

# 基于双椒派开发板的嵌入式开发环境搭建

(V1.1)

2023 年 4 月 28 日

修改历史

版本	位置	修改内容	修改人
V1.0		创建文档	
V1.1		格式重排	

## 目 录

<b>1 交叉编译环境介绍 .....</b>	<b>4</b>
1.1 什么是交叉编译 .....	4
1.2 为什么需要交叉编译? .....	4
1.3 如何交叉编译? .....	5
<b>2 UBUNTU20.04 LINUX 系统.....</b>	<b>6</b>
2.1 打开 WINDOWS 的子系统功能（支持 LINUX 系统） .....	6
2.2 下载安装 UBUNTU 镜像 .....	7
2.3 更新 UBUNTU 软件源 .....	8
<b>3 安装交叉编译器 .....</b>	<b>8</b>
3.1 下载交叉编译工具链 .....	8
3.2 安装工具链 .....	9
3.3 配置工具链（环境变量） .....	9
3.4 安装必要的软件包 .....	10
<b>4 串口访问开发板 .....</b>	<b>10</b>
4.1 串口线物理连接 .....	10
4.2 安装配置 MINICOM 串口工具 .....	11
4.3 启动开发板 .....	13
4.4 常见问题 .....	14
<b>5 SSH 访问开发板 .....</b>	<b>14</b>
5.1 开发板安装 SSH 服务 .....	14
5.2 SSH 登陆 .....	15
5.3 常见问题 .....	16

## 1 交叉编译环境介绍

### 1.1 什么是交叉编译

在现代，大家编写程序通常使用高级语言如：C/C++语言、go 语言。但是计算机只认识 1 和 0 组成的指令。编译就是把高级语言变成计算机可以识别的二进制可执行程序的过程。而不同体系架构的 CPU 使用的二进制指令集是不同的，比如 X86 的指令集与 ARM 不同。因而，不同 CPU 就需要有支持这种 CPU 架构指令集的编译程序。而在同一种 CPU 上运行的不同的操作系统，也不可能运行同一个可执行程序。比如将 Windows11 上的某个 .exe 程序拷贝到 Linux 系统上是无法加载运行的。

在一种计算机环境中运行的编译程序，能编译出在另外一种环境下运行的代码，我们就称这种编译器支持交叉编译，这个编译过程就叫交叉编译。简单地说，就是在一个平台上生成另一个平台上的可执行代码。这里需要注意的是所谓平台，参考上面的描述，实际上包含两个概念：CPU 体系结构（Architecture）与操作系统

（OperatingSystem）。同一个体系结构可以运行不同的操作系统；同样，同一个操作系统也可以在不同的体系结构上运行。举例来说，我们常说的 x86 Linux 平台实际上是 Intel x86 体系结构和 Linux for x86 操作系统的统称；而 x86 WinNT 平台实际上是 Intel x86 体系结构和 Windows NT for x86 操作系统的简称。

交叉编译这个概念的出现和流行是和嵌入式系统的广泛发展同步的。我们常用的计算机软件，都需要通过编译的方式，把使用高级计算机语言编写的代码（比如 C 代码）编译（compile）成计算机可以识别和执行的二进制代码。比如，我们在 Windows 平台上，可使用 Visual C++ 开发环境，编写程序并编译成可执行程序。这种方式下，我们使用 PC 平台上的 Windows 工具开发针对 Windows 本身的可执行程序，这种编译过程称为 native compilation，中文可理解为本机编译。然而，在进行嵌入式系统的开发时，运行程序的目标平台通常具有有限的存储空间和运算能力，比如早期的 ARM 平台，其一般的静态存储空间大概是 16 到 32MB，而 CPU 的主频大概在 100MHz 到 500MHz 之间。这种情况下，在 ARM 平台上进行本机编译就不太可能了，这是因为一般的编译工具链（compilation tool chain）需要很大的存储空间，并需要很强的 CPU 运算能力。为了解决这个问题，交叉编译工具就应运而生了。通过交叉编译工具，我们就可以在 CPU 能力很强、存储控件足够的主机平台上（比如 PC 上）编译出针对其他平台的可执行程序。

