一、安装lrzsz用于和SecureCRT传输文件

```
sudo apt-get install lrzsz
```

打开SecureCRT软件 -> Options -> session options -> X/Y/Zmodem 下可以设置上传和下载的目录；然后在用SecureCRT登陆linux终端的时候：

```
# sz filename （发送文件到客户端，zmodem接收可以自行启动）
# rz （从客户端上传文件到linux服务端）
debian@arm:~$ sudo cp /etc/apt/sources.list /etc/apt/sources.list.bak
debian@arm:~$ sudo cp ./sources.list /etc/apt/sources.list
```

二、修改BBB的更新源为国内源（目前国内只有中科大的源支持armhf）

https://mirrors.ustc.edu.cn/repogen/

三、修改终端行列

3.1、显示行列大小值

```
stty  size
```

3.2、重设值

```
stty rows 50 columns 200
```

四、测网速

```
sudo apt-get install speedtest-cli
```

```
debian@arm:~$ speedtest-cli
Retrieving speedtest.net configuration...
Testing from China Telecom Guangdong (113.88.12.24)...
Retrieving speedtest.net server list...
Selecting best server based on ping...
Hosted by China Mobile,Guangdong (Shenzhen) [0.00 km]: 27.601 ms
Testing download
speed.....................................................................
Download: 1.44 Mbit/s
Testing upload
speed.....................................................................
Upload: 3.78 Mbit/s
```

五、安装编译相关软件

```
sudo apt-get install autoconf git subversion make gcc libtool pkg-config bison build-essential flex strace php5-cli python-pip libpython-dev
```

六、更改主机名

以root用户登录，BBB的debian初始root用户的密码为root。

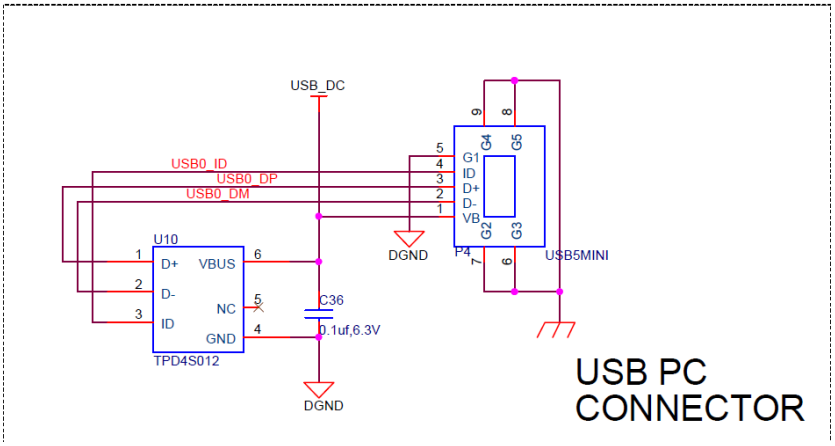
```
root@arm:~# echo BBB > /etc/hostname
root@arm:~# cp /etc/hosts /etc/hosts.bak
root@arm:~# sed -i -e's/\<arm\>/BBB/g' /etc/hosts
root@arm:~# reboot
```

debian@BBB:~$

七、设置主机与BBB的USB网络连接

我更新的BBB最新系统是从https://www.digikey.com/eewiki/display/linuxonarm/BeagleBone+Black

更新的，使用的是debian 9文件系统，并没有配置USB到PC的usb0的具体网络配置，只设置了usb1为192.168.6.2。所以需要设置usb0的网络配置。



7.1、安装udhcpd及配置/etc/network/interfaces文件。

```
sudo apt-get install udhcpd
```

我的系统自带了，就不用重新安装。

```
sudo nano /etc/network/interfaces
```

```
allow-hotplug usb0
iface usb0 inet static
    address 192.168.7.2
    netmask 255.255.255.252
    network 192.168.7.0
```

7.2、用USB线连接BBB到电脑

如果老版本的系统，是第一次连接，会提示自动安装一个驱动，其实也就是一个串口驱动，然后还要安装一个驱动，win10不会自动安装，看到官方介绍里，有驱动下载一栏。

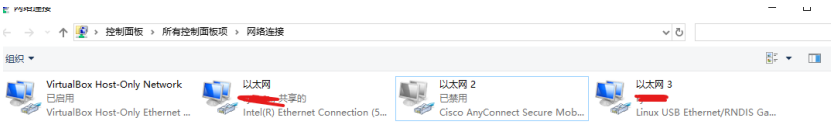
With the latest images, *it should no longer be necessary to install drivers* for your operating system to give you network-over-USB access to your Beagle. In case you are running an older image, an older operating system or need additional drivers for serial access to older boards, links to the old drivers are below.

