

目录

TOP

- [致尊敬的程序员用户](#)
- [通用型人工智能个人代码数据中心介绍](#)
- [产品概述](#)
 - [产品使用](#)
 - [设置IP地址](#)
 - [网路环境设置](#)
 - [samba登录](#)
 - [Mac](#)
 - [Windows](#)
 - [ssh登录](#)
 - [配置无线](#)
- [硬件升级与特色](#)
- [软件升级与特色](#)
 - [配置和安装gitea](#)
 - [通过SSH-port-forward对外提供服务](#)
 - [Raid1-ext4文件系统](#)
 - [DeepSeek-R1-distill大模型服务器热点](#)
 - [基于Rknn分类器的图片分类服务](#)
 - [Samba文件服务器](#)
 - [开发环境](#)
 - [NAS系统OpenMediaVault](#)
 - [Kodi家庭娱乐终端](#)
 - [安装Trae_cn-AI-SSH-agent-console](#)
- [应用场景](#)
- [技术支持](#)
- [售后服务](#)

[回目录](#)

致尊敬的程序员用户

衷心感谢您选择通用型人工智能个人代码数据中心.您的支持是我们不断前进的动力.我们承诺为您提供高质量的产品和专业的服务,帮助您在开发和创新的道路上取得成功

[回目录](#)

通用型人工智能个人代码数据中心介绍

[回目录](#)

1. 产品概述

本产品是一款为程序员用户设计的通用型人工智能个人代码数据. 基于Rock 5B Plus的强大硬件基础,我们进行了深度定制,增加了多项功能和优化,以满足程序员对数据中心的需求.

[回目录](#)

1.1 产品使用

[回目录](#)

设置IP地址

首先按照以下步骤设置家用路由器: 一华为路由器为例

[回目录](#)

1.1.1 网路环境设置

打开您的路由器控制界面，登录找到路由器局域网设置,记下已经使用的ip地址，后面设置固定ip的时候不要用同样的。

有线连接设备	允许上网	网络限速	限速值
2.4G 联网设备			
 82-53-7F-FA-97-2A  ↑ 8 Kbps ↓ 88 Kbps IP 地址: 192.168.2.252 MAC 地址: 82:53:7F:FA:97:2A 在线: 0 天 0 小时 3 分钟			↑ 不限速 ↓ 不限速 
 38-21-87-8F-95-43  ↑ 0 Kbps ↓ 0 Kbps IP 地址: 192.168.2.39 MAC 地址: 38:21:87:8F:95:43 在线: 37 天 10 小时 15 分钟			↑ 不限速 ↓ 不限速 
 MiAiSoundbox-L06A  ↑ 0 Kbps ↓ 0 Kbps IP 地址: 192.168.2.202 MAC 地址: 8C:53:C3:BE:84:09 在线: 37 天 10 小时 15 分钟			↑ 不限速 ↓ 不限速 
 ESP_27948C  ↑ 0 Kbps ↓ 0 Kbps IP 地址: 192.168.2.243 MAC 地址: 28:6C:07:27:94:8C 在线: 37 天 10 小时 15 分钟			↑ 不限速 ↓ 不限速 

找到路由器网络控制界面,如下图:



找到静态ip绑定,如下图:

静态IP地址绑定列表					
序号	主机	MAC 地址	IP 地址	启用	操作
1	deepseek	A8:E2:91:43:83:90	192.168.2.240	☒	

在硬件的正面有贴条上有wifi和有线网卡的地址,把有线和wifi网卡的MAC地址绑定为单独的静态ip

添加静态IP地址绑定

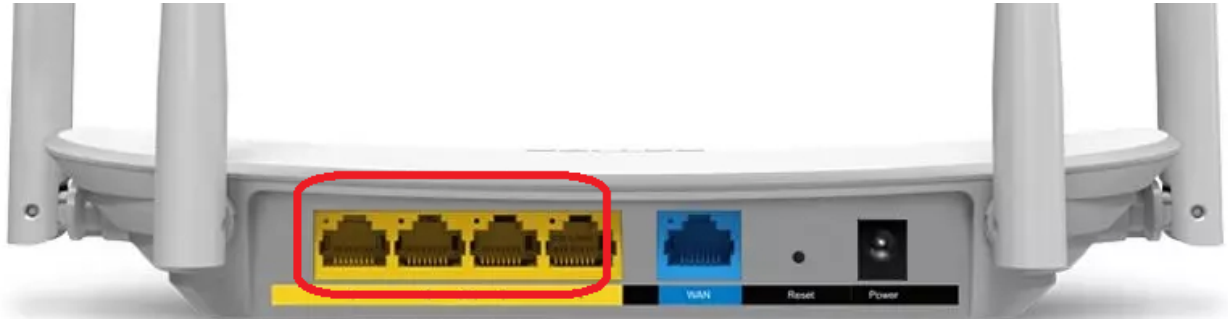
主机

MAC 地址

MAC地址格式: XX-XX-XX-XX-XX-XX或XX:XX:XX:XX:XX:XX

IP 地址

用随机器附赠的网线将本机和路由器的LAN端口相连。



电脑常用知识

[回目录](#)

1.1.2 samba 登录

[回目录](#)

Mac

通过Finder连接Samba服务器

打开Finder:

启动Finder应用程序。

连接到服务器:

在Finder的菜单栏中选择“前往”>“连接服务器”(或直接按 **Command + K** 快捷键)。

在弹出的对话框中, 输入Samba服务器的地址, 例如: **smb://192.168.1.100**。

输入凭据:

如果服务器需要认证, 请输入用户名和密码。

点击“连接”按钮完成连接。

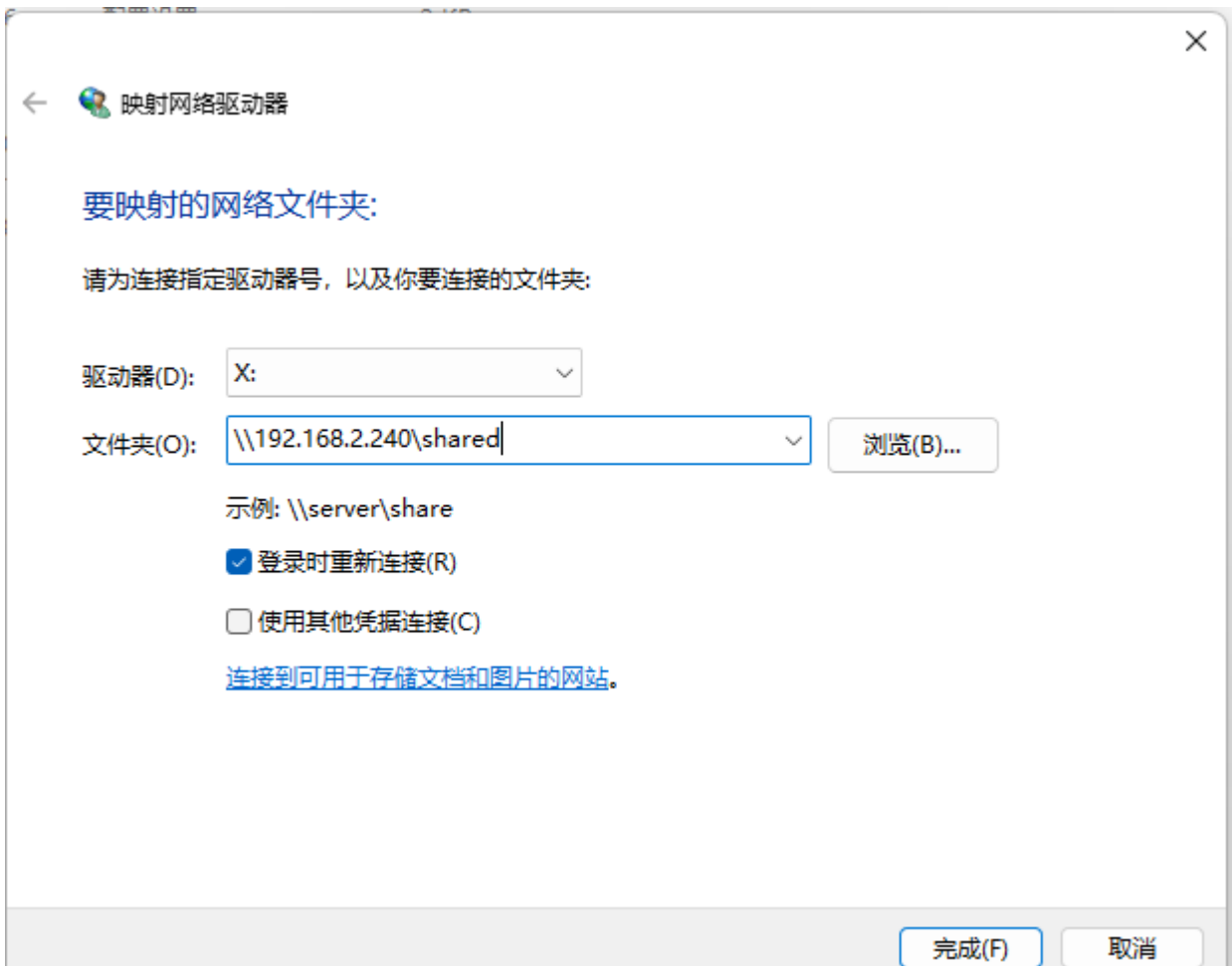
访问共享文件夹:

成功连接后, Finder窗口会显示Samba服务器上的共享文件夹, 您可以像访问本地文件一样操作。

回目录

Windows

— 点击我的电脑 (此电脑) -> 选择‘映射网络驱动器...’如下图



— 输入: 是设备的IP 地址

```
...  
//<ip>/shared  
...
```

— 在提示输入密码时候输入： 用户名： armbian 密码： armbian

 Windows 安全中心

输入网络凭据

输入你的凭据以连接到:192.168.2.240

用户名

用户名

密码

密码

☐

记住我的凭据

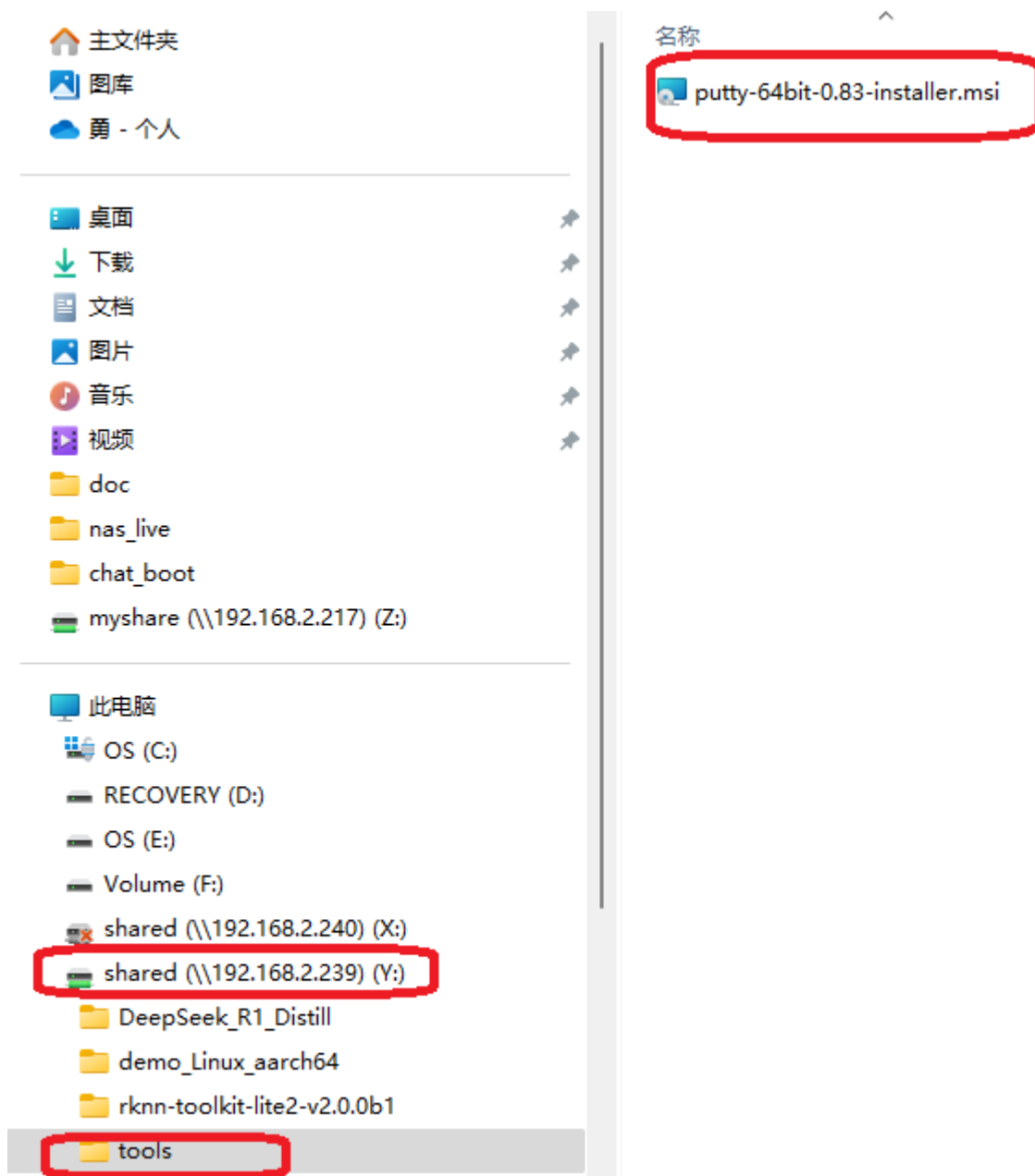
确定

取消

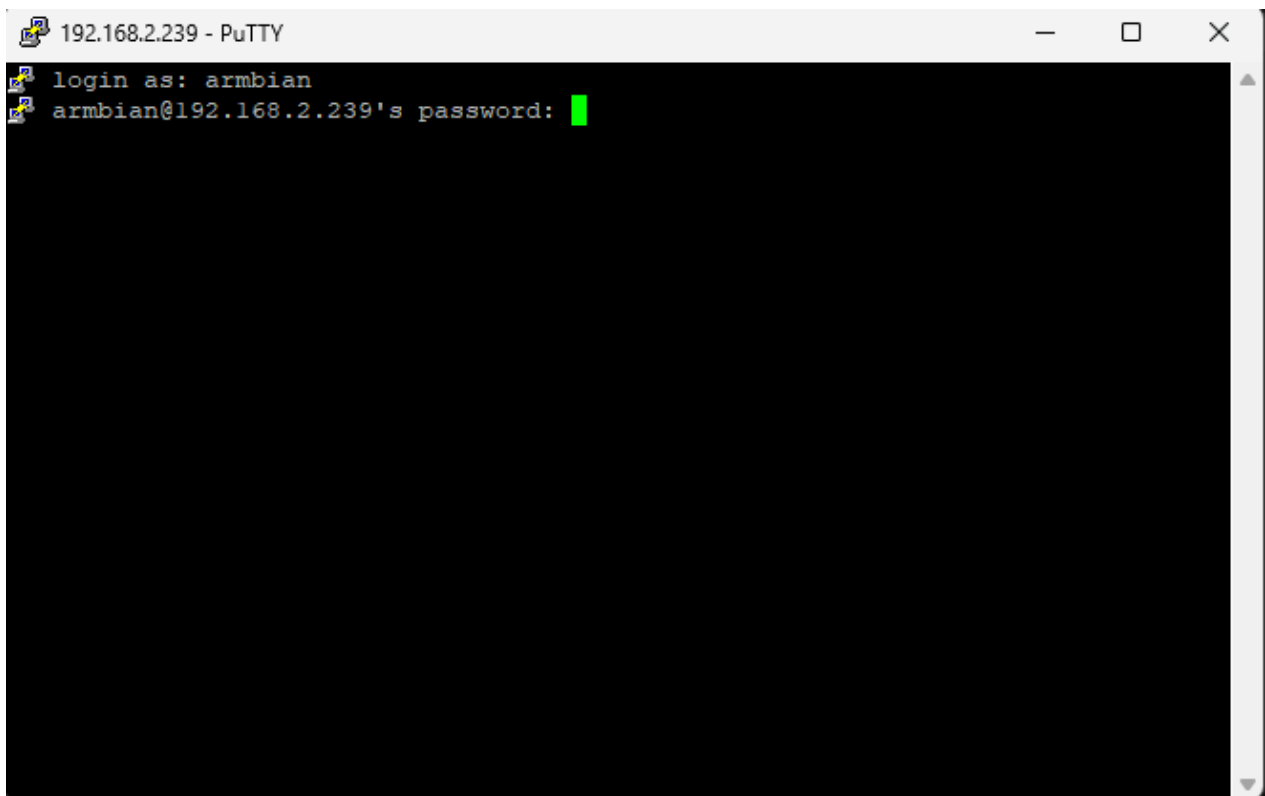
[回目录](#)

1.1.3 ssh 登录

在映射完网络服务器后， 打开映射的盘符可以看到: tools 目录里面有ssh client 安装文件
“putty-64bit-0.83-installer.msi”



双击安装后，通过ssh 登录机器实现程序控制， 如下图:



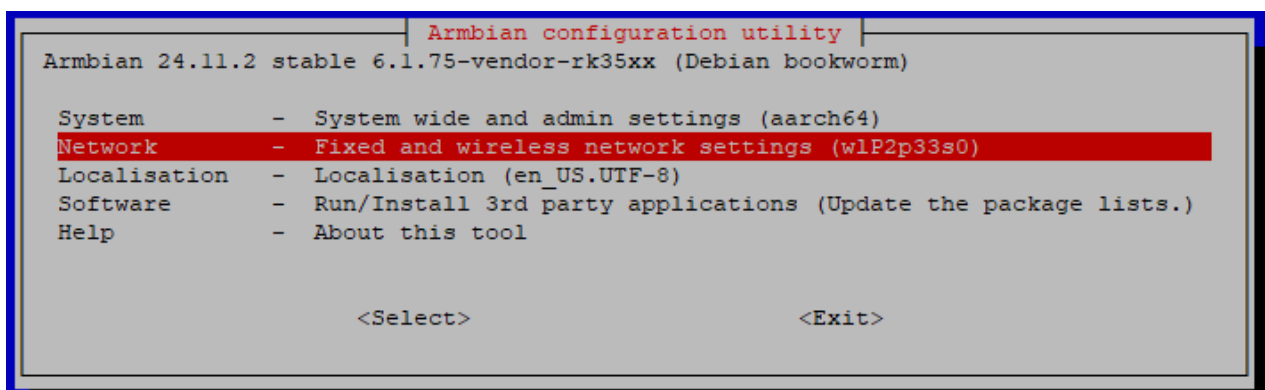
[回目录](#)