### 1.2 为什么需要交叉编译？

有时因为目标平台上不允许或不能够安装我们所需要的编译器，而我们又需要这个编译器的某些特征；

有时因为目标平台上的资源贫乏，无法运行我们所需要编译器；  
有时因为目标平台还没有建立，连操作系统都没有，根本谈不上运行什么编译器。

1.3 如何交叉编译？

要进行交叉编译，我们需要在主机平台上安装对应的交叉编译工具链（crosscompilation tool chain），然后用这个交叉编译工具链编译我们的源代码，最终生成可在目标平台上运行的代码。

例如：以如图 2-1 所示的开发环境，主机为 X86 PC 机，目标机为 E2000 双椒派开发板。主机负责编译目标机运行的软件，并且通过串口与目标机在安装启动等过程中进行通讯，加以控制。在 X86 Linux PC 上，运行 Linux 操作系统，使用 arm-linux-gcc 交叉编译器，可编译出针对 Linux ARM 平台的可执行代码。

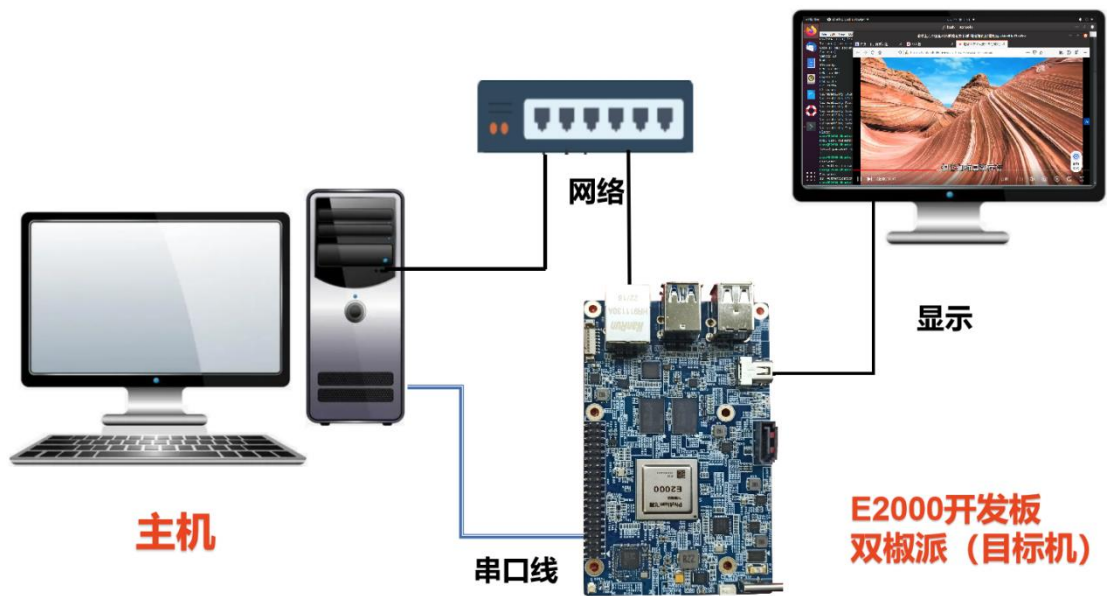


图 2-1

主机	<ul style="list-style-type: none"><li>• Ubuntu20.04 Linux 系统</li><li>• 交叉编译器</li><li>• 串口工具</li></ul>
----	---

目标机（开发板）	<ul style="list-style-type: none"><li>与主机可以通过串口交互</li><li>在 OS 支持情况下通过网络于主机交互</li><li>运行嵌入式 OS</li></ul>
----------	--

