



PowerControl 教程

版本：2026. 1. 18

前言：

最新教程[对比版本号]上传在 github

Github: <https://github.com/viklion/PowerControl>

↑含群晖、飞牛、unraid、istoreOS 小白级详细部署容器教程↑

视频教程：<https://www.bilibili.com/video/BV1cykZY7Er9>

(视频教程较早，所用版本落后，界面功能会有部分差异)

本容器可以实现的功能：

- 1、通过访问网页，本地/远程（端口转发或反向代理）唤醒设备，关闭设备，支持 Windows、Linux、MacOS（MacOS 不支持从关机状态唤醒）
- 2、通过巴法云接入米家，通过米家、小爱同学、巴法微信小程序、巴法 app 实现设备开机、关机
- 3、接入 homeassistant 通过 ha 实现设备开机、关机
- 4、设置 ios 快捷指令实现设备开机、关机，也可用 siri
- 5、通过 Server 酱、Qmsg 酱等推送消息
- 6、支持多设备管理
- 7、支持定时任务

必要的前置条件：

- 1、运行 docker 且 24 小时运行的设备，比如 nas、软路由等
- 2、电脑与 docker 设备在同一局域网
- 3、网络唤醒需要网卡支持，可能需要进 bios 设置。无线网卡可能不支持
- 4、关机和 ping 功能可能需要关闭或配置 windows 专用网络的防火墙
(一般同一网段不需要动防火墙，跨网段可能需要关掉或者单独配置)

目录

一、 更新内容：	1
二、 部署容器	3
三、 参数配置说明	5
四、 前置设置	8
五、 配置远程关机	14
六、 配置网络唤醒	19
七、 接入巴法云并接入米家	24
八、 接入 homeassistant	27
九、 ios 快捷指令	32
十、 yaml 配置文件详解	33
十一、 反馈	36

一、更新内容：

- 2024. 12. 27: 更新 UDP 方式关机方法教程, 涉及目录[前置设置]中的防火墙配置和目录[配置远程关机] (适用于 windows 家庭版、精简版无法配置的, 适用于怎么配置都无法成功的)
- 2025. 1. 8: 更新在 homeassistant 中创建电脑实体实现控制教程, 见目录[接入 homeassistant]
- 2025. 1. 15: 新增自动清理日志功能: 目录[配置容器参数], web 加入查看日志入口: 目录[部署容器]
- 2025. 1. 25: 新增支持非 root 用户启动容器, 加入参数-u uid:gid, 比如-u 1000:100, 目录[部署容器] (2025. 7. 5 已更新)。新增 shell 指令可关闭 Linux 设备, 或使用类似 Remote Shutdown Manager 的软件, 目录[配置远程关机]
- 2025. 2. 15: 增加网络唤醒检测能否收到唤醒魔包的软件 (WakeOn LanMonitor.exe)。网络唤醒: 可选发送至广播地址或设备地址, 一般默认 (全局广播) 无需修改, 目录[配置网络唤醒]
- 2025. 6. 1: 增加 web 修改网络唤醒端口的设置, 增加 web 修改关机指令超时时长的设置。
- 2025. 7. 5: 现使用更优雅的环境变量 PUID 和 PGID 以非 root 用户启动容器。
- 2025. 7. 15: 更新 homeassistant 配置文件代码语法。更新关闭 Linux 设备自定义指令 sshpass 教程, web 代码页中加入 sshpass 代码方便复制、修改。
- 2025. 7. 25: 新增自定义指令快速生成页面, 快速生成 Linux、Mac OS 关机指令。网络唤醒和 ping 新增支持自定义指令。
- 2025. 8. 18[V3. 1]: 支持多设备管理。*主要 api 改动:
 - 总览: /config?key=密钥 --> 现已改为设备总览页, 点击设备进入设备服务编辑页
 - 唤醒: /wol?key=密钥 --> /wol/设备 id 或别名?key=密钥

关机：/shutdown?key=密钥 --> /shutdown/设备 id 或别名?key=密钥

ping：/ping?key=密钥 --> /ping/设备 id 或别名?key=密钥

例：设备 id 为 device01，别名设置为 windows，则 api 接口为/wol/device01?key=密钥，或，/wol/windows?key=密钥

- 2025.12.6[V3.8]：更新 homeassistant 配置代码语法（configuration.yaml），旧版语法将在 homeassistant 更新至 2026.6 版本后失效（2026 年 6 月版本）
- 2026.1.1[V3.9]：更新关机配套软件 PCshutdown，支持自定义监听端口，支持关机、重启、睡眠、休眠
- 2026.1.18[V4.0]：添加定时开关机功能：支持单次（日期时间）和循环（cron 表达式）两种定时任务；支持定时任务执行前 x 分钟推送通知提醒

二、部署容器

建议在 [github](#) 中复制代码

需要修改 3 处，标红（非 root 运行修改 5 处）

1、docker-cli：

- 默认 root 用户运行容器：

```
docker run -d -v /your/path:/app/data -e WEB_PORT=7678 -e  
WEB_KEY=yourkey --network host --restart unless-stopped --name  
powercontrol viklion/powercontrol:latest
```

- 设置变量 PUID 和 PGID, 指定 user 运行：

```
docker run -d -v /your/path:/app/data -e WEB_PORT=7678 -e  
WEB_KEY=yourkey -e PUID=1000 -e PGID=100 --network host --restart  
unless-stopped --name powercontrol viklion/powercontrol:latest
```

2、docker-compose（注意检查缩进）

```
services:  
  powercontrol:  
    image: viklion/powercontrol:latest  
    container_name: powercontrol  
    volumes:  
      - /your/path:/app/data  
    environment:  
      - WEB_PORT=7678  
      - WEB_KEY=yourkey  
      # 默认 root 用户运行，去掉下两行的#，设置指定 user 运行  
      #- PUID=1000  
      #- PGID=100  
    restart: unless-stopped  
    network_mode: host
```

/your/path：主机映射目录

WEB_PORT=7678：网页端口（不设置该环境变量默认使用端口：7678）

WEB_KEY=yourkey：密钥（不设置该环境变量默认使用密钥：admin）

PUID=1000：

PGID=100：

*注意网络模式为 host（不支持 bridge，支持 ipvlan 和 macvlan）

容器启动后访问：

ip:端口	首页
ip:端口/config?key=密钥	进入配置总览

api 接口：

ip:端口/wol/设备 id 或别名?key=密钥	网络唤醒
ip:端口/shutdown/设备 id 或别名?key=密钥	关闭设备
ip:端口/ping/设备 id 或别名?key=密钥	ping 设备
ip:端口/pdf	进入 pdf 教程页
ip:端口/logs?key=密钥	查看日志记录
ip:端口/changelog	查看更新日志
ip:端口/code	查看复制教程内的代码
ip:端口/getcommand	快速生成关机自定义指令

*以下为高级配置项：

定制容器：环境变量 APK_ADD 和 SHELL_ADD

APK_ADD: 创建容器时安装 alpine 系统指定软件包, 如 APK_ADD="vim openssh",
会在创建容器时安装 vim 和 openssh

SHELL_ADD: 创建容器时运行 bash 指令, 如 SHELL_ADD="echo 'hello world' &&
pip install fastapi", 会在创建容器时打印'hello world' 并且安装 python fastapi

三、参数配置说明

首页输入 key 跳转

或浏览器输入 容器 ip:端口/config?key=密钥 进入配置总览页

例如：192.168.11.11:7678/config?key=admin

*V3.1 版本后无特殊不需手动重启容器

1、全局配置（点击主程序卡片）：



日志等级：调试选 DEBUG，平时选 INFO

保留天数：日志每天 0 点轮换，填 0 只保留当天日志

巴法断线重连推送：如遇网络波动导致断连，将在网络恢复重连时发送消息

推送渠道：可以启用一个，也可以同时启用多个

保存即可生效

2、设备配置（点击设备卡片）：

(1) 服务：



启用：该设备服务总开关

api 别名：只影响网络唤醒、关机、ping 的 api 调用，方便记忆，只允许英文

例：设备 id 为 device01，别名设置为 windows，则 api 接口为 http://192.168.x.x:7678/wol/device01?key=密钥，或，http://192.168.x.x:7678/wol/windows?key=密钥

(2) 巴法：可选择接入或不接入，接入可用巴法小程序、app 和米家共同控制，不接入可选择将端口转发或反向代理访问网页实现远程开关

接入部分见‘接入巴法云并接入米家’部分，填入参数



(3) 设备：名称可自由更改。网络唤醒见‘配置网络唤醒’部分，远程关机见‘配置远程关机’部分。填入参数，设备 ip 需要设置为静态 ip, ping 时长建议设置为 60s 至 120s, 不宜过短。超时时长默认 2 秒，指令发送超过该时长则判定发送失败，低配置 arm 设备如遇关机指令报错，可适当延长该时长。

The screenshot shows the 'Device' configuration interface. It includes fields for '设备名称' (Device Name) set to '电脑'; '设备ip地址' (Device IP Address) set to '192.168.2.11'; '启用网络唤醒' (Enable Network Wakeup) with a checked checkbox and a link to test; '唤醒方法' (Wake-up Method) set to '内置网络唤醒' (Built-in Network Wakeup); '设备网卡mac地址' (Device Network Adapter MAC Address) set to 'd0:11:e5:66:21:11'; '目标地址' (Target Address) set to '定向广播(192.1...)' (Broadcast); '端口' (Port) set to '9'; a link to download the '唤醒魔包' (Wake-up Magic Package) for testing; '启用远程关机' (Enable Remote Shutdown) with a checked checkbox and a link to test; '关机方法' (Shutdown Method) set to '关机账户[Wind...' (Shutdown Account); '账户' (Account) set to 'sd'; '密码' (Password) set to '*****'; '延迟关机时长(秒)' (Shutdown Delay Time (Seconds)) set to '30'; '指令超时时长(秒)' (Command Timeout Time (Seconds)) set to '2'; '启用ping检测' (Enable ping detection) with a checked checkbox and a link to test; 'ping方法' (ping Method) set to '内置ping' (Built-in ping); and 'ping间隔时长(秒)' (ping Interval Time (Seconds)) set to '60'.