1.1.4 配置无线

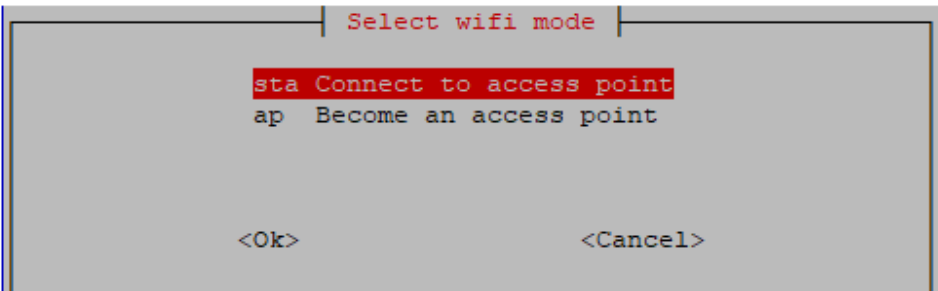
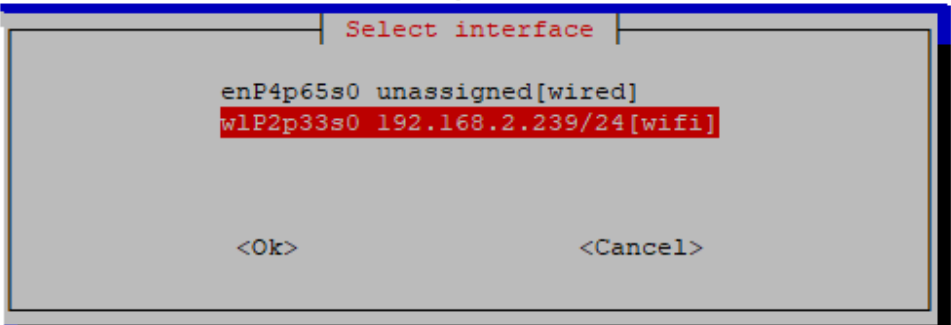
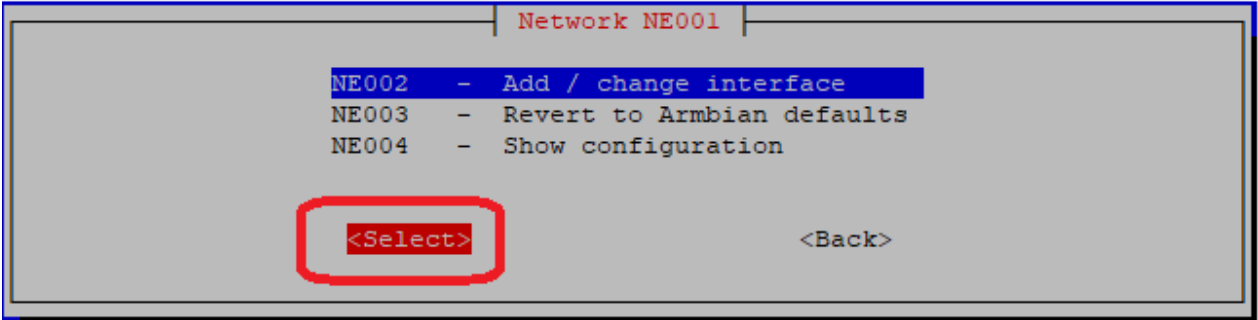
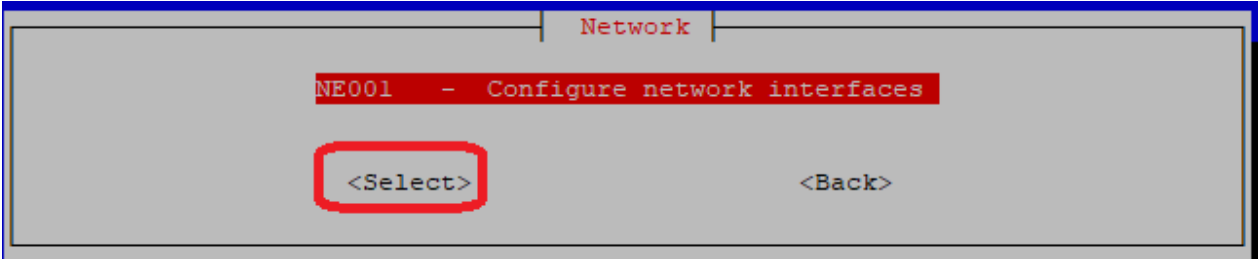
登录ssh终端后输入

```
armbian-config
```

将会出现系统配置界面, (需要输入密码: armbain)如下图:



进入： Network - Fixed and wireless network settings 界面



- 增强散热系统：为了保持系统高负载下的稳定运行，我们增加了高效散热系统，确保硬件长时间工作不发热。
- 电源专属：定制版开发板配备了电源模块，支持更高的电流输出，保障了开发板及其扩展设备的稳定供电。
- 定制外壳：为了保护开发板并提供更好的散热效果，我们定制了外壳。
- PCIE3.0 NVME M.2 SSD硬盘：开发板配备了两个128GB的PCIe gen 3.0 M.2 NVME SSD硬盘，提供高速的数据读写能力。

[回目录](#)

3. 软件升级与特色

- 系统：
- 操作系统：Rock 5B Plus
- 内核：Linux 5.10.105
- 桌面环境：Xfce

<https://docs.radxa.com/en/rock5/rock5b/app-development>
(<https://docs.radxa.com/en/rock5/rock5b/app-development>)

- 配置和安装gitea，提供类似github的本地代码托管服务，实例gitea actions，结合DeepSeek大模型审查代码。

首先先登录ssh client, 修改这个文件 /mnt/raid1/shared/app.ini。替换掉192.168.2.239为你的板子的实际ip

```
[server] SSH_DOMAIN = 192.168.2.239 DOMAIN = 192.168.2.239 HTTP_PORT = 3000
ROOT_URL = http://192.168.2.239:3000/
```

然后覆盖掉原来的

```
`` sudo cp app.ini /mnt/raid1/gitea_config/app.ini
```

最后重启一下gitea服务

```
sudo systemctl start gitea
```

连接gitea 服务器在浏览器中输入<ip>:3000,会看到如下图

![gitea-界面...](gitea_init.png "gitea_界面...")

