



---

# 喵喵派开发手册

---



2024-8-17

## 目录

WSL 开发环境搭建.....	2
Clion 交叉编译环境配置.....	3

# WSL 开发环境搭建

## 1. 开启 WSL

参考教程：[链接](#)

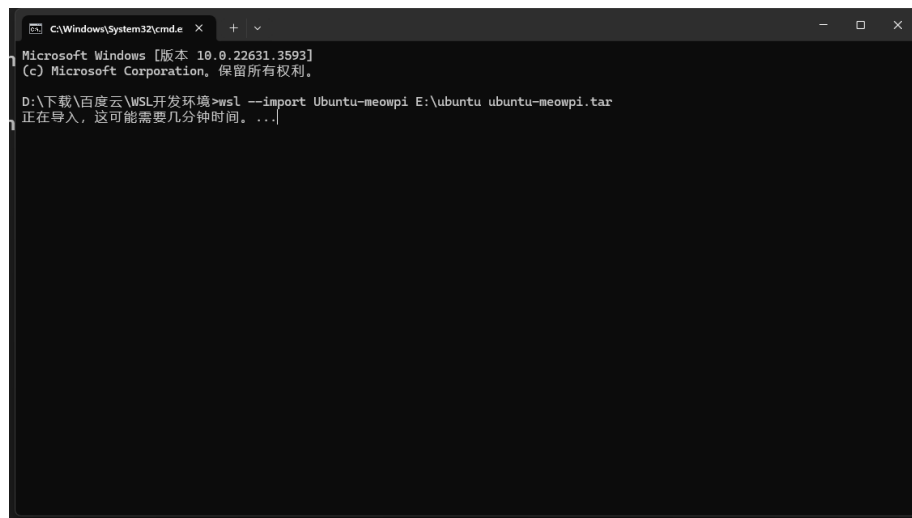
## 2. 使用下载的 Ubuntu 映像进行 WSL 还原

下载并解压 ubuntu-meowpi.tar。

打开 CMD，输入以下命令：

```
wsl --import Ubuntu-meowpi E:\ubuntu E:\ubuntu-meowpi.tar
```

其中 E:\ubuntu 为 WSL 的安装路径，E:\ubuntu-meowpi.tar 为映像文件的路径。

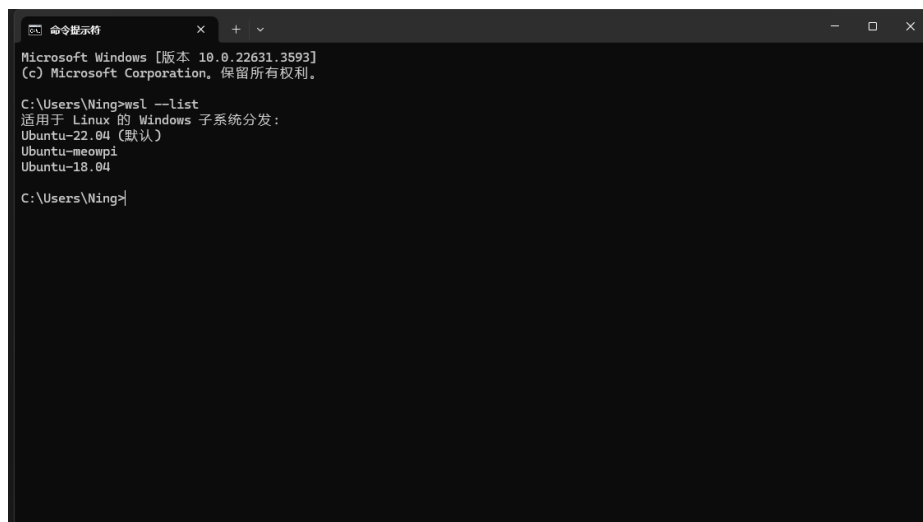


```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.22631.3593]
(c) Microsoft Corporation. 保留所有权利。

D:\下载\百度云\WSL开发环境>wsl --import Ubuntu-meowpi E:\ubuntu ubuntu-meowpi.tar
正在导入，这可能需要几分钟时间。...
```