本章我们来逐步搭建主机环境

## 2 Ubuntu20.04 Linux 系统

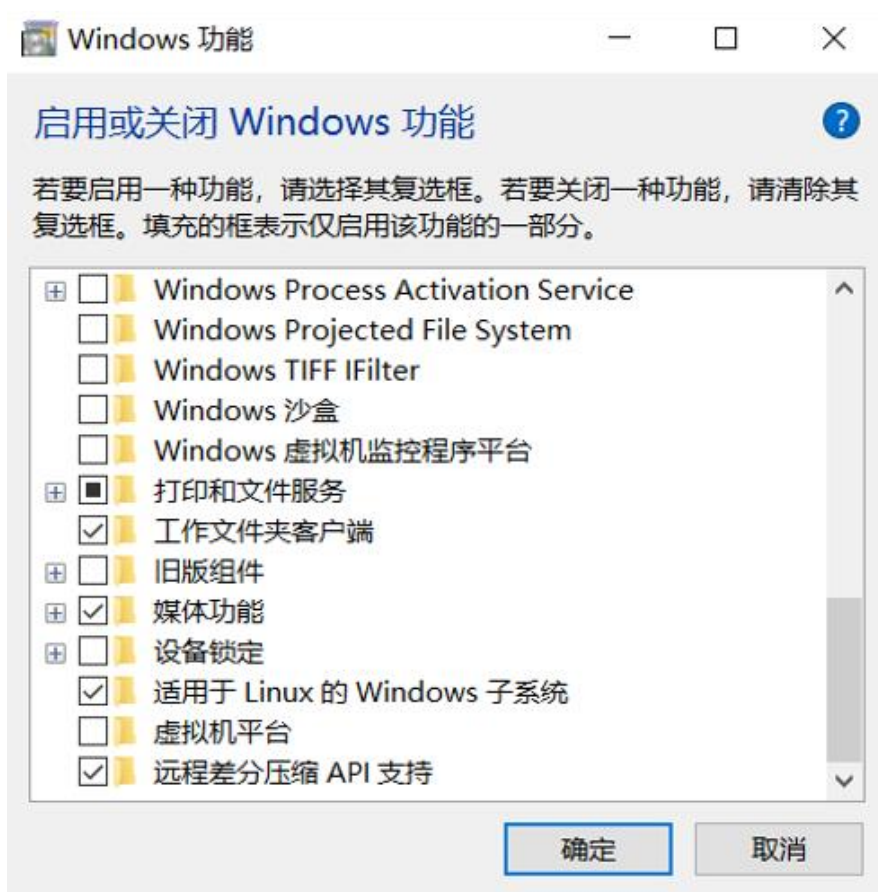
主机的 Ubuntu20.04 可以直接安装在物理主机上，也可以安装在虚拟机中。这里我们以在 windows 10 系统安装 Ubuntu 虚拟机为例：

### 2.1 打开 windows 的子系统功能（支持 Linux 系统）

#### 1、控制面板



#### 2、打开下面的选型



## 2.2 下载安装 ubuntu 镜像

打开 win10 的应用商店搜索 ubuntu 免费安装即可



## 2.3 更新 ubuntu 软件源

由于 ubuntu 默认的源是国外的，所以我们需要更换国内源。

### 1、备份原软件源

Plain Text

```
mv /etc/apt/source.list /etc/apt/source.list_bak
```

### 2、根据清华大学开源软件站的 ubuntu 软件源帮助手册替换软件源

<https://mirror.tuna.tsinghua.edu.cn/help/ubuntu/>

### 3、更新软件源

```
sudo apt-get update  
sudo apt-get upgrade
```

## 3 安装交叉编译器

### 3.1 下载交叉编译工具链

#### 1、下载地址：

<https://releases.linaro.org/components/toolchain/binaries/7.4-2019.02/aarch64-linux-gnu/>

#### 2、下载 x86 平台的交叉编译工具链

[gcc-linaro-7.4.1-2019.02-x86\\_64\\_aarch64-linux-gnu.tar.xz](#)