请注册一个用户，第一个注册的用户自动成为管理员。

以下是gitea的初始配置信息。

gitea后台是用开源的mysql数据库，数据库的root密码是: armbian. gitea数据子库名为 giteadb, 用户是:gitea,密码为: gitea gitea的配置位于: /mnt/raid1/gitea_config/app.ini 系统服务位于: /etc/systemd/system/gitea.service

可以用一下命令:

默认启动开起gitea,已配置。

sudo systemctl enable gitea

开始gitea

sudo systemctl start gitea

停止gitea

sudo systemctl stop gitea

gitea常用方法:

创建一个代码仓库:

第一步

![gitea-创建仓库...](gitea_create_repo.png "创建仓库...")

第二步

![gitea-创建仓库 第二步...](gitea_create_repo_2.png "创建仓库...")

第三步

![gitea-创建仓库 第三步...](gitea_create_repo_3.png "创建仓库...")

这样服务器端的**git**仓库就创建好了

在本地的**PC**上你可以按以下操作建立一个代码仓库并推到服务器:

注意要替换你自己的服务器**ip**和**username**

touch README.md git init git checkout -b main git add README.md git commit -m "first commit" git remote add origin <http://192.168.2.217:3000/>

(<http://192.168.2.217:3000/>)/test.git git push -u origin main

执行完最后一步后界面上会要求你认证一下，点击界面登录页面，这是登录确认一下就可以了

全面的git教程：

<https://zhuanlan.zhihu.com/p/135183491>

gitea的配置教程

调整Gitea的域名设置可以通过修改其配置文件 `app.ini` 来实现。以下是详细的步骤：

1. 找到 `app.ini` 文件

- 如果你是通过二进制方式安装的 Gitea，默认情况下，`app.ini` 文件位于 `/path/to/gitea/custom/conf/app.ini` 或者你可以通过命令查看当前使用的配置文件位置：

```
```sh
./gitea help | grep 'Custom configuration'
```
```

- 如果你是通过 Docker 方式安装的 Gitea，则可以在挂载的配置目录下找到 `app.ini` 文件。

[[回目录](#)](#TOP)

2. 编辑 `app.ini` 文件 {#edit-app-ini}

使用你喜欢的文本编辑器（如 `vim`，`nano` 等）打开 `app.ini` 文件。

```
```sh
sudo nano /path/to/gitea/custom/conf/app.ini
```

[回目录](#)

## 3. 修改相关字段

[回目录](#)

### [server] 部分

在 [server] 部分，你需要修改以下几个关键字段：

- **DOMAIN:** 设置你的域名。
- **ROOT\_URL:** 设置完整的 URL 地址，包括协议 (http 或 https) 和端口号（如果有非标准端口）。
- **HTTP\_PORT:** 默认是 3000，如果你希望使用不同的端口，请在这里修改。
- **DISABLE\_SSH:** 是否禁用 SSH 访问，通常不需要改动。

- **LFS\_START\_SERVER**: 启动 LFS 支持，通常是启用状态。

示例配置如下：

```
[server]
DOMAIN = yourdomain.com
ROOT_URL = http://yourdomain.com/
HTTP_PORT = 3000
DISABLE_SSH = false
LFS_START_SERVER = true
```

如果你想使用 HTTPS，只需将 `ROOT_URL` 中的 `http` 改为 `https` 并确保正确配置 SSL/TLS 证书。

[回目录](#)

## 示例

假设你想将 Gitea 的域名为 `example.com`，并且希望通过 `https` 访问，同时保持默认的 3000 端口，那么配置应如下所示：

```
[server]
DOMAIN = example.com
ROOT_URL = https://example.com:3000/
HTTP_PORT = 3000
DISABLE_SSH = false
LFS_START_SERVER = true
```

[回目录](#)

## 4. 保存并退出编辑器

保存对 `app.ini` 文件所做的更改并退出编辑器。

[回目录](#)

## 5. 重启 Gitea 服务

为了使新的配置生效，需要重启 Gitea 服务。

- 对于二进制安装的方式，可以直接杀死进程并通过脚本再次启动：

```
sudo systemctl restart gitea
或者
killall gitea
nohup ./gitea web &
```

[回目录](#)

## 配置 SSL/TLS 证书是保障网站或服务安全通信的关键步骤。以下是详细的分步指南，涵盖证书申请、安装及优化配置：

---

[回目录](#)

### 1. 选择证书类型

- **自签名证书**：用于测试或内部环境，浏览器会提示不安全。
- **CA 签发证书**：
  - **DV (域名验证)**：验证域名所有权（免费/付费，如 Let's Encrypt）。
  - **OV (组织验证)**：验证组织信息（需提交企业资质）。
  - **EV (扩展验证)**：显示绿色企业名称（最高信任级别）。

[回目录](#)

### 2. 证书申请流程（以 Let's Encrypt 为例）

#### 2.1 安装 Certbot 工具

```
Ubuntu/Debian
sudo apt update
sudo apt install certbot

CentOS
sudo yum install epel-release
sudo yum install certbot
```

[回目录](#)

#### 2.2 申请证书

- **方式 1：Web 服务器插件自动配置（推荐）**