## 3. 验证安装

输入 `wsl --list`，确认 Ubuntu-meowpi 是否正确安装。



```
命令提示符
Microsoft Windows [版本 10.0.22631.3593]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\Ning>wsl --list
适用于 Linux 的 Windows 子系统分发:
Ubuntu-22.04 (默认)
Ubuntu-meowpi
Ubuntu-18.04

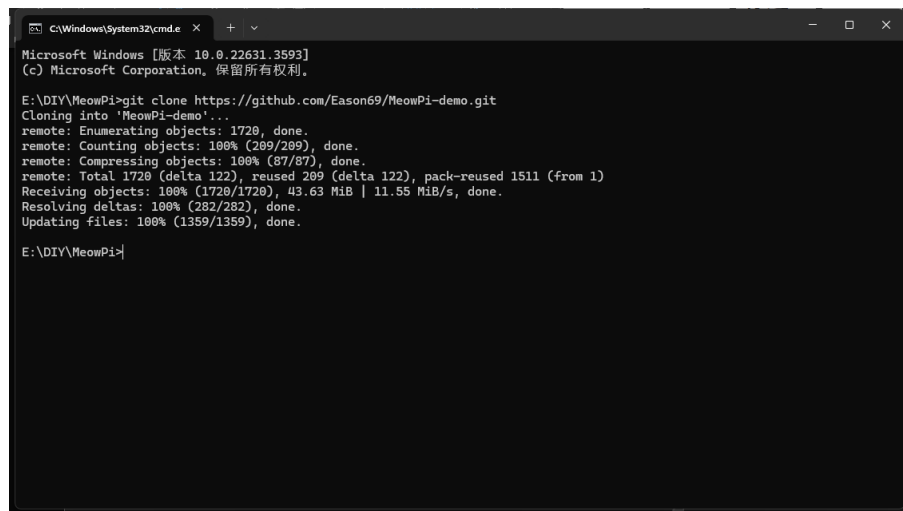
C:\Users\Ning>
```

# Clion 交叉编译环境配置

所有 IDE 的环境配置基本相同，这里以 clion 为例子进行配置

通过 Git 克隆我们的开源项目，或创建一个新的 C++ 项目：

git clone <https://github.com/Eason69/MeowPi-demo.git>

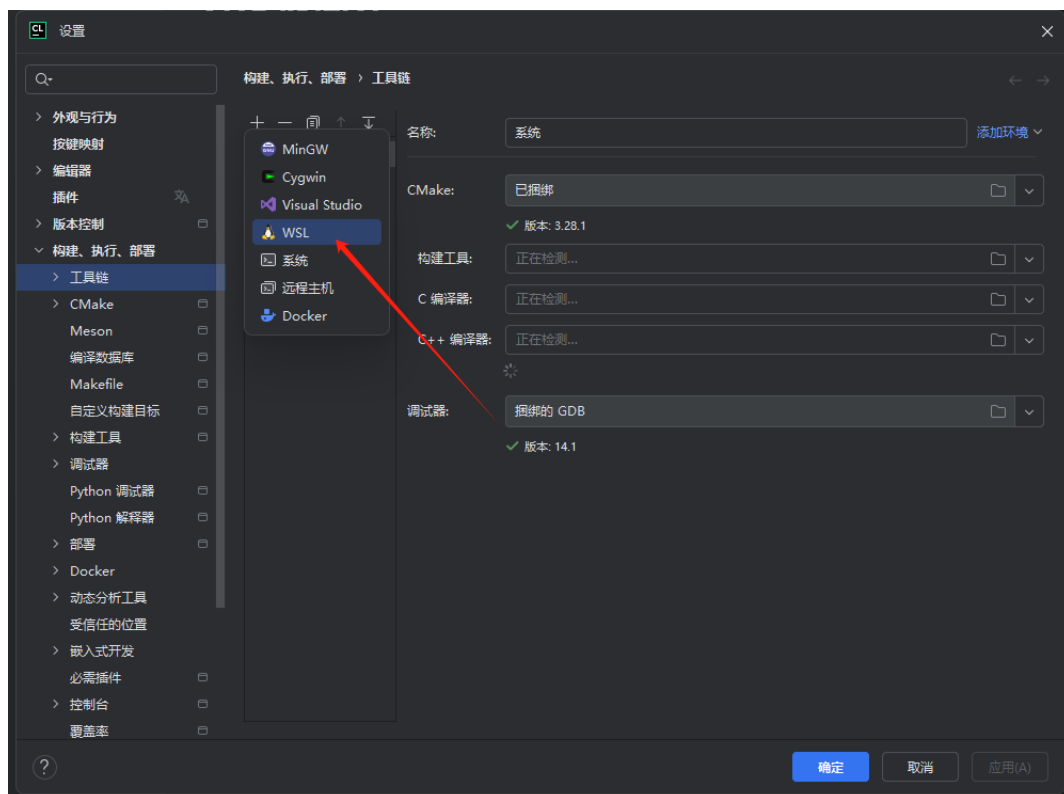


```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.22631.3593]
(c) Microsoft Corporation. 保留所有权利。

E:\DIY\MeowPi>git clone https://github.com/Eason69/MeowPi-demo.git
Cloning into 'MeowPi-demo'...
remote: Enumerating objects: 1720, done.
remote: Counting objects: 100% (209/209), done.
remote: Compressing objects: 100% (87/87), done.
remote: Total 1720 (delta 122), reused 209 (delta 122), pack-reused 1511 (from 1)
Receiving objects: 100% (1720/1720), 43.63 MiB | 11.55 MiB/s, done.
Resolving deltas: 100% (282/282), done.
Updating files: 100% (1359/1359), done.

E:\DIY\MeowPi>
```