(4) 消息推送：该设备的消息推送开关

(5) 定时任务: id 不可更改; 名称可自由更改, 如“每周工作日晚 7 点开机”; 类型单次为选择日期时间, 一次性任务, 循环为 cron 表达式, 可借助内置网页工具或 AI 快速生成表达式; 下次执行时间将在保存并重启服务后更新; 提前推送提醒需配置好主程序通知渠道, 并同时打开主程序和单个设备的推送消息开关。

定时:

启用定时任务:

id: 1
名称: 示例任务1 [?](#)

启用:

类型: 循环

cron表达式: 0 9 * * 1-5 [?](#)
*快速生成cron表达式: 本地版 在线版

操作: 关机

下次执行时间: 无

提前推送提醒: [?](#)

提前时间(分钟): 5

[删除](#)

[添加任务](#)

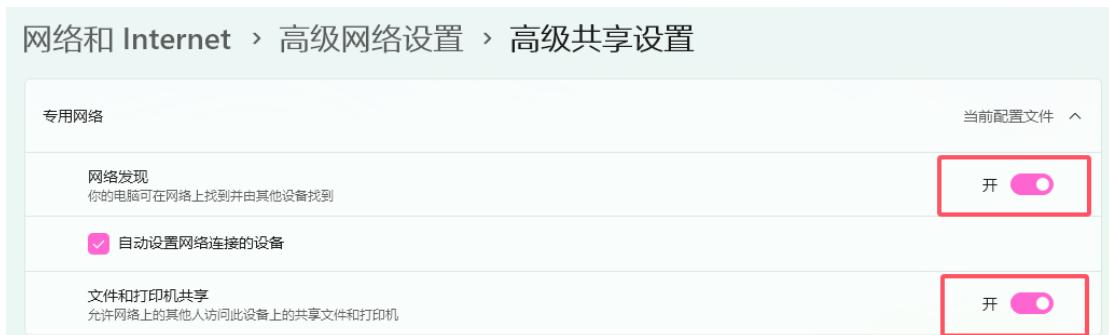
*每次保存完配置需要点击重启生效
*不需要开启的功能可以不填相关参数
*调试唤醒和关机可以点击配置页执行开机/关机/ping 或通过米家
*建议将参数保存一份到本地, 防止日后用到

四、前置设置

1、修改网络为【专用网络】



2、启用网络发现



3、关闭或配置专用网络防火墙

*调试阶段建议先关掉，调试没问题了打开，如果打开了不影响关机和 ping，则不用再配置

*如果打开后功能失效了：

- a. 关掉防火墙，不用进一步配置了
- b. 如果觉得关闭防火墙不安全，打开防火墙，跳至②配置防火墙

① 关闭防火墙

点击上上图中的配置防火墙和安全设置，打开 windows 安全中心

(1) 防火墙和网络保护

哪些人和哪些内容可以访问你的网络。

域网络

防火墙已打开。

专用网络 (使用中)

防火墙已打开。

Microsoft Defender 防火墙

在专用网络上时，有助于保护设备。

 专用防火墙已关闭。你的设备可能易受攻击。



关

② 配置防火墙

Ping:

(1) 防火墙和网络保护

应用和浏览器控制

设备安全性

设备性能和运行状况

家庭选项

保护历史记录

专用网络 (使用中)