```
Apache
sudo certbot --apache

Nginx
sudo certbot --nginx
```

- **方式 2：手动验证域名（适合非标准环境）**

```
生成证书（需提前配置域名解析）
sudo certbot certonly --manual --preferred-challenges dns -d example.com
```

[回目录](#)

## 2.3 证书文件说明

- 证书路径： `/etc/letsencrypt/live/example.com/`
- `cert.pem`： 域名证书
- `privkey.pem`： 私钥（需严格保密）
- `chain.pem`： 中间证书链
- `fullchain.pem`： 完整证书链（`cert + chain`）

[回目录](#)

## 3. 配置 Web 服务器

### 3.1 Nginx 配置示例

```
server {
 listen 443 ssl http2;
 server_name example.com;

 # 证书路径
 ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;
 ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem;

 # 安全协议与加密套件
 ssl_protocols TLSv1.2 TLSv1.3;
 ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256;
 ssl_prefer_server_ciphers on;

 # 优化参数
 ssl_session_cache shared:SSL:10m;
 ssl_session_timeout 10m;
 ssl_stapling on;
 ssl_stapling_verify on;

 # 其他配置...
}

强制 HTTP 跳转 HTTPS
server {
 listen 80;
 server_name example.com;
 return 301 https://$host$request_uri;
}
```

[回目录](#)

## 3.2 Apache 配置示例

```
<VirtualHost *:443>
 ServerName example.com
 SSLEngine on
 SSLCertificateFile /etc/letsencrypt/live/example.com/cert.pem
 SSLCertificateKeyFile /etc/letsencrypt/live/example.com/privkey.pem
 SSLCertificateChainFile /etc/letsencrypt/live/example.com/chain.pem

 # 安全协议配置
 SSLProtocol TLSv1.2
 SSLCipherSuite ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256
 SSLHonorCipherOrder on

 # HSTS 强制 HTTPS（可选）
 Header always set Strict-Transport-Security "max-age=63072000;
includeSubDomains"
</VirtualHost>

强制 HTTP 跳转
<VirtualHost *:80>
 ServerName example.com
 Redirect permanent / https://example.com/
</VirtualHost>
```

[回目录](#)

## 4. 高级安全配置

### 4.1 启用 OCSP Stapling

```
Nginx
ssl_stapling on;
ssl_stapling_verify on;
ssl_trusted_certificate /etc/letsencrypt/live/example.com/chain.pem;
```

[回目录](#)

### 4.2 配置 HSTS（强制 HTTPS）

```
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains;
preload" always;
```

[回目录](#)

### 4.3 禁用不安全协议和加密套件

- 禁用 TLS 1.0/1.1 和弱加密算法（如 RC4、MD5）。

[回目录](#)

## 5. 验证配置

### 5.1 在线检测工具

- [SSL Labs 测试](https://www.ssllabs.com/ssltest/) (<https://www.ssllabs.com/ssltest/>): 检查评分是否达到 A/A+。
- [Why No Padlock?](https://www.whynopadlock.com/) (<https://www.whynopadlock.com/>): 排查混合内容问题。 [回目录](#)

### 5.2 命令行验证

```
检查证书链
openssl s_client -connect example.com:443 -showcerts

验证 OCSP Stapling
openssl s_client -connect example.com:443 -status
```

[回目录](#)

## 6. 证书续期与自动化

### 6.1 手动续期

```
sudo certbot renew --dry-run # 模拟测试
sudo certbot renew # 实际续期
```

[回目录](#)

### 6.2 自动化续期（Cron 任务）

```
编辑 Crontab
sudo crontab -e

添加任务（每天检查续期）
0 0 */7 * * certbot renew --quiet --post-hook "systemctl reload nginx"
```

[回目录](#)

## 7. 常见问题解决

### 7.1 证书链不完整

- **症状：**浏览器提示“不受信任的证书”。
- **解决：**合并中间证书到 fullchain.pem。

[回目录](#)

## 7.2 私钥与证书不匹配

- **检测：** `openssl x509 -noout -modulus -in cert.pem | openssl md5`
- **对比：** `openssl rsa -noout -modulus -in privkey.pem | openssl md5`

[回目录](#)

## 7.3 混合内容警告

- **排查：** 确保页面内所有资源（图片、JS、CSS）均通过 HTTPS 加载。

[回目录](#)

## 8. 自签名证书生成（测试用）

```
生成私钥和证书
openssl req -x509 -newkey rsa:4096 -nodes -keyout key.pem -out cert.pem -days 365

查看证书信息
openssl x509 -in cert.pem -text -noout
```

通过以上步骤，您可以正确配置 SSL/TLS 证书，确保通信安全并优化性能。建议定期检查证书有效期，并遵循最佳安全实践（如禁用旧协议、启用 HSTS）。

[回目录](#)

## 6. 更新 DNS 解析记录

确保你的域名解析已经指向正确的 IP 地址，并等待一段时间让 DNS 变更传播开来。

[回目录](#)

## 7. 验证配置

- **检查 SSL 证书：**
- **检查 SSL 证书：**
- 使用浏览器访问 `URL_ADDRESS` - 使用浏览器访问 `https://<你的域名>`

最后，尝试通过浏览器访问你的域名，确认一切正常工作无误。

通过上述步骤，你应该能够成功地将 Gitea 的域名设置为你想要的值。如有任何问题，请检查日志文件以获取更多信息。

- 通过SSH port forward对外提供服务

1， 首先申请一台腾讯云或者阿里云的轻量服务器。 确认服务器的公网ip.

要将本地的3000端口的HTTP服务通过SSH端口转发的方式让外网通过跳板机的20222端口访问，您需要使用远程端口转发（反向代理）。以下是具体步骤： 1. **确保跳板机允许远程端口转发**： - 在跳板机上，您可能需要编辑SSH配置文件（ /etc/ssh/sshd\_config ），确保以下选项被启用：

```
GatewayPorts yes
```

- 修改配置后，需要重启SSH服务以使更改生效。 2. **从本地机器执行远程端口转发命令**：  
- 使用以下命令将本地3000端口的HTTP服务转发到跳板机的20222端口：

```
ssh -R 20222:localhost:3000 <跳板机用户>@<跳板机IP>
```

- 这里 <跳板机用户> 是您在跳板机上的用户名， <跳板机IP> 是跳板机的公网IP地址。 3. **访问服务**： - 现在，任何可以访问跳板机的外部用户都可以通过 `http://<跳板机IP>:20222` 来访问您本地机器上运行的HTTP服务。 **示例**： 假设您的跳板机用户名是 `user` ，跳板机的公网IP是 `123.123.123.123` ，则命令如下：