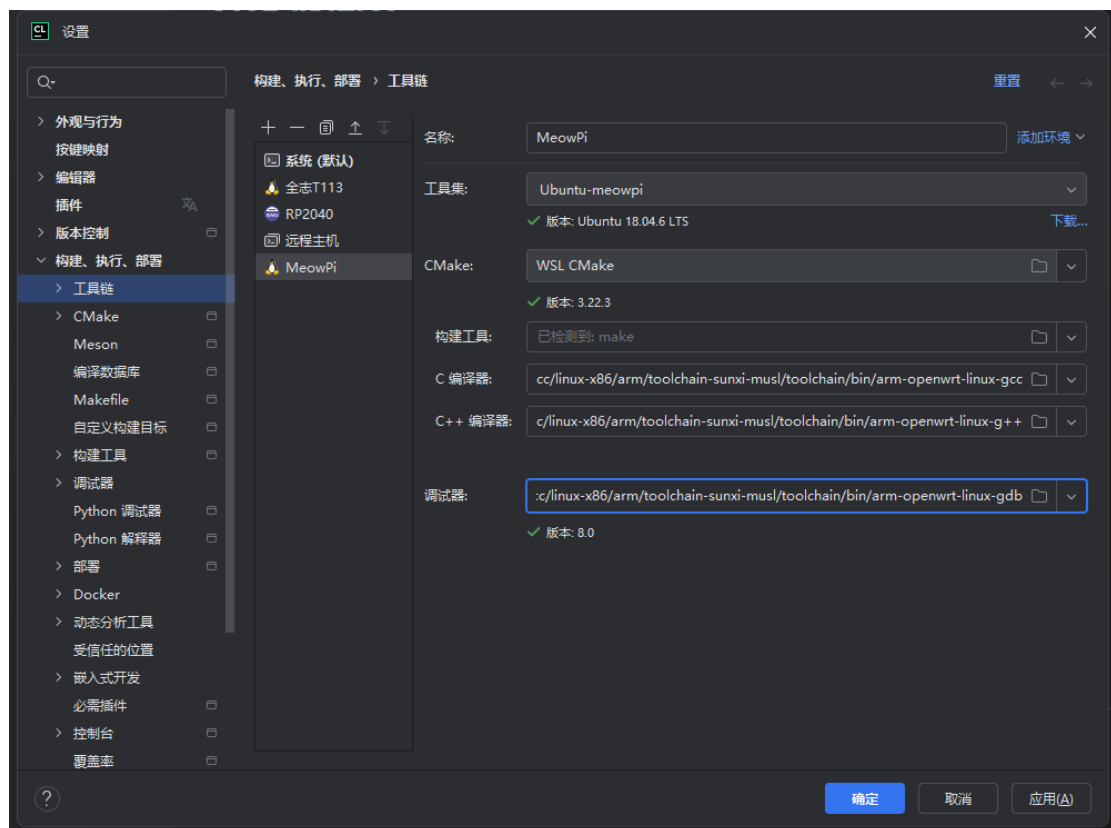
使用 clion 打开或者新建一个 cmake 程序，然后在设置 工具链 里添加一个 WSL