防火墙已打开。

公用网络

防火墙已打开。

允许应用通过防火墙

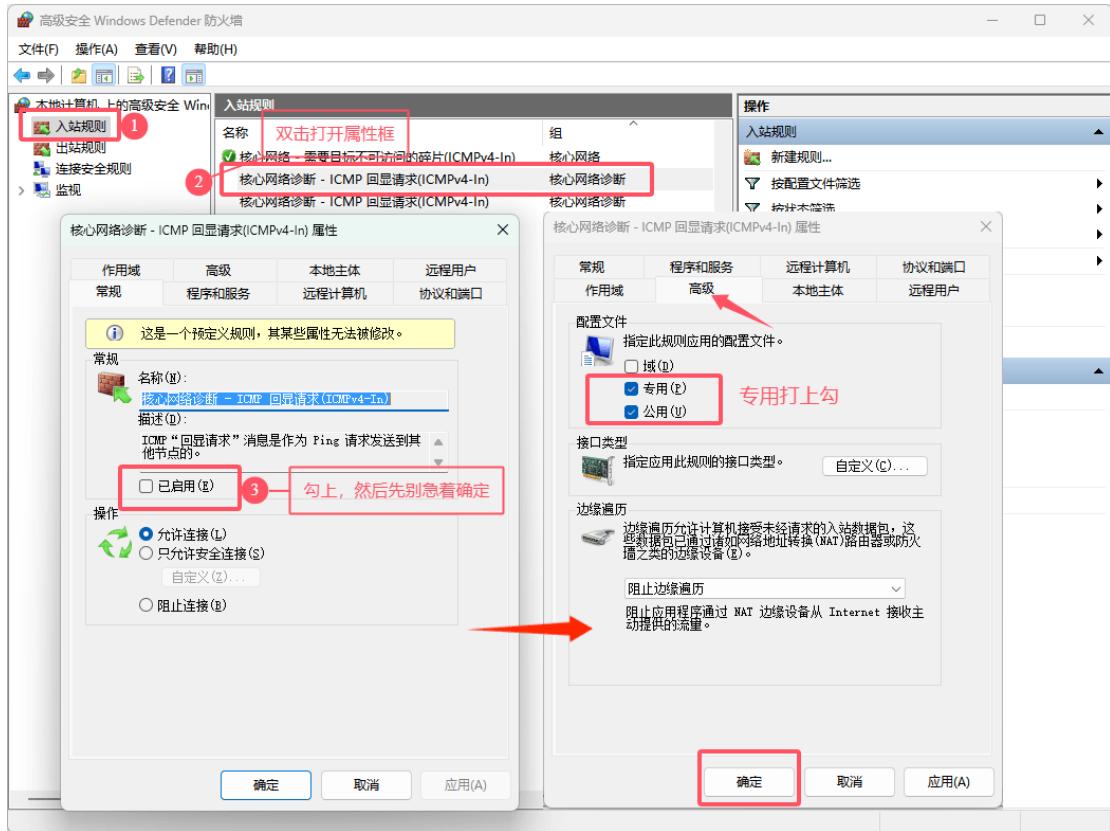
网络和 Internet 疑难解答程序

防火墙通知设置

高级设置

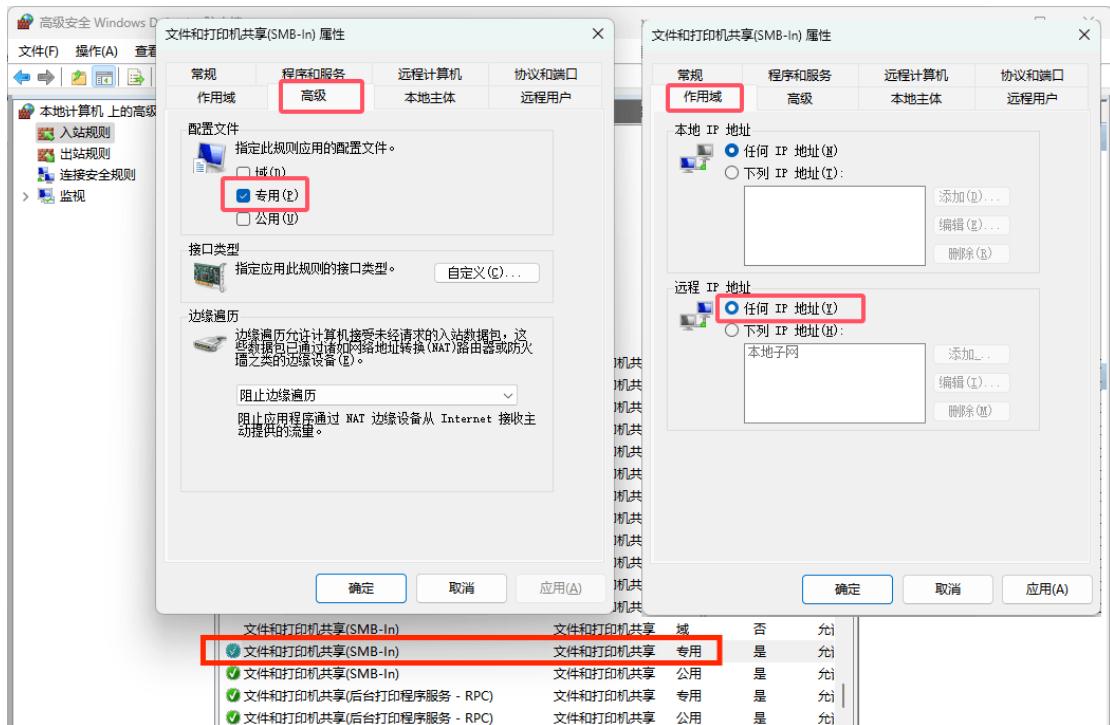
将防火墙还原为默认设置

找到核心网络诊断，按图设置，顺便检查一下作用域，本地和远程都设置为任何 ip，设置完可以 ping 通

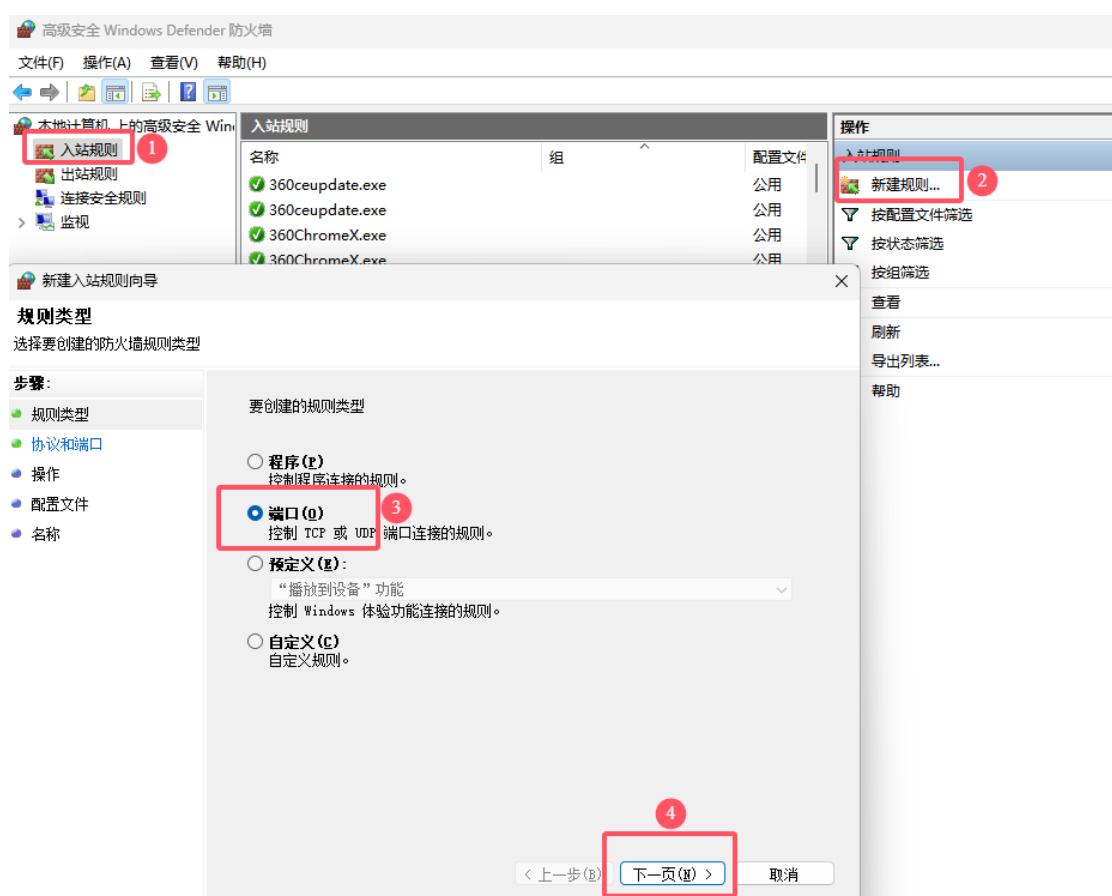
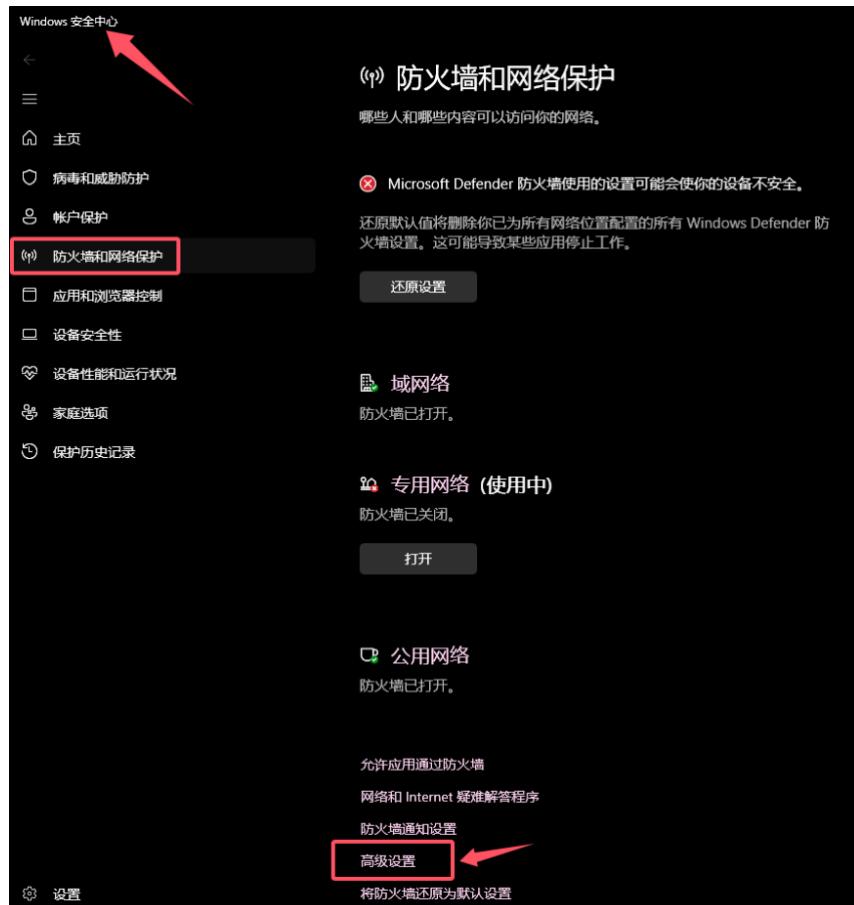