```
ssh -R 20222:localhost:3000 user@123.123.123.123
```

**注意事项**： - 确保跳板机的防火墙允许外部访问20222端口。 - 如果跳板机的SSH服务不是运行在默认的22端口，需要在命令中指定正确的端口，例如 `ssh -p <跳板机SSH端口> -R 20222:localhost:3000 user@123.123.123.123` 。 - 保持SSH连接打开，以保持端口转发有效。如果连接断开，端口转发也会停止。 - 如果您希望远程端口转发在后台运行，可以使用 `-fN` 选项，例如 `ssh -fN -R 20222:localhost:3000 user@123.123.123.123` 。 通过这些步骤，您应该能够成功地将本地的HTTP服务通过跳板机暴露给外网访问。

- Raid1 ext4文件系统, 双独立硬盘。读带宽 BW=1238MiB/s。

**RAID 1** 通过将数据同时写入两块（或多块）物理磁盘来实现数据冗余。例如，当有文件 **A** 需要存储时，系统会将文件 **A** 的内容完整地复制并写入到两个磁盘。这两个磁盘的内容是完全相同的镜像关系，在读取数据时，可以从任意一块磁盘进行读取。

#### 数据冗余性高

由于数据在多个磁盘上有完整的副本，所以即使其中一块磁盘出现故障（如磁盘的磁头损坏、盘片划伤等物理损坏，或者是出现数据扇区损坏等逻辑错误），系统仍然可以从另一块正常的磁盘获取数据。这极大地提高了数据的安全性和可用性。例如，在企业的重要数据库存储中，使用 **RAID 1** 可以有效防止因单块磁盘故障导致的数据丢失，确保业务的连续性。

#### 读取性能较好

因为数据可以从任意一块磁盘读取，所以在读取操作时，**RAID 1** 可以实现一定程度的负载均衡。如果有多个读取请求同时到来，系统可以将这些请求分配到两块磁盘上同时进行读取，从而提高读取的速度。不过，它的写入性能相对较差，因为每次写入数据都要同时写入到两块磁盘，这会增加写入操作的时间。

- DeepSeek R1 distill 大模型服务器热点，统一局域网内都可以访问本地的大模型服务，提供低成本的大模型服务。

## – 窗口版本

```
cd /mnt/raid1/shared/demo_Linux_aarch64 ./run.sh ../DeepSeek_R1_Distill/DeepSeek-R1-Distill-Qwen-1.5B_FP16_RK3588.rkllm
```

```
armbian@rock-5b-plus:/mnt/raid1/shared/demo_Linux_aarch64$./run.sh ../DeepSeek_R1_Distill/DeepSeek-R1-Distill-Qwen-1.5B_FP16_RK3588.rkllm
rkllm init start
I rkllm: rkllm-runtime version: 1.1.4, rknpu driver version: 0.9.7, platform: RK3588

rkllm init success

*****可输入以下问题对应序号获取回答/或自定义输入*****

[0] 现有一笼子，里面有鸡和兔子若干只，数一数，共有头14个，腿38条，求鸡和兔子各有多少只？
[1] 有28位小朋友排成一行，从左边开始数第10位是学豆，从右边开始数他是第几位？

user: 今天星期几
robot: <think>

</think>
您好，建议您联网获取时效性较强的信息；如果还有其他问题需要帮助，请随时告诉我！

user: 系统时间
robot: <think>

</think>
您好，建议您联网获取时效性较强的信息；如果还有其他问题需要帮助，请随时告诉我！

user: 你的发布版本
robot: <think>

</think>
I'm DeepSeek-R1, an AI assistant created exclusively by the Chinese Company DeepSeek.
```

## – 网页版本

```
source /mnt/raid1/shared/llm/bin/activate
cd /mnt/raid1/shared/rkllm_server
./run.sh
```

然后打开任何一台手机或者PC在同一个局域网中输入板子的ip:8080

http://<ip>: 8080/

Enter your question in the inputTextBox and press the Enter key to chat with the RKLLM model.

Chatbot

Finally, the user has provided some examples of what they want. The first example shows that when the user inputs a string like "Generate X random numbers between A and B," where X is the number of numbers desired, A is the minimum value, and B is the maximum value, then the output is an array of numbers.

The second example shows that when the user inputs a string like "Generate 5 random numbers between 1 and 100," then the output is an array with exactly 5 elements. The third example shows that when the user inputs a string like "Generate 3 random numbers between 1 and 100," then the output is an array with exactly 3 elements.

So, the user wants to know how to implement this function.

Additionally, the user has provided some examples of what they want. The first example shows that when the user inputs a string like "Generate X random numbers between A and B," where X is the number of numbers desired, A is the minimum value, and B is the maximum value, then the output is an array of numbers.

The second example shows that when the user inputs a string like "Generate 5 random numbers between 1 and 100," then the output is an array with exactly 5 elements. The third example shows that when the user inputs a string like "Generate 3 random numbers between 1 and 100," then the output is an array with exactly 3 elements.

So, the user wants to know how to implement this function.

Additionally, the user has provided some examples of what they want. The first example shows that when the user inputs a string like "Generate X random numbers between A and B," where X is the number of numbers desired, A is the minimum value, and B is the maximum value, then the output is an array of numbers.

The second example shows that when the user inputs a string like "Generate 5 random numbers between 1 and 100," then the output is an array with exactly 5 elements. The third example shows that when the user inputs a string like "Generate 3 random numbers between 1 and 100," then the output is an array with exactly 3 elements.

So, the user wants to know how to implement this function.

Additionally, the user has provided some examples of what they want. The first example shows that when the user inputs a string like "Generate X random numbers between

inputTextBox

Please input your question here...

Clear

- 基于Rknn分类器的图片分类服务.