Name	Last modified	Size	License
Parent Directory			
gcc-linaro-7.4.1-2019.02-i686-mingw32_aarch64-linux-gnu.tar.xz	26-Jan-2019 00:03	351.8M	open
gcc-linaro-7.4.1-2019.02-i686-mingw32_aarch64-linux-gnu.tar.xz.asc	25-Jan-2019 06:38	97	open
gcc-linaro-7.4.1-2019.02-i686_aarch64-linux-gnu.tar.xz	26-Jan-2019 00:04	110.2M	open
gcc-linaro-7.4.1-2019.02-i686_aarch64-linux-gnu.tar.xz.asc	25-Jan-2019 06:39	89	open
gcc-linaro-7.4.1-2019.02-linux-gnu.tar.xz	25-Jan-2019 06:39	10.1K	open
gcc-linaro-7.4.1-2019.02-linux-gnu.tar.xz.asc	25-Jan-2019 06:39	10.1K	open
gcc-linaro-7.4.1-2019.02-win32-manifest.txt	25-Jan-2019 06:39	10.1K	open
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu.tar.xz	26-Jan-2019 00:04	111.5M	open
gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu.tar.xz.asc	25-Jan-2019 06:39	91	open
runtime-gcc-linaro-7.4.1-2019.02-aarch64-linux-gnu.tar.xz	26-Jan-2019 00:04	6.7M	open
runtime-gcc-linaro-7.4.1-2019.02-aarch64-linux-gnu.tar.xz.asc	25-Jan-2019 06:39	92	open
sysroot-glibc-linaro-2.25-2019.02-aarch64-linux-gnu.tar.xz	26-Jan-2019 00:04	45.6M	open
sysroot-glibc-linaro-2.25-2019.02-aarch64-linux-gnu.tar.xz.asc	25-Jan-2019 06:39	155	open

## 3.2 安装工具链

1、在/opt 下创建一个 toolchain 文件夹

```
mkdir /opt/toolchain
```

2、将下载的 gcc-linaro-7.4.1-2019.02-x86\_64\_aarch64-linux-gnu.tar.xz 复制到 /opt/toolchain 目录下并解压。

```
cp gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu.tar.xz
/opt/toolchain
tar -xf gcc-linaro-7.4.1-2019.02-x86_64_aarch64-linux-gnu.tar.xz
```

## 3.3 配置工具链（环境变量）

1、修改环境变量

1) 打开/etc/profile 文件

```
Plain Text
vi /etc/profile
```

2) 在文本底部增加如下两行：

```
Plain Text
export PATH=$PATH:/opt/toolchain/gcc-linaro-7.4.1-2019.02-x86_64_aarch64-
linux-gnu/bin
export CROSS_COMPILE=aarch64-linux-gnu- //路径为交叉编译工具链绝对
路径
```

3) 保存并退出文本编辑

## 2、生效 profile 配置文件

执行如下命令，更新 profile 配置文件

```
Plain Text  
source /etc/profile
```

## 3.4 安装必要的软件包

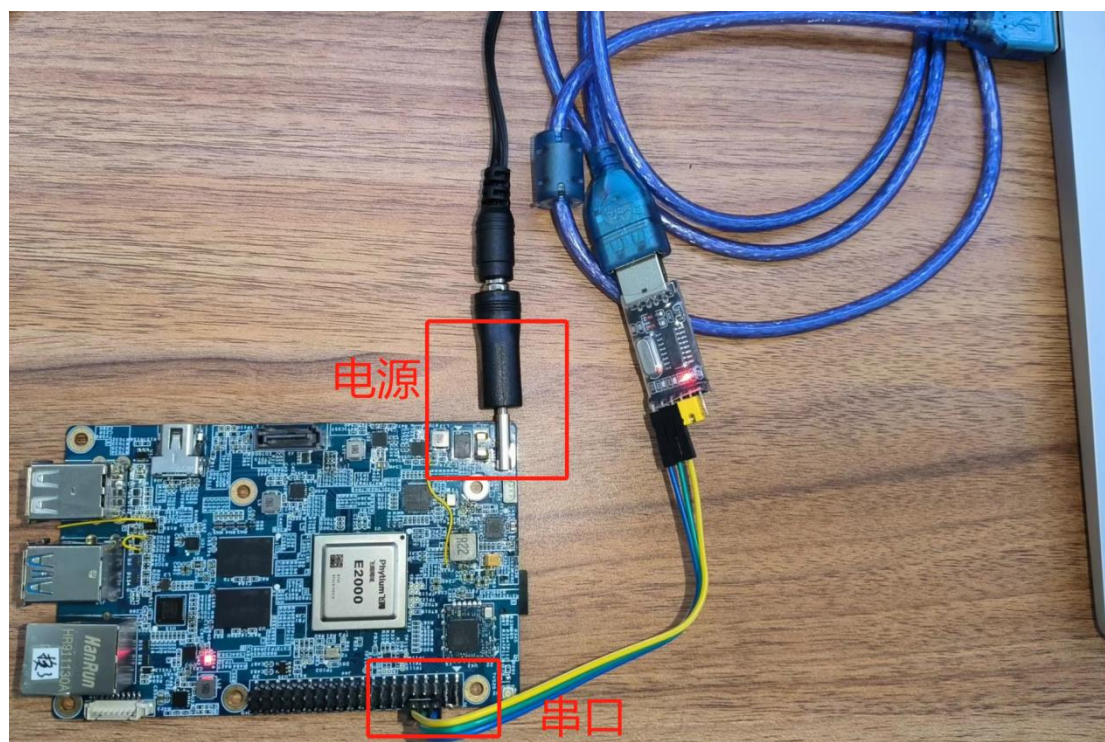
```
sudo apt-get install debootstrap qemu-system-common qemu-user-static  
binfmt-support
```