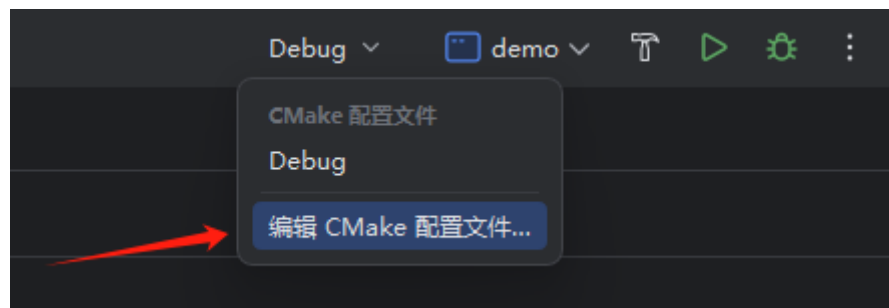
## 配置工具链

在 CLion 中打开项目或新建一个 CMake 项目，在设置中添加 WSL 工具链。

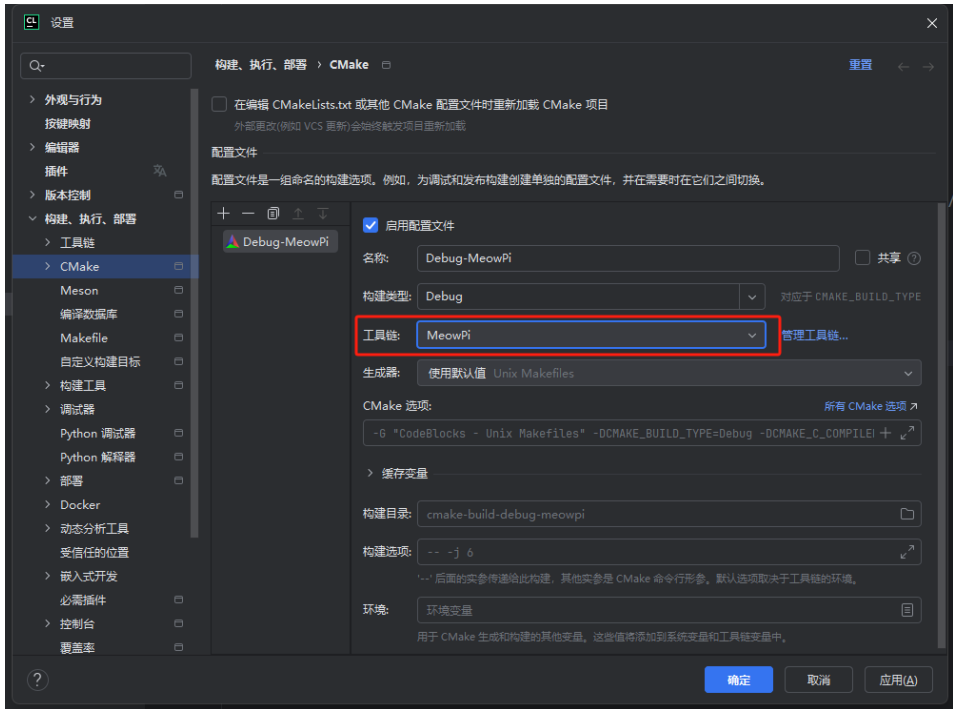
- 名称: MeowPi
- 工具集: Ubuntu-meowpi
- C 编译器: /home/book/tina-d1-h/prebuilt/gcc/linux-x86/arm/toolchain-sunxi-musl/toolchain/bin/arm-openwrt-linux-gcc
- C++ 编译器: /home/book/tina-d1-h/prebuilt/gcc/linux-x86/arm/toolchain-sunxi-musl/toolchain/bin/arm-openwrt-linux-g++
- 调试器: /home/book/tina-d1-h/prebuilt/gcc/linux-x86/arm/toolchain-sunxi-musl/toolchain/bin/arm-openwrt-linux-gdb