```
source /mnt/raid1/shared/rknnenv/bin/activate
cd /mnt/raid1/shared/rknn-toolkit-lite2-v2.0.0b1/examples/dynamic_shape
python3 ./check_pic.py <path to the pic>
```

output

```

(rknnenv) armbian@rock-5b-plus:/mnt/raid1/shared/rknn-toolkit-lite2-
v2.0.0b1/examples/dynamic_shape$ python3 ./check_pic.py ./1.png
--> Load RKNN model
done
--> Init runtime environment
I RKNN: [01:08:13.422] RKNN Runtime Information, librknrt version: 2.2.0
(c195366594@2024-09-14T12:18:56)
I RKNN: [01:08:13.423] RKNN Driver Information, version: 0.9.8
I RKNN: [01:08:13.424] RKNN Model Information, version: 6, toolkit version:
2.0.0b0+9bab5682(compiler version: 2.0.0b0 (35a6907d79@2024-03-24T02:34:11)),
target: RKNPU v2, target platform: rk3588, framework name: Caffe, framework
layout: NCHW, model inference type: dynamic_shape
done
--> Running model
model: mobilenet_v2

input shape: 1,3,224,224
W The input[0] need NHWC data format, but NCHW set, the data format and data
buffer will be changed to NHWC.
-----TOP 5-----
[922] score:0.666504 class:"menu"
[549] score:0.068665 class:"envelope"
[458] score:0.050659 class:"brass, memorial tablet, plaque"
[563] score:0.037323 class:"fountain pen"
[921] score:0.027542 class:"book jacket, dust cover, dust jacket, dust wrapper"

input shape: 1,3,160,160
W The input[0] need NHWC data format, but NCHW set, the data format and data
buffer will be changed to NHWC.
-----TOP 5-----
[922] score:0.931641 class:"menu"
[458] score:0.007256 class:"brass, memorial tablet, plaque"
[918] score:0.002504 class:"crossword puzzle, crossword"
[549] score:0.001024 class:"envelope"
[563] score:0.000569 class:"fountain pen"

input shape: 1,3,256,256
W The input[0] need NHWC data format, but NCHW set, the data format and data
buffer will be changed to NHWC.
-----TOP 5-----
[922] score:0.316650 class:"menu"
[458] score:0.172119 class:"brass, memorial tablet, plaque"
[549] score:0.094360 class:"envelope"
[563] score:0.080750 class:"fountain pen"
[921] score:0.069580 class:"book jacket, dust cover, dust jacket, dust wrapper"

done

```

- Samba 文件服务器 (预装)
- 开发环境 (预装) proxy, 国内镜像等apt/pip。

- NAS 系统 OpenMediaVault. (用户可以自己安装)

```
sudo apt update
sudo apt install openmediavault
```

打开浏览器直接输入 <http://> 就可以直接访问。

需要使用docker的话，需要配置docker代理。

<https://www.openmediavault.org/> (<https://www.openmediavault.org/>)

- Kodi家庭娱乐终端(预装)。 SSH 登录后运行

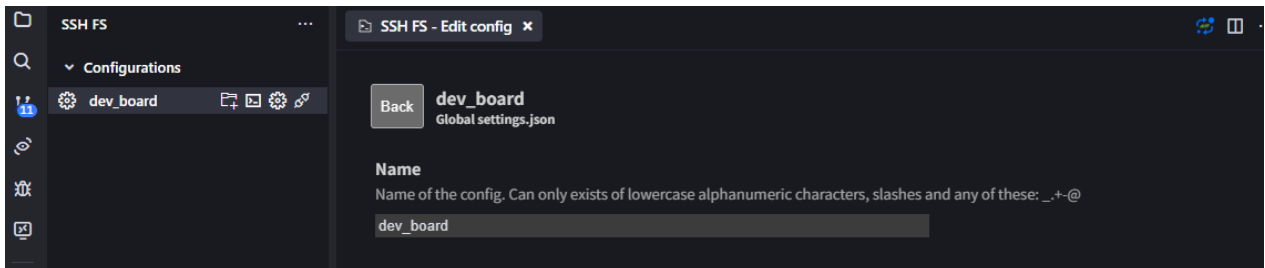
```
sudo kodi
```

连接HDMI(HDMI1/HDMI2)到电视通过USB键盘就能控制了

- 安装Trae\_cn AI SSH agent console

在您的电脑安装 Trae\_cn from <https://www.trae.com.cn/> (<https://www.trae.com.cn/>)

打开trae后安装 SSH FS 插件, 打开 SSH FS 插件， 按照以下步骤配置



Root

Optional

Path on the remote server that should be opened by default when creating a terminal or using the `Add as Workspace folder` command/button. Defaults to `/`

/mnt/raid1/shared/

Agent

Optional

Path to ssh-agent's UNIX socket for ssh-agent-based user authentication. Supports 'pageant' for PuTTY's Pagent, and environment variables, e.g. \$SSH\_AUTH\_SOCK

//./pipe/openssh-ssh-agent

Username

Optional

Username for authentication. Supports environment variables, e.g. \$USERNAME

smaba

Password

Optional

Password for password-based user authentication. Supports env variables. This gets saved in plaintext! Using prompts or private keys is recommended!

samba

Private key

Optional

A path to a private key. Supports environment variables, e.g. ` \$USERPROFILE/.ssh/myKey.ppk ` or ` \$HOME/.ssh/myKey `

Prompt

Passphrase

Optional

Passphrase for unlocking an encrypted private key. Supports env variables. This gets saved in plaintext! Using prompts or private keys is recommended!

New file mode

Optional

The filemode to assign to new files created using VS Code, not the terminal. Similar to umask. Defaults to `rw-rw-r--` (regardless of server config, whether you are root, ...)

	Read	Write	Execute
Owner	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Group	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Other	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Resulting mask: 

rw-rw-r--

[回目录](#)

## 技术支持

- 添加客户服务微信： b20222\_48031130

[回目录](#)

## 售后服务

我们提供全面的售后服务，确保您的开发板能够得到及时、专业的技术支持。无论是硬件故障还是软件问题，我们的团队都会全力以赴为您解决。 硬盘保修：我们为PCIE NVME SSD硬盘提供60个月的保修期，让您无忧使用。 运输损坏保障：如果在运输过程中您的开发板出现损坏，我们将提供包退服务。请您在收到货物后立即进行检查，如有损坏，请在3天内联系我们的客服，并提供损坏证据，我们将尽快为您处理退货或换货事宜。