## 4 串口访问开发板

这里我们以 Ubuntu 下使用 minicom 串口软件为例，演示连接到开发板的串口，用以交互命令与文件。仅就连接开发板串口这一项功能而言，主机使用其他的系统或串口软件也是可以的，比如在 windows 上使用 MobaXterm、putty 等。

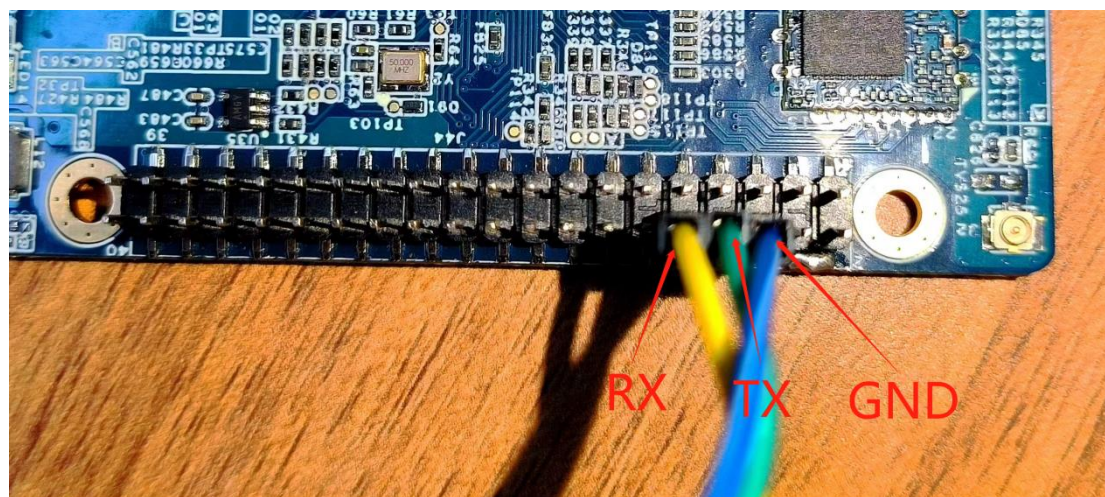
### 4.1 串口线物理连接

1、将 USB 转 TTL 调试串口线连接到 Ubuntu PC 机 USB 接口



## 2、将串口线与开发板相连

检查开发板上的丝印信息，确保调试串口线连接正确，开发板的调试串口位于 40pin 排插从右下角第 3(GND)，4(TX)，5(RX) 管脚。保证开发板的 TX 连接到串口转换器的 RX 管脚，开发板的 RX 管脚连接到转换器的 TX 管脚。具体如下图所示：



