](#_Toc24604)

[3.2. 节能计时器 4](#_Toc15947)

[3.3. 计算机节能 5](#_Toc9690)

[3.4. 显示管理 6](#_Toc7368)

[3.4.1. 显示节能 6](#_Toc21341)

[3.4.2. 显示亮度调节 7](#_Toc12531)

[3.5. 电源托盘 7](#_Toc23007)

[3.6. 通知区域 8](#_Toc2302)

[3.7. 键盘按键 8](#_Toc28969)

[四、 接口设计 9](#_Toc16003)

[4.1. 枚举值 9](#_Toc20882)

[4.2. DBUS接口 9](#_Toc16637)

[4.2.1. 函数 9](#_Toc27433)

[4.2.2. 属性 10](#_Toc27858)

[4.2.3. 信号 10](#_Toc27390)

# 概述

本文档对电源设置后台程序的概要设计进行了阐述。电源设置后台是作为一个插件由控制中心后台(kiran-session-daemon)加载，目前主要应用于桌面环境，为上层的桌面应用提供必要的接口。。

## 电源模式

### 计算机

计算机电源模式分为正常(normal)、挂起(suspend)、待机(standby)、休眠(hibernate)、睡眠(sleep)、关机(shutdown)。

挂起模式下，系统将机器的硬盘、显示器等外部设备停止工作，而CPU、内存仍然工作，等待用户随时唤醒，再次唤醒需要按键盘上的Fn键数次。

待机模式下，将当前处于运行状态的数据保存在内存中，机器只对内存供电，而硬盘、屏幕和CPU等部件则停止供电。由于数据存储在速度快的内存中，因此进入等待状态和唤醒的速度比较快。不过这些数据是保存在内存中。

体眠模式下，它将内存中的数据保存于硬盘中，使CPU也停止工作，当再次使用时需按开关机键，机器将会恢复到您的执行休眠时的状态，而不用再次执行启动操作系统这个复杂的过程。

睡眠模式是结合了待机和休眠的优点，系统将内存中的数据全部转存到硬盘上的休眠文件中（类似休眠），然后关闭除了内存外所有设备的供电，让内存中的数据依然维持着（类似待机）。当我们想要恢复的时候，如果在睡眠过程中供电没有发生过异常，就可以直接从内存中的数据恢复，但如果睡眠过程中供电异常，内存中的数据已经丢失了，还可以从硬盘恢复，只是速度会慢一点。

目前版本不支持待机(standby)和睡眠(sleep)。

### 显示器

通过X Display Power Management Signaling(DPMS)扩展，可以将显示器设置分为四个等级的电源模式，分别为开启(on)、待机(standby)、挂起(suspend)、关闭(off)。开启状态下显示器正常工作。待机和挂起状态下显示器以低功率状态运行。关闭状态下显示器变为未激活，需要进行唤醒。显示器待机、挂起和关闭都会让显示器进入黑屏(blank)状态。

# 总体设计

电源管理包括了节能计时器、计算机节能、显示管理、电源托盘、通知区域和键盘按键，各模块之间的关系如下：

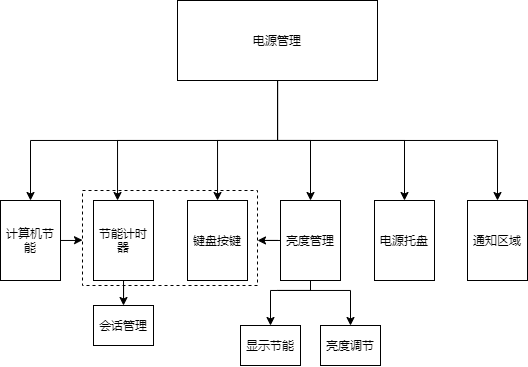


图 1 电源管理整体设计

# 模块设计

## 电源管理

该模块主要是用于初始化子模块，协调各子模块的功能，并向外部提供dbus接口。

## 节能计时器

用户可以设置当计算机进入空闲状态一定时间后，计算机或者显示设备进入节能模式。

空闲状态从会话管理的dbus接口中进行捕获，当收到空闲信号时，首先会发送显示器变暗的信号，然后再根据用户设定的计算机/显示设备进入节能状态的时间创建相应的计时器，当计时器时间到达后，发送需要对计算机/显示设备进行节能的信号。因此这里一共有四个信号：