关机：

关机账户方法：



配套软件方法：





填写软件监听的端口号





名称随便写，出站规则也同样配置

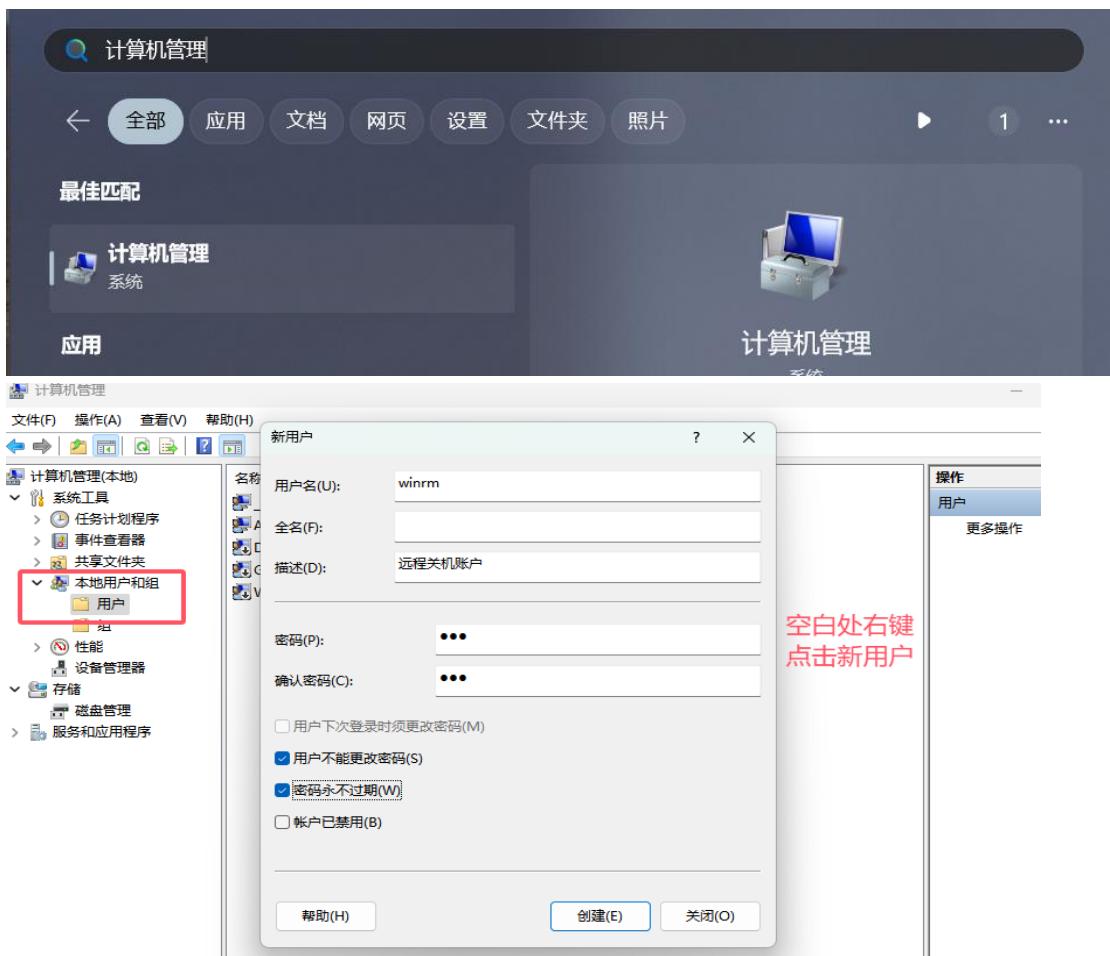
五、配置远程关机

如果系统是家庭版或精简版等，找不到组件（比如组策略），或是配置好后怎么都不成功的，可使用→2、配套软件方法。

1、关机账户方法：

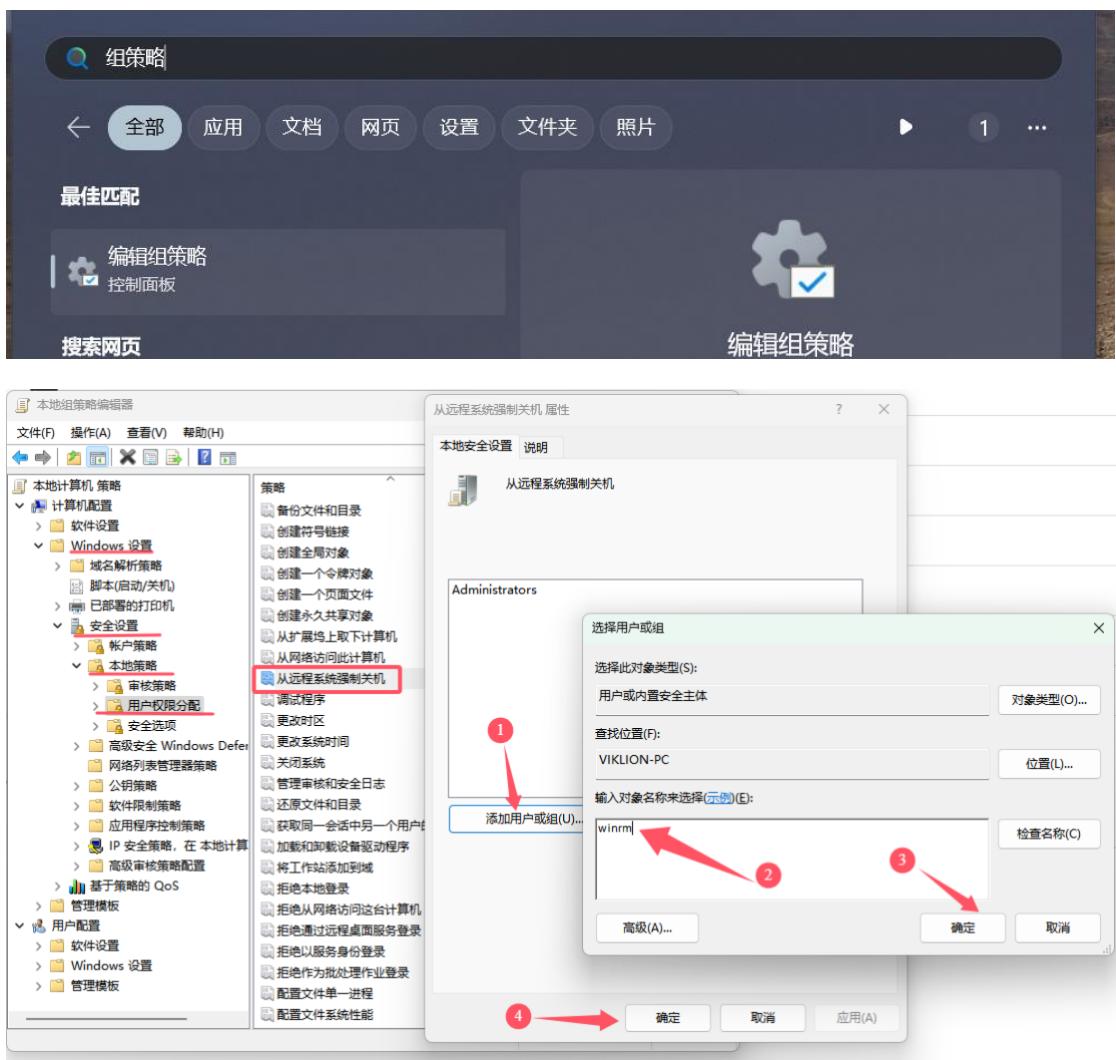
①建立关机专用账户

打开计算机管理-系统工具-本地用户和组-用户-新建用户，用户名随意，比如：winrm，并设置密码，按下图勾选：

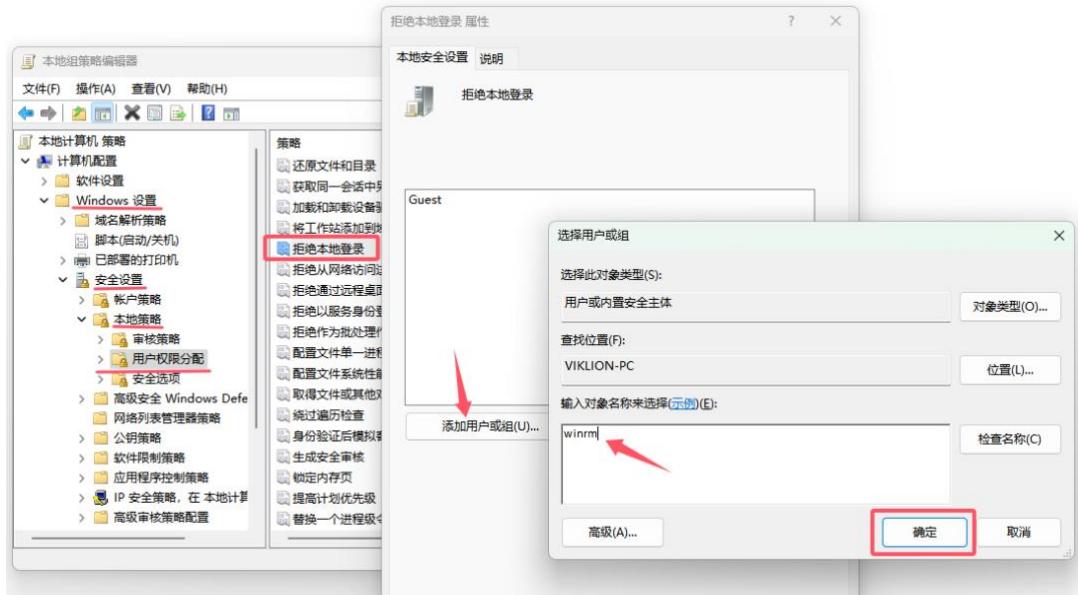


②配置组策略

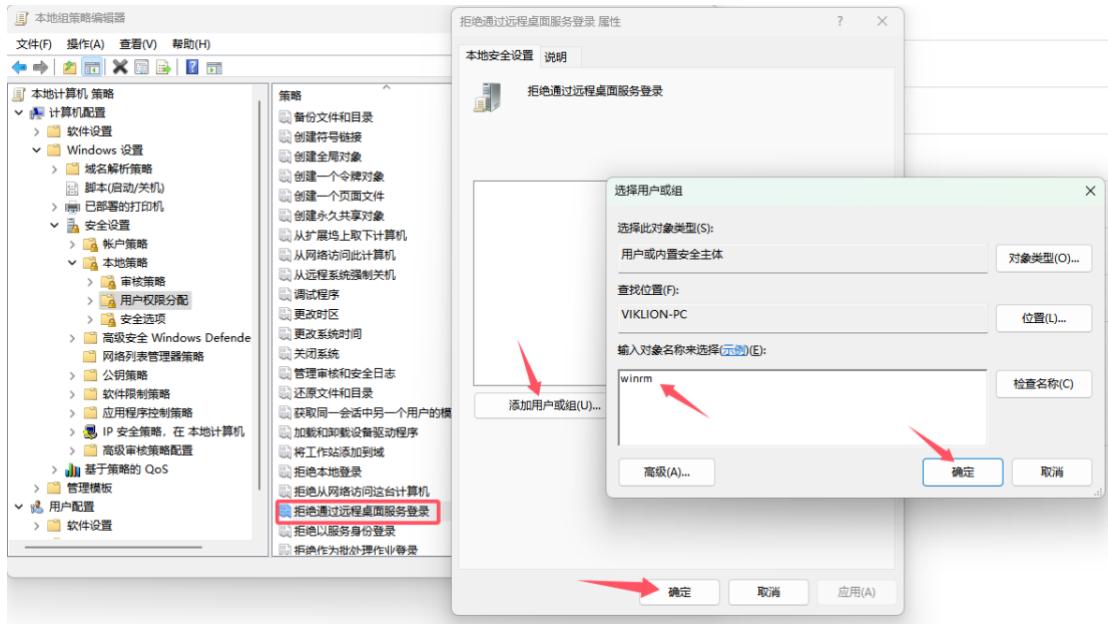
A、计算机配置-windows 设置-安全设置-本地策略-用户权限分配：从远程系统强制关机，添加刚刚创建的用户



B、拒绝本地登录，添加刚刚创建的用户

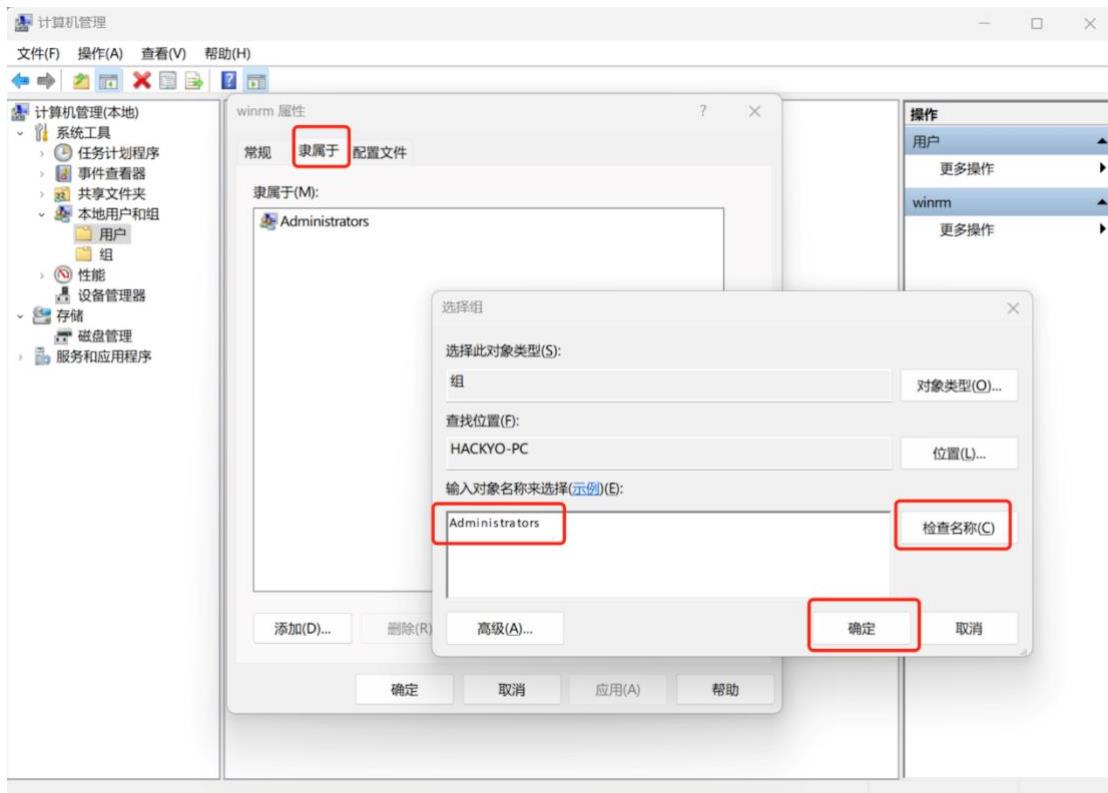


C、拒绝通过远程桌面登录，添加刚刚创建的用户



配置完成

*如果无法收到关机指令，尝试将关机用户加入到管理员组，一般不需要添加
计算机管理-本地用户和组-用户-右键关机用户-属性-隶属于-添加输入：
Administrators