## 4.2 安装配置 minicom 串口工具

- 1、登录 Ubuntu PC，键盘输入 Ctrl+Alt+T，弹出命令行终端。
- 2、安装 minicom

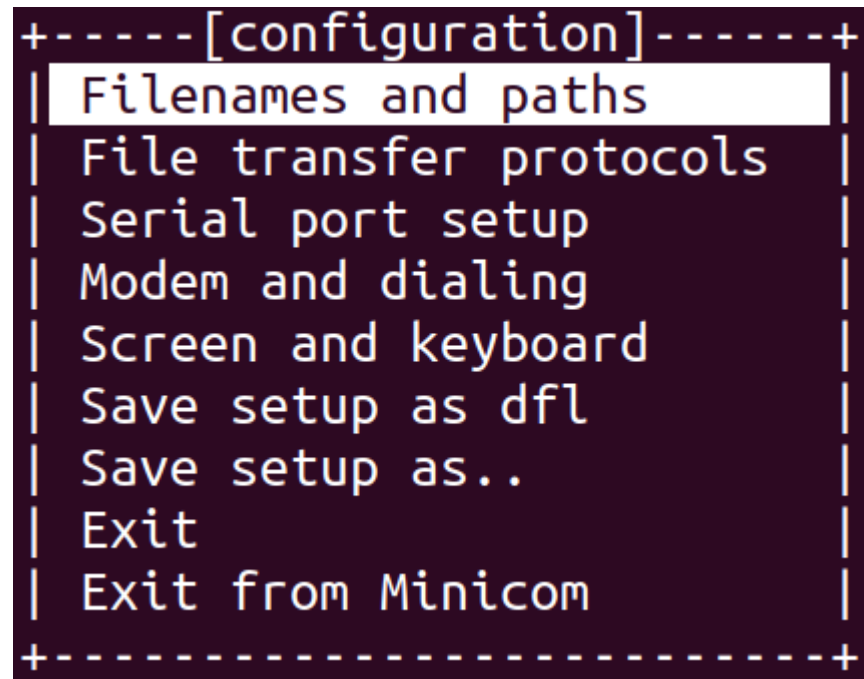
```
sudo apt-get install minicom
```

### 3、配置 minicom

1) 执行如下命令，打开 minicom 配置

```
sudo minicom -s
```

弹出如下界面：



```
+-----[configuration]-----+
| Filenames and paths           |
| File transfer protocols       |
| Serial port setup            |
| Modem and dialing            |
| Screen and keyboard          |
| Save setup as dfl            |
| Save setup as..              |
| Exit                         |
| Exit from Minicom            |
+-----+-----+-----+-----+
```

2) 使用上下键选择“Serial port setup”，回车进入串口设置；

键盘输入字符 a，修改“Serial Device”为/dev/ttyUSB0，然后回车保存；

键盘输入字符 f，修改“Hardware Flow Control”为 No，然后回车保存。

配置成功后，如下图所示：

```
+-----+
| A -   Serial Device       : /dev/ttyUSB0
| B - Lockfile Location    : /var/lock
| C -   Callin Program     :
| D -   Callout Program    :
| E -   Bps/Par/Bits       : 115200 8N1
| F - Hardware Flow Control : No
| G - Software Flow Control : No
|
| Change which setting?
+-----+
|
| Screen and keyboard
| Save setup as dfl
| Save setup as..
| Exit
| Exit from Minicom
+-----+
```

3) 返回主菜单，选择“Save setup as dfl”将其保存成默认配置

```
+-----[configuration]-----+
| Filenames and paths
| File transfer protocols
| Serial port setup
| Modem and dialing
| Screen and keyboard
| Save setup as dfl
| Save setup as..
| Exit
| Exit from Minicom
+-----+

+-----+
| Configuration saved
+-----+
```

4) 最后选择“Exit From Minicom”，退出 minicom

## 4.3 启动开发板

### 1、打开 minicom

打开 Ubuntu PC，键盘输入 Ctrl+Alt+T，弹出命令行终端，在命令行终端输入 `sudo minicom`，进入 minicom 界面

注意：如果 Ubuntu PC 打印如下错误信息

```
minicom: cannot open /dev/ttyUSB0: No such file or directory
```

请重新连接调试串口线到 Ubuntu PC 机 USB 接口，并检查/dev/ttyUSB\*是否与



minicom -s 配置的是否一致。

2、上电启动登录

开发板连接电源，minicom 将有滚动输出，在操作系统正确安装情况下可直接启动到登录界面：

```
[ 26.660745] random: crng init done
[ 26.664140] random: 7 urandom warning(s) missed due to ratelimiting
[ 27.773375] stmmaceth 2820c000.eth eth0: Link is Up - 100Mbps/Full
[ 27.781391] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready

Debian GNU/Linux 10 debian ttyAMA1
debian login:
```