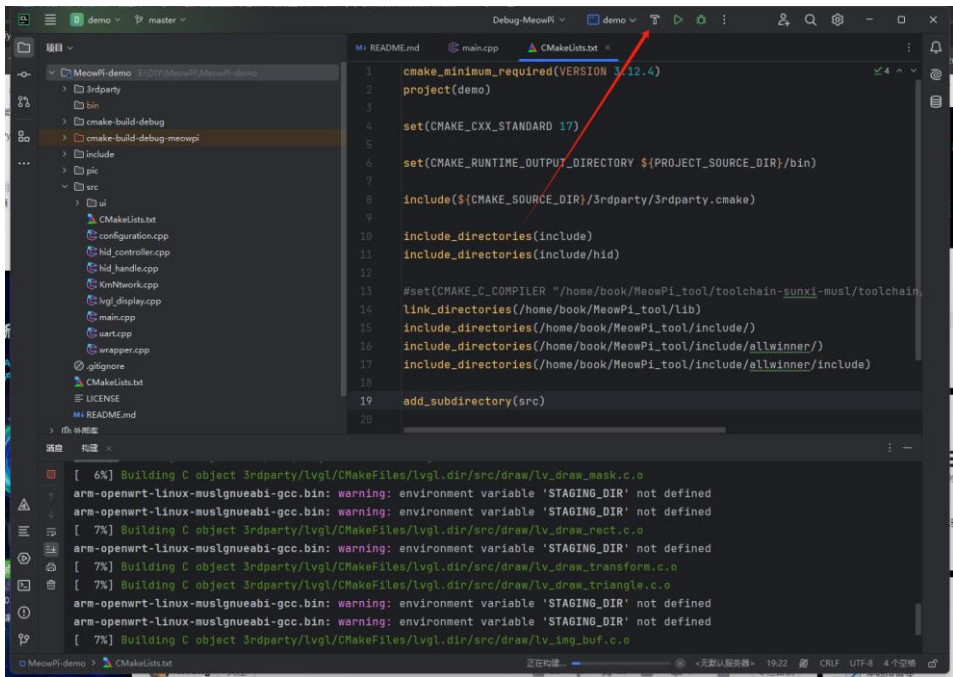
点击编辑 CMakepz



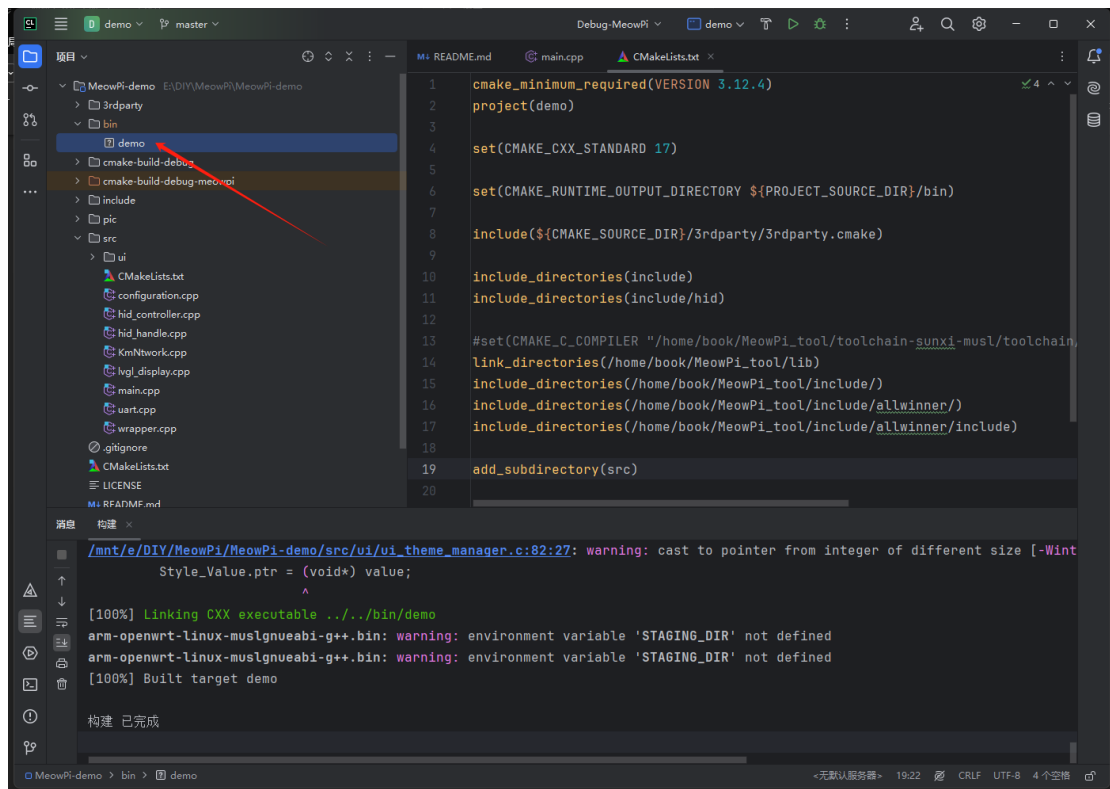
工具链选择刚刚配置的 meowpi，然后确定。