2、配套软件方法：

①配置页面下载 exe 程序



*如果想自行编译 exe 可以去 github 下载 PCshutdown 文件夹

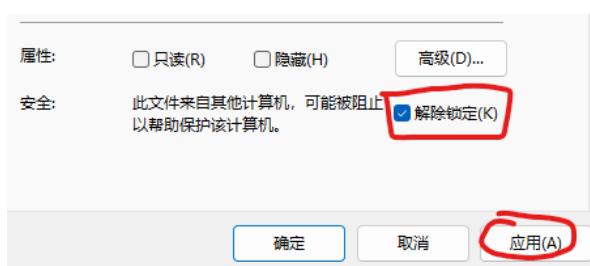
需要下载安装 WinLibs MinGW-w64，并配置环境变量

编译：g++ PCshutdown.cpp resource.o -o PCshutdown.exe -municode -mwindows -static -lws2_32 -lpowrprof

②移动至你想放的文件夹内

③运行

如果提示阻止，右键软件→属性→勾选解除锁定→应用→确定，再次打开



软件界面：



如果勾选开机自启无效，则可通过<任务计划程序>创建开机自启任务实现

3、自定义指令方法：

可用于关闭 Linux、MacOS 设备

*可访问快速生成指令页面一键生成：



Shell 指令，默认只支持 sshpass 和 curl，可手动修改 yaml 文件配置

1、关闭 Linux 设备，sshpass 指令：

```
sshpass -p "用户密码" ssh -o "StrictHostKeyChecking=no" -p 端口  
用户名@ip "echo 'sudo 密码' | sudo -S poweroff"  
修改上面标红字段
```

示例：立刻关机：

```
sshpass -p "abc123" ssh -o "StrictHostKeyChecking=no" -p 22  
username@192.168.1.12 "echo 'abc123' | sudo -S poweroff"
```

示例：延迟关机，修改 sleep 后秒数：

```
sshpass -p "abc123" ssh -o "StrictHostKeyChecking=no" -p 22  
username@192.168.1.12 "echo 'abc123' | sudo -S sleep 30 && echo  
'abc123' | sudo -S poweroff"
```

2、curl

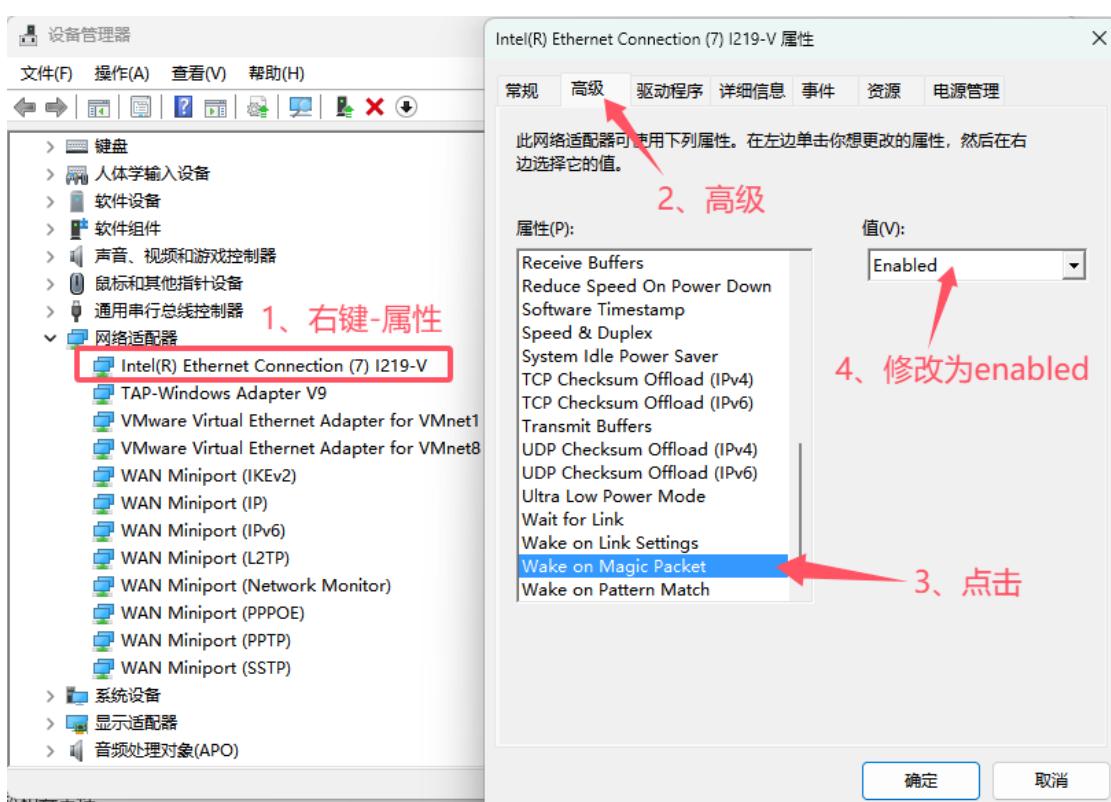
如果使用类似 Remote Shutdown Manager 软件 (<https://github.com/karpathy/remote-shutdown-pc>)，可以使用该项

六、配置网络唤醒

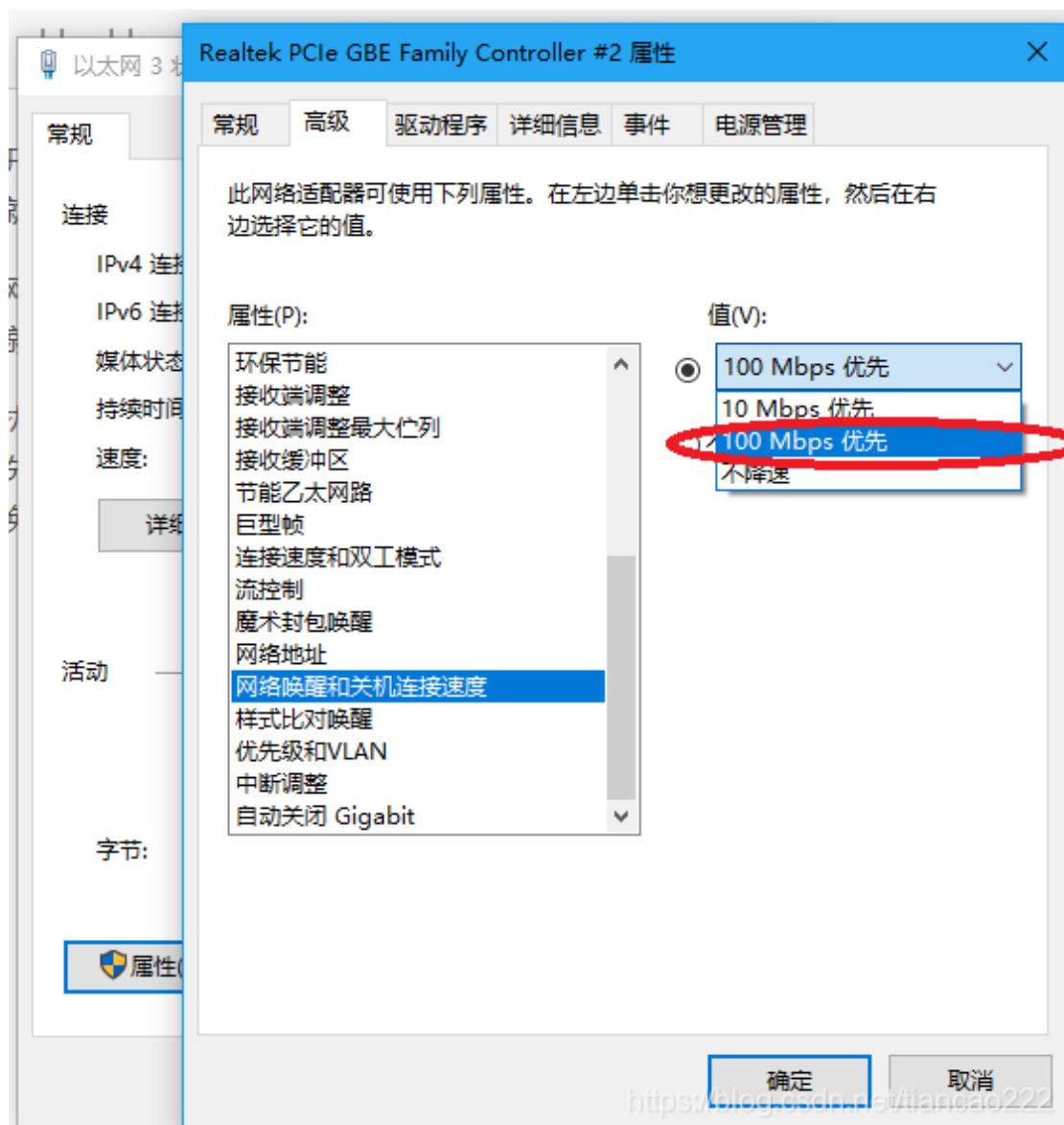
主板 bios 设置各厂商主板方法都不相同，自行查阅主板型号的设置方法，一般可能在高级-选项名称中带有 wake up, wake on 之类的