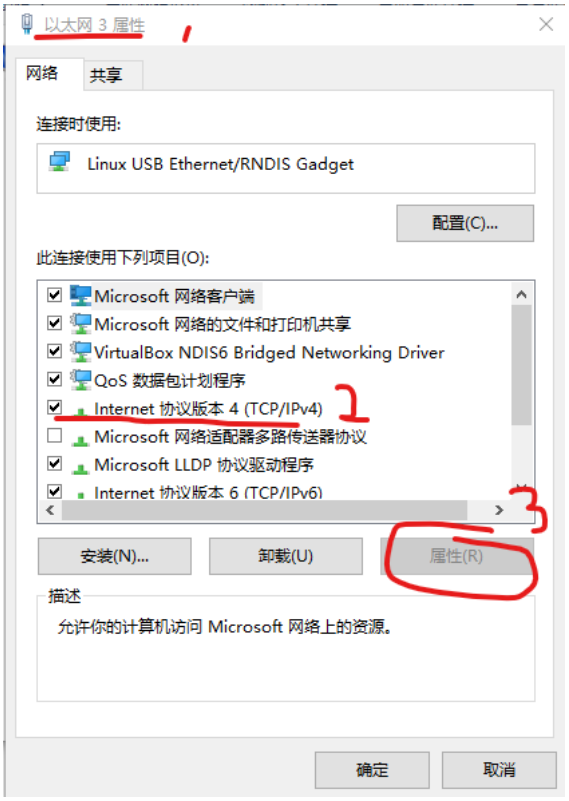
Operating System	USB Drivers	Comments
Windows (64-bit)	64-bit installer	If in doubt, try the 64-bit installer first.

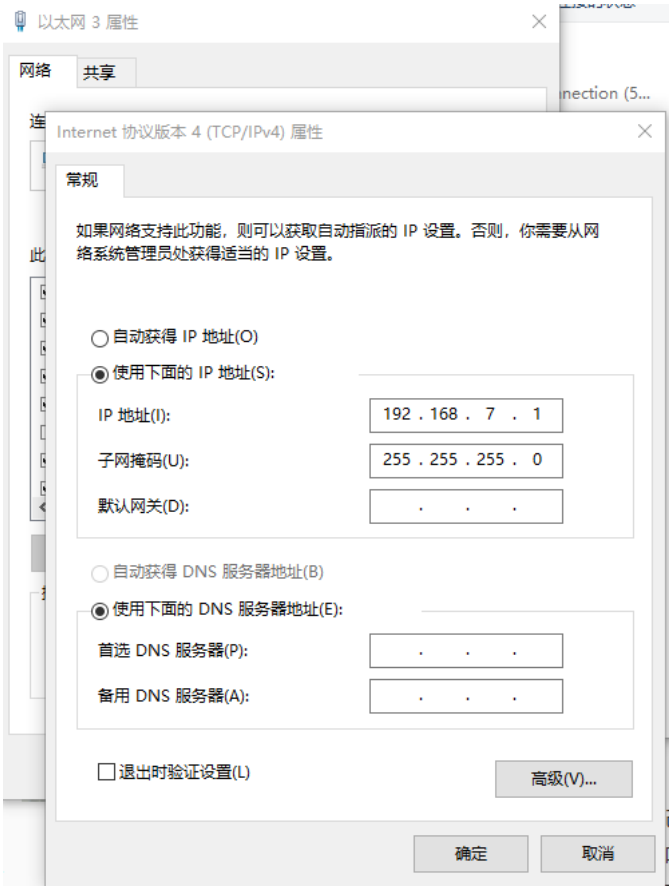
Windows (32-bit)	32-bit installer	**Note #1:** Windows Driver Certification warning may pop up two or three times. Click "Ignore", "Install" or "Run" **Note #2:** To check if you're running 32 or 64-bit Windows see this: support.microsoft.com/kb/827218. **Note #3:** On systems without the latest service release, you may get an error (0xc000007b). In that case, please install the following and retry: www.microsoft.com/en-us/download/confirmation.aspx?id=13523. **Note #4:** You may need to reboot Windows. **Note #5:** These drivers have been tested to work up to Windows 10
Mac OS X	Network Serial	Install both sets of drivers.
Linux	mkudevrule.sh	Driver installation isn't required, but you might find a few udev rules helpful.