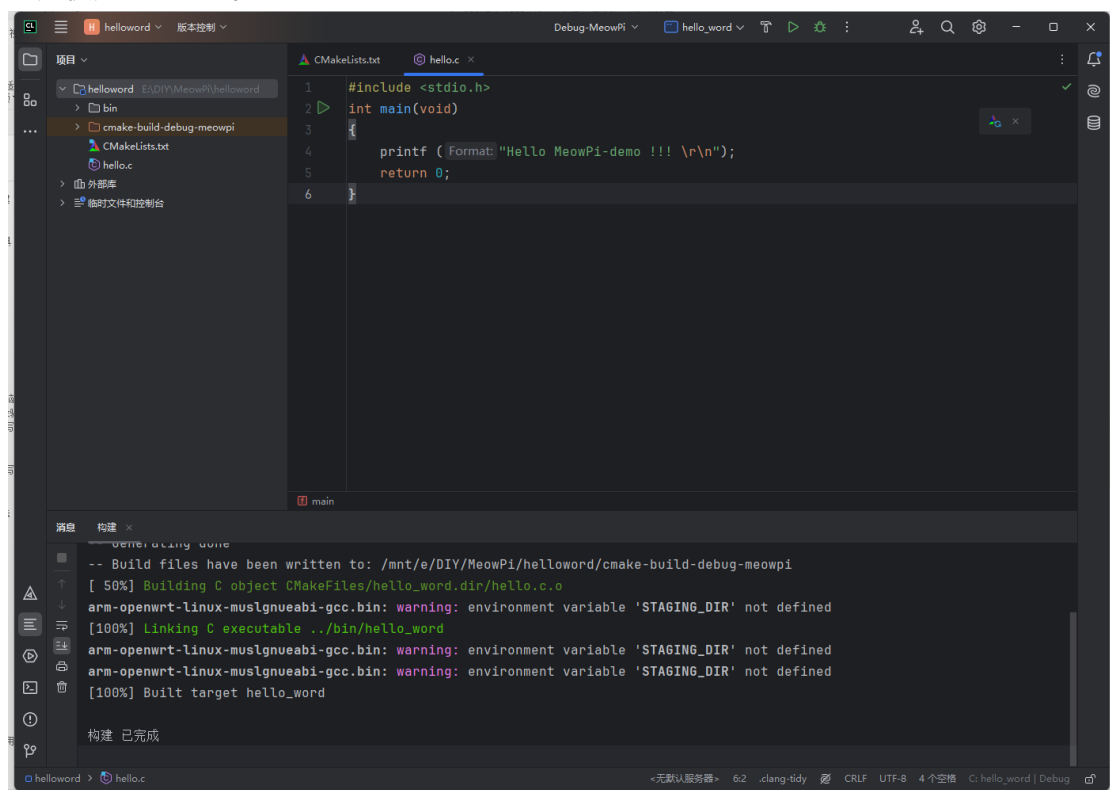
点击构建，ide 开始编译。



构建完成后，bin 目录下就是编译完成的文件。



这是使用 hello 程序。



hello.c:

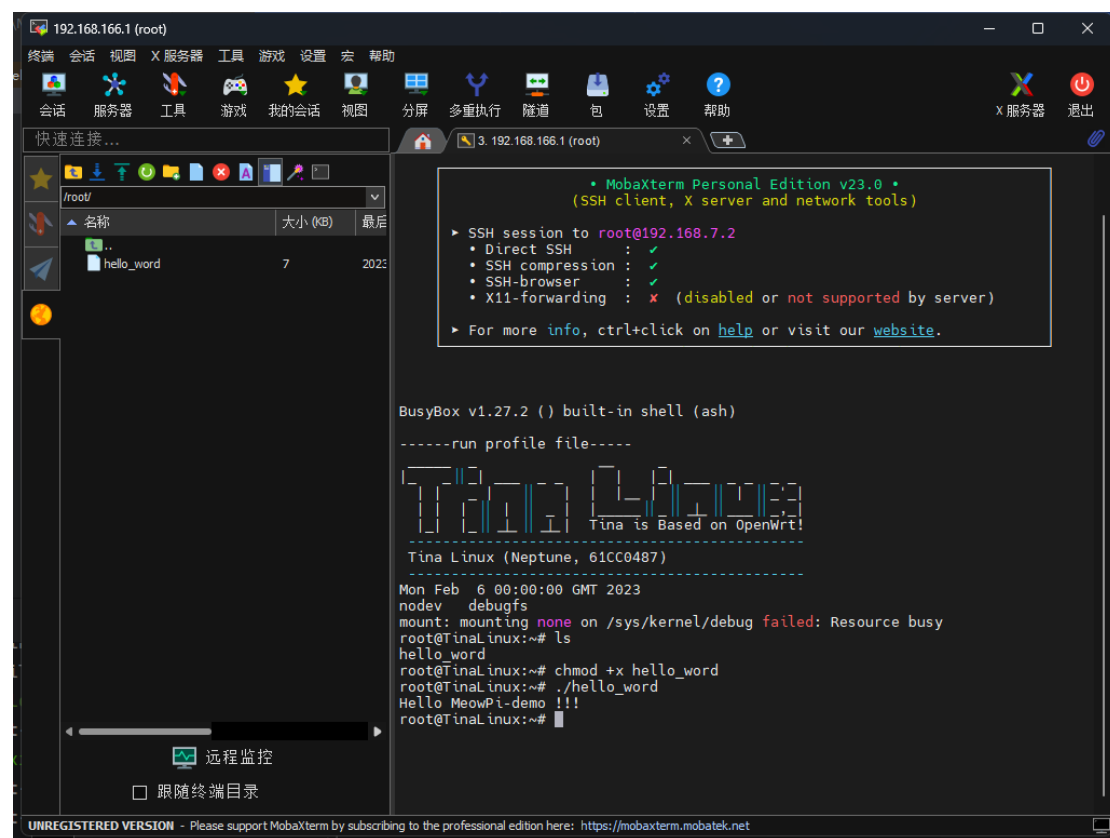
```
#include <stdio.h>
int main(void)
{
    printf("Hello MeowPi-demo !!! \r\n");
    return 0;
}
```

CMakeLists.txt:

```
cmake_minimum_required(VERSION 3.10)
project(helloworld)
SET(CROSS_COMPILE 1)
set(CMAKE_SYSTEM_NAME Linux)
set(CMAKE_RUNTIME_OUTPUT_DIRECTORY ${PROJECT_SOURCE_DIR}/bin)

add_executable(hello_word hello.c)
```

把 bin 文件夹下的 hello\_word 通过 MobaXterm 的 SFTP 上传。



使用 `chmod +x hello_word` 给 hello 程序添加可执行权限  
`./hello_word` 运行