参考：微星主板，官网-[主板网络唤醒设置](#)；华硕主板，官网-[\[主板\]BIOS 如何设置启用网络唤醒\(WOL: Wake On Lan\)功能 | 官方支持 | ASUS 中国](#)

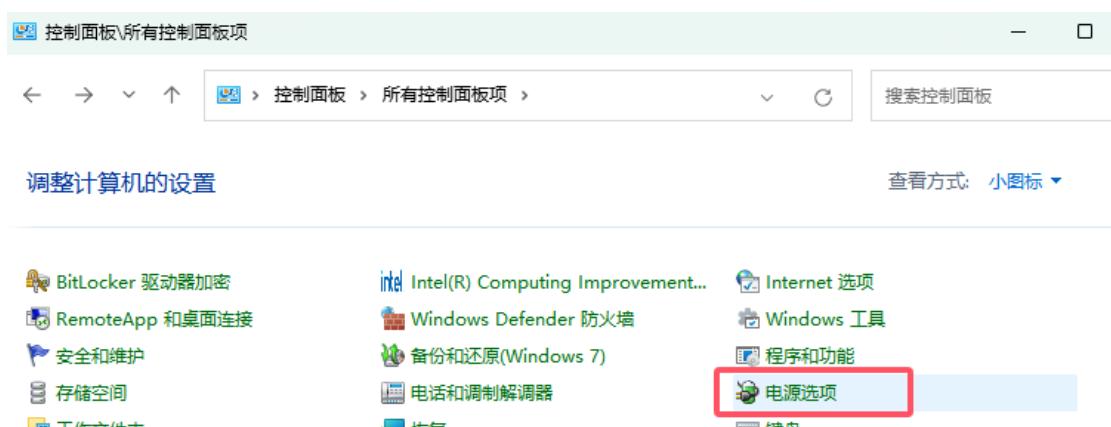
1、配置：

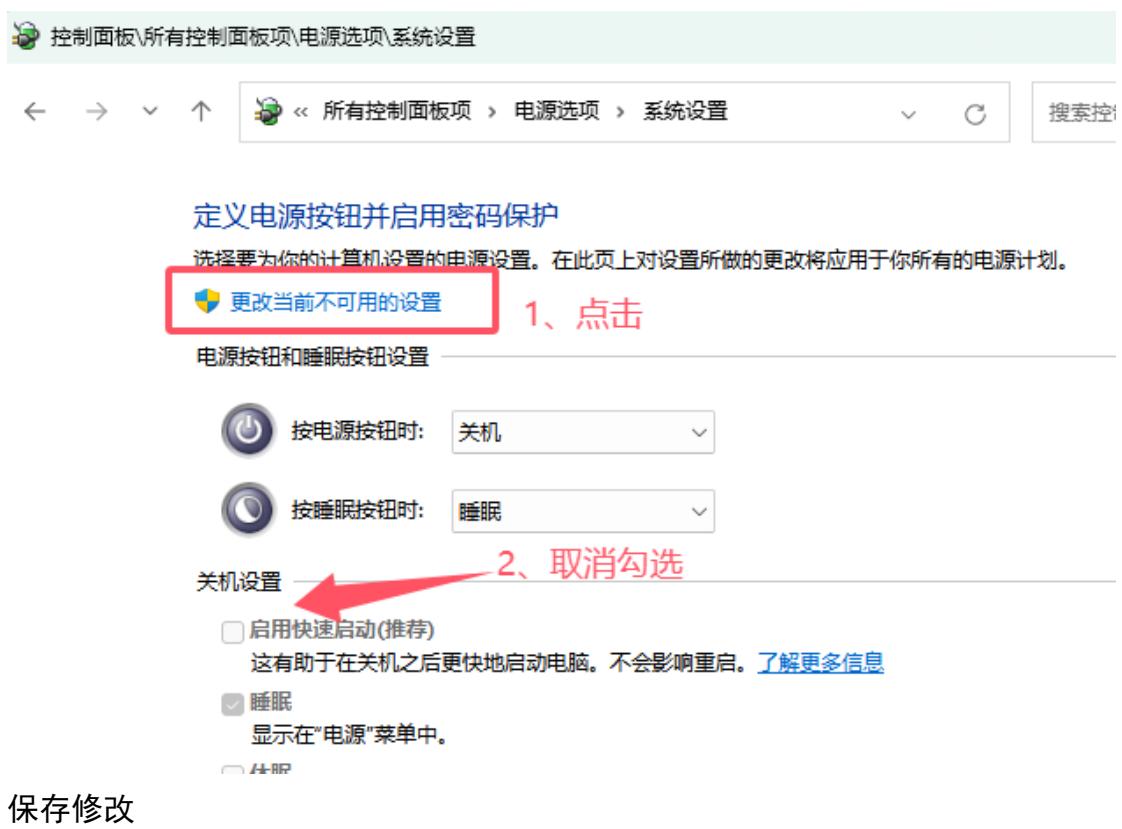


如果有此选项，修改为 100Mbps 优先



取消快速启动：





保存修改

记录 mac 地址和 ip 地址:



2、调试

如无法成功网络唤醒，使用软件可缩小排除对象范围：

① 软件能接收到唤醒包，排除 docker、路由器等因素，主要解决电脑 bios、网卡、关闭快速启动等设置问题

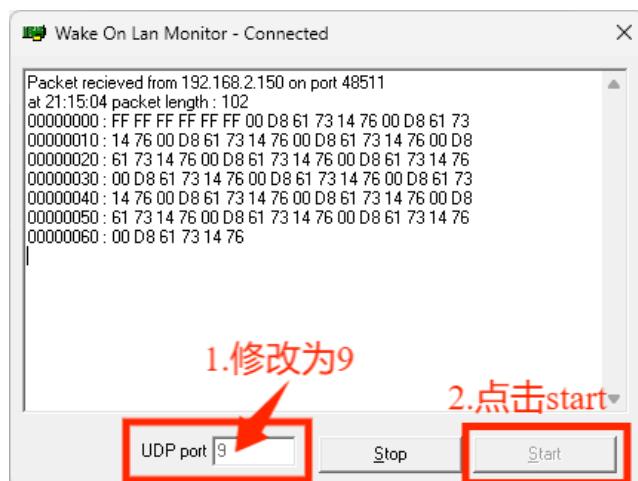
② 软件无法接收到唤醒包，优先解决 docker、网络、路由器等问题

*重要：请务必暂时关闭专用网络防火墙，否则软件接收不到任何内容

下载 WakeOnLanMonitor.exe



打开软件，如果弹出允许连接网络点击允许，端口号修改为 9，点击 start



启动后点击执行开机，正常则可以接收到内容如上图

*目标地址默认（全局广播）即可，如果接收不到可以尝试另外 2 个。如果改为设备 ip 后可以接收到，并且能够网络唤醒，请注意可能会遇到刚关机一段时间内可以唤醒，时间久了无法唤醒的情况。如遇此情况，请尝试路由器中将电脑 ip 和 mac 地址 ARP 绑定，尝试更新网卡驱动，并且确认网卡禁用节电模式，bios 中禁用如深度休眠、ERP 等配置。在进行尝试之前，备份相关设置、驱动，记录修改的操作，防止出现问题无法回退。