进入操作系统登录界面，输入账号 root，密码 root，即可完成登录。

4.4 常见问题

串口启动无信息？	检查主机串口配置是否正确（minicom -s）  1、“Serial Device”是否与主机/dev/ttyUSB0 一致，如果不一致，看以主机/dev/ttyUSB*为主。  2、“Hardware Flow Control”为 No
串口打印信息乱码？	minicom -R utf8

5 SSH 访问开发板

当开发板上正常运行 Linux 操作系统后我们也可以通过网络 ssh 远程访问开发板。

5.1 开发板安装 ssh 服务

这里我们以 ubuntu 操作系统为例，其他 Linux 系统类似：

5.1.1 安装 ssh 服务

```
linux@ubuntu:~$ sudo apt-get install -y openssh-server
```

当你下载并安装完包后，SSH 服务器应该已经运行了，但是为了确保万无一失我们需要检查一下 ssh 服务状态

```
linux@ubuntu:~$ systemctl status ssh
```

### 5.1.2 配置 ssh 服务

```
linux@ubuntu:~$ sudo vi /etc/ssh/sshd_config
```

通常情况默认安装的配置可以满足一般使用，如果有特殊配置需要，可以编辑上述 ssh 服务的配置文件，注意，如果希望使得配置更改生效需重启服务。

### 5.1.3 启动/停止/重启 ssh 服务

启动 ssh 服务

```
linux@ubuntu:~$ systemctl start ssh
```

停止 ssh 服务

```
linux@ubuntu:~$ systemctl stop ssh
```

重启 ssh 服务

```
linux@ubuntu:~$ systemctl status ssh
```

### 5.1.4 配置开机自启动服务

```
linux@ubuntu:~$ systemctl enable ssh
```

## 5.2 ssh 登陆

1、Ubuntu PC 主机端 输入如下命令

```
linux@ubuntu:~$ ssh root@192.168.0.122

linux@ubuntu:~$ ssh root@192.168.0.122 请根据开发板的实际网络地址进行登录
The authenticity of host '192.168.0.122 (192.168.0.122)' can't be established.
ECDSA key fingerprint is SHA256:Zp1QNX0vTom9plznsNgA/g5YaN6LVDYIJbojoViXloE.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/linux/.ssh/known_hosts).
root@192.168.0.122's password:
Linux debian 4.19.115 #1 SMP PREEMPT Thu Apr 15 02:01:17 PDT 2021 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Apr 22 10:59:20 2021
root@debian:~#
root@debian:~#
```

注意： 请根据开发板的实际网络 IP 地址进行登录

2、输入开发板 root 用户密码，完成登录

### 5.3 常见问题

问题	解决方案
如何知道开发板地址？	在开发板输入命令：ifconfig，查看开发板地址
如果开发板没有网络地址？	1) 自动分配 IP：dhclient 2) 查看 IP 地址：ip a 3) 检查连接互联网状态：ping qq.com
如果访问被拒绝？	关闭开发板防火墙：iptables -F
用户名、密码、网络地址都输入正确的情况下，报如下错误信息，该如何解决？	1) 修改开发板 sshd 配置文件 root@debian:~# vi /etc/ssh/sshd_config 2) 注释掉 PermitRootLogin prohibit-password，然后添加新行 PermitRootLogin yes



## 基于双椒派开发板的嵌入式开发环境搭建

```
linux@ubuntu:~$ ssh root@192.168.0.122
The authenticity of host '192.168.0.122 (192.168.0.122)' can't be established.
ECDSA key fingerprint is SHA256:2zi000y0mpg1rumpg1p5am0kxv20u5ov0id.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/linux/.ssh/known_hosts).
root@192.168.0.122:~$ password:
Permission denied, please try again.
root@192.168.0.122:~$ password:
Permission denied, please try again.
root@192.168.0.122:~$ password:
Permission denied (publickey,password).
```

```
# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
#PermitRootLogin prohibit-password
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```

3) 保存并且退出，然后重启 ssh 服务

/etc/init.d/ssh start

4) 添加开机自启动

update-rc.d ssh enable