* 正常状态信号。
* 显示器变暗信号
* 显示器进入节能状态信号
* 计算机进入节能状态信号

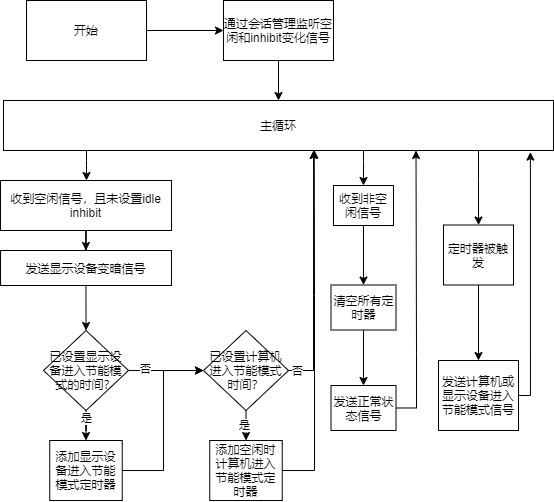


图 2 节能计时器流程图

## 计算机节能

当电源管理收到节能计时器发送的计算机需要进入节能模式的信号时，根据用户偏好设置，计算机会进入挂起、休眠或者关机。这三个操作实际是通过调用systemd-logind服务实现，在计算机进入挂起和休眠模式之前，会根据用户偏好设置来决定是否进行锁屏操作。

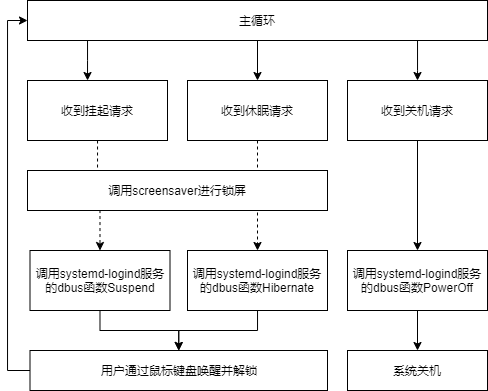


图 3 计算机节能设置

## 显示管理

### 显示节能

显示节能跟计算机节能类似，即当收到显示设备变暗或者节能信号时，将显示设备变暗或者变黑。

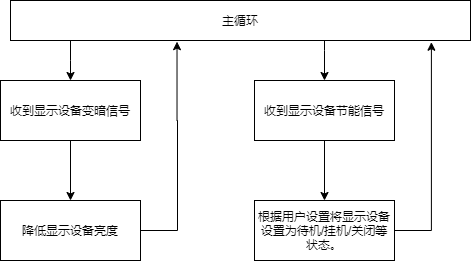


图 4 显示节能流程

### 显示亮度调节

可进行亮度调节的设备包括显示器和键盘。

设置显示器的背光(亮度)值的方式为：

* 通过XRR扩展修改显示设备(Output)的Backlight/BACKLIGHT属性来调整显示器的背光值。
* 根据显示器型号，通过/sys/class/backlight/${model}/brightness文件读写背光值，通过/sys/class/backlight/${model}/max\_brightness获取最大可设置的背光值。

设置

键盘亮度值可通过upowerd服务提供的dbus接口(org.freedesktop.UPower.KbdBacklight)设置。

亮度设置模块会监听其它模块、gsettings和按键等信号，当发生以下几种情况时会收到亮度调节请求并触发显示亮度变化：

1. 通过UI界面或者dbus请求调节亮度。
2. 通过键盘亮度设置按钮调节显示器亮度。
3. 用户如果设置了空闲一定时间后显示设备亮度变暗，那么节能计时器在空闲时间到了之后就会发送显示设备亮度需要变暗的信号，然后由该模块调节显示亮度。

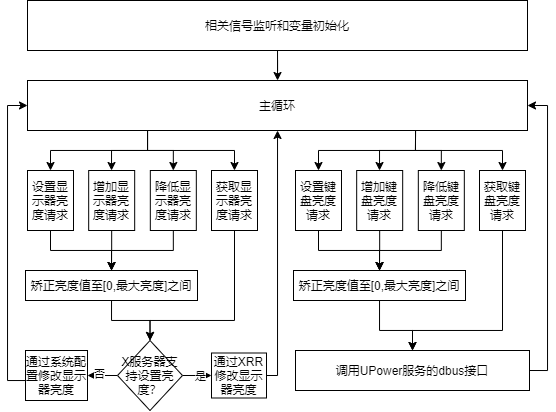


图 5 显示亮度设置

## 电源托盘

电源托盘区域可根据电池电量状态显示不同图标，状态包括：

* 电量正常。
* 电池正在充电。
* 电池电量过低。(<20%)
* 电池电量严重过低。(<10%)

## 通知区域

当使用电池的设备处于不同状态时，对用户进行提示。使用电池的设备可能为：计算机电池、UPS、显示器、鼠标、键盘、PDA(掌上电脑)、手机。

不同状态的提示（提示信息只是作为参考，最终的术语待定）如下：

|  |  |
| --- | --- |
| 电池状态 | 提示信息 |
| 电池容量较低 | 电池容量过低，电池可能损坏 |
| 电池断开充电 | 电池断开充电 |
| 电池开始充电 | 电池正在充电 |
| 电池充满电 | 电池已充满 |
| 电池电量过低 | 设备电量不足，及时充电或更换电池 |
| 电池电量严重过低 | 设备很快就要被停止使用了，设备会进入待机、休眠或者关机状态。 |
| 电池电量不足 | 设备即将停止使用。(给用户预留几十秒的时间，以便让用户能够插上电源充电或做其它操作) |