七、接入巴法云并接入米家

打开网页 <https://cloud.bemfa.com/>

注册并登录后，进入控制台

记录下巴法云私钥：



新建主题，英文，最后必须以 001 结尾，例如：mypc001, tony001



点击进入主题，修改昵称为自己想要的名称，比如：电脑，计算机，托尼



TCP设备云: mypc001

昵称: 插座

计算机

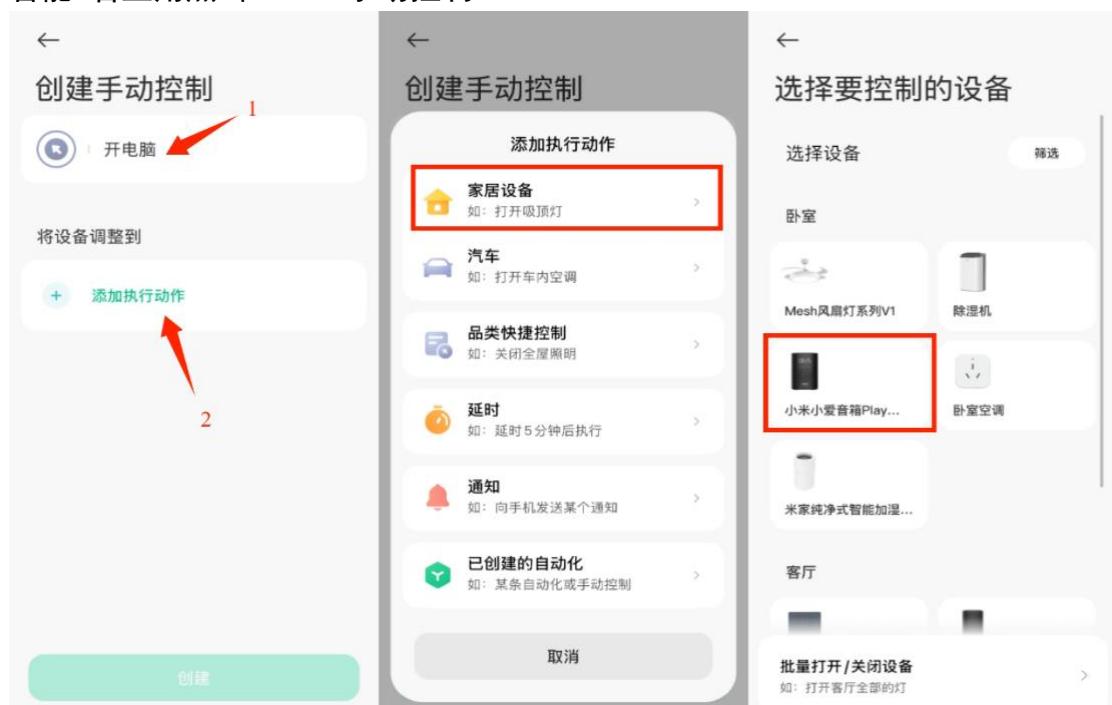
更新昵称

将参数填入配置中

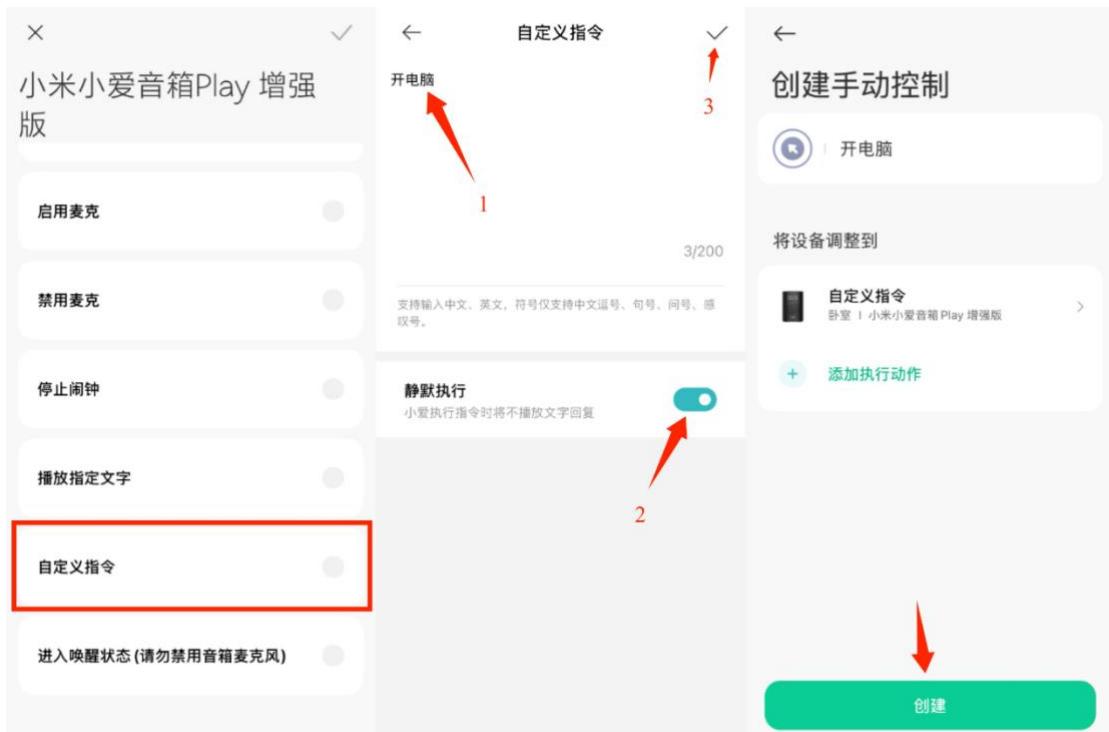
手机打开米家，我的-连接其他平台-添加-搜索巴法，登录后同步设备
同步完设备不会显示在米家中，但是可以对小爱同学说：打开/关闭 XXX (XXX
为上面设置的昵称)，可以控制

家里有小爱音箱的话，添加手动控制，可以通过米家点击标签或小组件快速
控制：

智能-右上角点击“+”-手动控制：



↑上图中的“开电脑”，“电脑”可以是你想要的任何名称



↑上图中的“开电脑”，“电脑”为巴法云中的设备昵称

通过点击卡片或小组件快捷控制



八、接入 homeassistant

需要打开 ping 功能，需要修改 ha 的配置文件（configuration.yaml）和自动化配置文件（automations.yaml）

请在开始前先进行备份，请在开始前先进行备份，请在开始前先进行备份。

*复制代码请访问配置页顶部点击跳转代码页，或访问/code，方便复制

1、File editor 编辑 configuration.yaml

*如果没有安装 File editor，设置-加载项-右下角加载项商店-找到 File editor 安装。或者使用其他方法编辑 configuration.yaml

添加以下内容（请认真检查缩进和冒号后需要有一个空格）：

将标红处替换为你的数据，参考图上标的 2 处也可改，其余不用改，保存

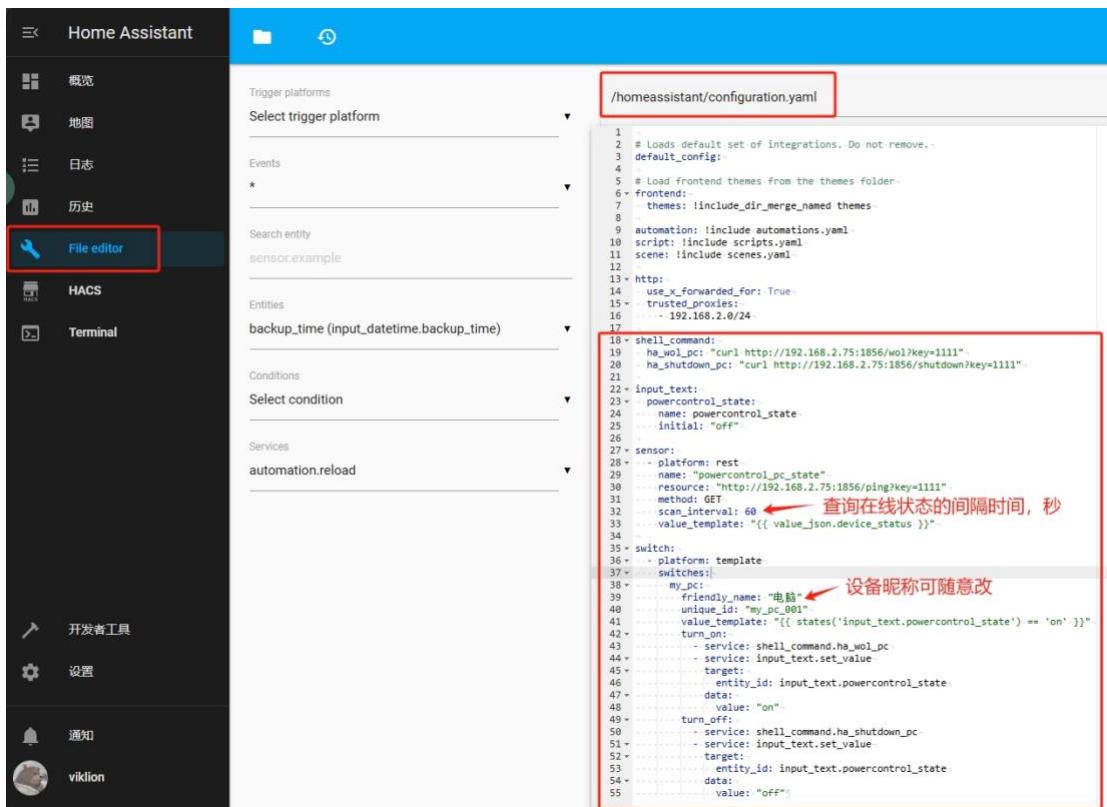
```
shell_command:  
    ha_wol_pc: "curl http://192.168.2.75:1856/wol/设备 id 或别名?key=1111"  
    ha_shutdown_pc: "curl http://192.168.2.75:1856/shutdown/设备 id  
或别名?key=1111"  
  
input_text:  
    powercontrol_state:  
        name: powercontrol_state  
        initial: "off"  
  
sensor:  
    - platform: rest  
        name: "powercontrol_pc_state"  
        resource: "http://192.168.2.75:1856/ping/设备 id 或别名?key=11  
11"  
        method: GET  
        scan_interval: 60  
        value_template: "{{ value_json.device_status }}"  
  
template:  
    - switch:  
        - default_entity_id: switch.my_pc  
            name: 电脑  
            unique_id: my_pc_001  
            icon: mdi:desktop-tower-monitor  
            state: "{{ states('input_text.powercontrol_state') == 'on  
' }}"
```

```

turn_on:
  - action: shell_command.ha_wol_pc
  - action: input_text.set_value
    target:
      entity_id:
        - input_text.powercontrol_state
    data:
      value: "on"
turn_off:
  - action: shell_command.ha_shutdown_pc
  - action: input_text.set_value
    target:
      entity_id:
        - input_text.powercontrol_state
    data:
      value: "off"

```

参考图（图中代码如有部分不同按代码页中为准）：



*请检查原配置文件中是否已经有 shell_command:、input_text:、sensor:、template:。如果有，你需要在存在的条目下添加内容，并且不要把上述这些复制，注意缩进。

2、编辑 automations.yaml

添加以下内容（请认真检查缩进和冒号后需要有一个空格）：
可修改内容见参考图，修改完保存

```
- id: '20250108'  
  alias: 更新电脑在线状态  
  triggers:  
    - trigger: time_pattern  
      minutes: "/1"  
  conditions:  
    - condition: template  
      value_template:  
        "{{ states('sensor.powercontrol_pc_state') != states('input_text.powercontrol_state') }}"  
  actions:  
    - action: input_text.set_value  
      target:  
        entity_id: input_text.powercontrol_state  
      data:  
        value: "{{ states('sensor.powercontrol_pc_state') }}"
```

参考图（图中代码如有部分不同按代码页中为准）：

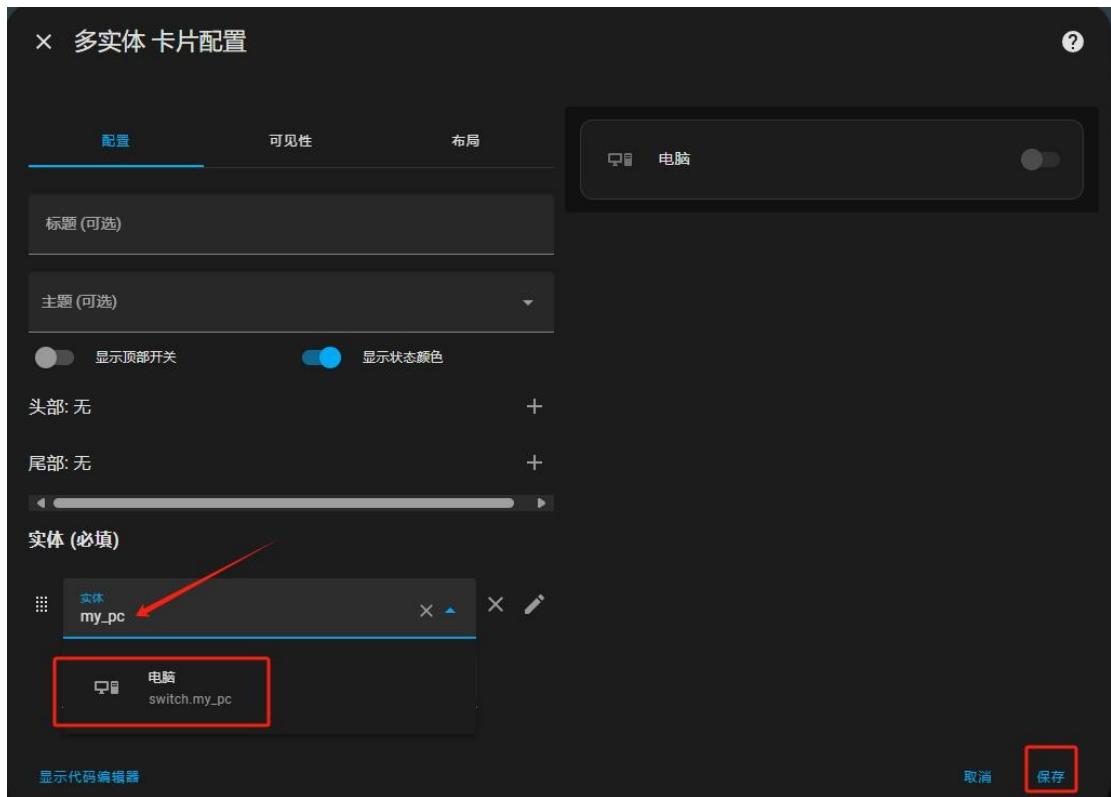
```
54  
55 - id: '20250108' ← id, 数字, 可随意改  
56   alias: 更新电脑在线状态 ← 自动化显示名称, 可随意改  
57   trigger:  
58     - platform: time_pattern  
59       minutes: "/2" ← 每隔多久更新状态, 当前为每2分钟, 如需秒, 则修改成 seconds: "/30", 注意冒号后需有一个空格, 数字  
60       1-59  
61     - condition:  
62       - condition: template  
63       - value_template: "{{ states('sensor.powercontrol_pc_state') != states('input_text.powercontrol_state') }}"  
64     - action:  
65       - service: input_text.set_value  
66       target:  
67         entity_id: input_text.powercontrol_state  
68         data:  
69           value: "{{ states('sensor.powercontrol_pc_state') }}"  
70  
71
```

3、添加实体

完成上述文件配置后，设置-右上角三个点-重启 homeassistant-快速重载
进入设置-设备与服务-实体-输入 my_pc，找到实体



4、在面板中添加实体





*开关控制由访问 web 接口实现，不经过办法

*自动更新开关（在线）状态，受 2 个时间参数影响，一个是 sensor 中的 scan_interval: 60，这是获取设备在线状态的间隔时间；第二个是自动化中的 minutes: "/2"，这是更新开关状态的间隔时间，该时间过短会造成关闭开关后马上更新为打开状态，反之同理。

*homeassistant 自带可以实现开关电脑的插件，本教程为使用本 docker 容器接入 homeassistant

九、ios 快捷指令

添加完可以长按-分享-添加到主屏幕，快捷操作



十、yaml 配置文件详解

1、main.yaml（主程序配置文件）

```
log:
  level: INFO      #日志等级
  keep_days: 7     #日志保留天数, 0 为只保留当天的日志
message:
  enabled: False    #是否启用消息推送
  bemfa_reconnect: True   #是否推送巴法重连消息
  ServerChan_turbo:    #Server 酱 turbo 推送参数配置
    enabled: False    #是否启用
    SendKey: YOUR SENDKEY    #Server 酱 turbo 的 key
    channel: 9       #消息通道, 默认为 9 (微信服务号)
  ServerChan3:      #Server 酱 3 推送参数配置
    enabled: False    #是否启用
    SendKey: YOUR SENDKEY    #Server 酱 3 的 key
  Qmsg:      #qq 消息推送参数配置
    enabled: False    #是否启用
    key: YOUR KEY     #Qmsg 酱的 key
    qq: 123456789    #QQ 号
  WeChat_webhook:   #企业微信机器人消息推送参数配置
    enabled: False    #是否启用
    url: https://qyapi.weixin.qq.com/cgi-bin/webhook/send?key=xxxx
  Gotify:      #Gotify 消息推送参数配置
    enabled: False    #是否启用
    url: http(s)://ip:port    #自建服务地址, 勿省略 http(s)://
    token: YOUR APPS TOKEN    #APPS 中的 token
  PushPlus:      #PushPlus 消息推送参数配置
    enabled: False    #是否启用
    token: YOUR TOKEN    #PushPlus 的 token
    channel: wechat    #消息通道, 默认为 wechat (微信服务号)
  Bark:      #Bark 消息推送参数配置
    enabled: False    #是否启用
    url: YOUR BARK URL    #服务地址, 官方或自建, 勿省略 http(s)://
    key: YOUR KEY     #服务的 Key
  WeChat_app:    #企业微信应用消息推送参数配置
    enabled: False    #是否启用
    corpId: YOUR CORPID    #企业微信-企业 ID
    agentId: YOUR AGENTID    #企业微信-应用 AgentId
    secret: YOUR SECRET    #企业微信-应用 Secret
```

2、device01.yaml（设备配置文件）

```
main:
  enabled: True     #设备服务总开关
```

```
alias:      #api 别名
bemfa:     #巴法参数配置
    enabled: False      #是否连接巴法平台
    uid: aaabbccdddeeffggg    #巴法 uid
    topic: mypc001      #巴法设备主题
devices:   #设备参数配置
    name: 电脑      #设备昵称
    ip: 192.168.100.100    #设备局域网 ip 地址
    wol:      #网络唤醒参数配置
        enabled: True      #是否启用网络唤醒
        method:      #网络唤醒方法
            wakeonlan: True      #内置网络唤醒方法
            shell: False      #自定义指令唤醒
            mac: 00-11-22-33-44-55    #设备网卡 mac 地址 xx-xx 格式或 xx:xx 格式
            destination: broadcast_ip_global    #网络唤醒发送地址, broadcast_ip_global: 全局广播(255.255.255.255), broadcast_ip_direct: 定向广播(设备所在网段的广播地址), device_ip: 设备 ip 地址
            port: 9      #网络唤醒端口, 一般无需修改, 默认: 9
            interface: default      #指定网卡发送, default: 默认网卡 ip, 多网卡下可修改成指定 ip 地址
            shell_script: yours痼      #自定义指令内容
    shutdown:   #关机参数配置
        enabled: True      #是否启用关机
        method:      #关机方法
            netrpc: True      #通过关机账户
            udp: False      #通过配套软件
            shell: False      #通过自定义指令
            account: youraccount    #关机账户-账户
            password: yourpassword    #关机账户-密码
            shell_script: yours痼      #自定义指令
            udp_port: 17678      #配套软件监听的端口, 默认: 17678
            win_cmd: shutdown      #shutdown: 关机, reboot: 重启, sleep: 睡眠, hibernate: 休眠
time: 60      #延迟关机时长, 1 为立刻关机
timeout: 2      #指令超时时长, 默认: 2
ping:      #ping 查询设备在线状态参数配置
    enabled: True      #是否启用 ping 查询设备在线状态
    time: 60      #查询间隔时长
    method:      #ping 方法
        pcping: True      #内置 ping 方法
        shell: False      #自定义指令 ping
        shell_script: yours痼      #自定义指令内容
        on_keyword: 'on'      #判断设备在线关键字
        off_keyword: 'off'      #判断设备离线关键字
```

```
message:  
    enabled: False      #该设备的消息推送开关  
schedule:      #定时任务参数配置  
    enabled: False      #是否启用定时任务  
    plans:      #定时任务列表  
        - id: '1'      #任务 id  
        enabled: False      #是否启用该任务  
        name: '示例任务 1'      #任务名称  
        type: cron      #任务类型, cron: 循环任务, datetime: 单次任务  
        datetime: '2026-1-1 20:00:00'      #单次任务时间  
        cron: '0 9 * * 1-5'      #循环任务时间, 5 段 cron 表达式  
        action: shutdown      #任务执行操作, wol: 开机, shutdown: 关机  
        remind: False      #是否启用任务执行前提醒  
        advance: 5      #任务提前提醒时间, 单位: 分钟
```

十一、反馈

<https://github.com/viklion/PowerControl>

- 访问容器网页的教程不一定保持最新的，github 里的教程是最新
- 注意映射目录是否存在权限问题，可查看容器运行日志。
- 配置文件损坏/出错，将配置文件删除，重启容器，恢复默认。
- 网络唤醒和远程关机的配置较为繁琐，如果出现问题还需要自行摸索，原因众多，包括杀毒软件防护，查看防护日志，有没有拦截请求，防火墙、路由器、巴法是否使用了代理连接等

by viklion

2026.1.18