表 1 电池不同状态的提示信息

## 键盘按键

与电源管理相关的按键如下：

|  |  |
| --- | --- |
| 按键(keysym) | 触发行为 |
| XF86XK\_PowerOff | 根据配置决定挂起/休眠/黑屏/关机等 |
| XF86XK\_Suspend | 根据配置决定挂起/休眠/黑屏/关机等 |
| XF86XK\_Sleep | 根据配置决定挂起/休眠/黑屏/关机等 |
| XF86XK\_Hibernate | 根据配置决定挂起/休眠/黑屏/关机等 |
| XF86XK\_MonBrightnessUp | 增加显示器亮度 |
| XF86XK\_MonBrightnessDown | 降低显示器亮度 |
| XF86XK\_ScreenSaver | 锁屏 |
| XF86XK\_Battery | 显示电源相关信息 |
| XF86XK\_KbdBrightnessUp | 增加键盘亮度 |
| XF86XK\_KbdBrightnessDown | 减少键盘亮度 |
| XF86XK\_KbdLightOnOff |  |

表 2 电源相关按键

# 接口设计

## 枚举值

|  |  |  |
| --- | --- | --- |
| 枚举类 | 枚举值 | 说明 |
| 设备类型 | POWER\_DEVICE\_COMPUTER | 计算机 |
| POWER\_DEVICE\_MONITOR | 显示器 |
| POWER\_DEVICE\_KBD | 键盘 |
| 供电类型 | POWER\_SUPPLY\_BATTERY | 电池 |
| POWER\_SUPPLY\_AC | 交流电 |
| POWER\_SUPPLY\_UPS | UPS供电 |
| 事件类型 | POWER\_EVENT\_KEY\_PRESSED\_POWEROFF | 按下关机键 |
| POWER\_EVENT\_KEY\_PRESSED\_SUSPEND | 按下挂机键 |
| POWER\_EVENT\_KEY\_PRESSED\_SLEEP | 按下休眠键 |
| POWER\_EVENT\_KEY\_PRESSED\_HIBERNATE | 按下休眠键 |
| POWER\_EVENT\_LID\_IS\_CLOSED | 盖子合上 |
|  |  |
| 行为类型 | POWER\_ACTION\_DISPLAY\_STANDBY | 显示器待机 |
| POWER\_ACTION\_DISPLAY\_SUSPEND | 显示器挂起 |
| POWER\_ACTION\_DISPLAY\_OFF | 显示器黑屏 |
| POWER\_ACTION\_COMPUTER\_SUSPEND | 计算机挂起 |
| POWER\_ACTION\_COMPUTER\_HIBERNATE | 计算机休眠 |
| POWER\_ACTION\_COMPUTER\_SHUTDOWN | 计算机关机 |
|  |  |  |

## DBUS接口

### 函数

SetIdleAction(Int32 device, Int32 supply, Int32 idle\_timeout, Int32 action\_mask);

函数说明：设置当空闲时间超过idle\_timeout时，在不同供电（电池/电源）情况下不同设备（计算机/显示器）的处理方式（待机/休眠等）。

参数说明：

* device: 设备类型，可以是计算机或者显示器。
* supply：供电方式，可以是电池和交流电。
* idle\_timeout：设备进入空闲状态后等待的超时时间（秒）。
* action\_mask：超时时间到达后触发的行为，可以是待机、休眠等操作

返回值：如果参数不合法或者设备不支持则返回错误。

Struct(Int32, Int32) GetIdleAction(Int32 device, Int32 supply);

函数说明：在不同供电情况下不同设备的空闲超时设置和触发的行为。

函数说明：见SetIdleAction

SetEventAction(Int32 event, Int32 action\_mask)

函数说明：当事件类型event触发后，执行action\_mask指定的行为。

参数说明：参考[枚举值](#_枚举值)

Int32 GetEventAction(Int32 event);

函数说明：获取事件类型event被触发时的行为。

参数说明：参考[枚举值](#_枚举值)

SetBrightness(Int32 device, Int32 brightness);

函数说明：设置指定设备的亮度百分比。

参数说明：

* device: 设备类型，可以是显示器或键盘。
* brightness：设置的亮度百分比，[0,100]。

返回值：如果参数错误或者设备不支持，则返回错误。

Int32 GetBrightness(Int32 device);

函数说明：获取指定设备亮度百分比。

返回值：返回设备的亮度，如果设备不支持或者不存在则返回-1。

### 属性

|  |  |
| --- | --- |
| 属性 | 说明 |
| on\_battery | 系统是否使用电池 |
| lid\_is\_present | 系统是否存在盖子设备。（例如笔记本上盖） |
| has\_backlight | 显示器是否支持亮度/背光调节 |
|  |  |
|  |  |

### 信号

IdleActionChange(Int32 device, Int32 supply);

信号说明：在通过supply供电情况下的设备device的空闲超时和超时后的行为发生了变化。

EventActionChanged(Int32 event);

信号说明：事件event的行为发生了变化。

BrightnessChanged(Int32 device);

信号说明：设备device的亮度发生了变化。