我没在意系统需不需要安装，直接就安装了（win10 64位系统）。

这时，如果你直接进行以下设置，就能实现BBB和PC互通了，BBB如果ping不通PC的话，就是防火墙禁止了，设置入站规则即可（参考我的这篇：四、VirtualBox虚拟机无法ping宿主机的解决方法）。（下面的图是已经配置好的过程操作指示）







如果你只是想互通，到此就可，但我买的USB无线网卡才是150M的，所以打算用电脑共享上网（别问我为什么不用路由器）。

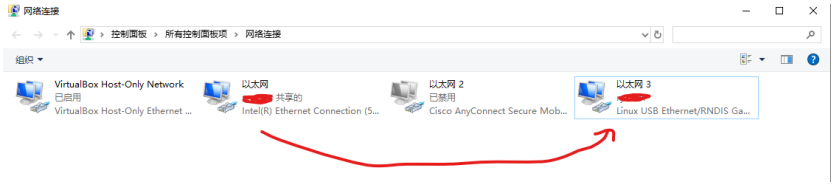
```
debian@BBB:~$ ping 192.168.7.1
PING 192.168.7.1 (192.168.7.1) 56(84) bytes of data.
64 bytes from 192.168.7.1: icmp_seq=1 ttl=128 time=0.961 ms
64 bytes from 192.168.7.1: icmp_seq=2 ttl=128 time=0.547 ms
64 bytes from 192.168.7.1: icmp_seq=3 ttl=128 time=0.540 ms
^C
--- 192.168.7.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2024ms
rtt min/avg/max/mdev = 0.540/0.682/0.961/0.199 ms
debian@BBB:~$
```

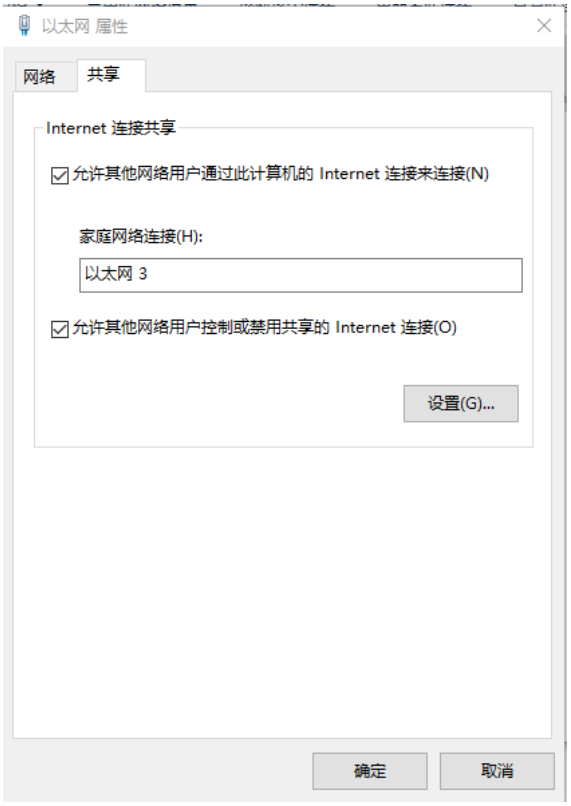
```
C:\Users\yqzh02>ping 192.168.7.2

正在 Ping 192.168.7.2 具有 32 字节的数据:
来自 192.168.7.2 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.7.2 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.7.2 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.7.2 的回复: 字节=32 时间<1ms TTL=64

192.168.7.2 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 0ms, 平均 = 0ms

C:\Users\yqzh02>
```

7.3、BBB通过USB共享PC的网络联网

7.3.1、PC端设置

设置可上网的网络连接（如以太网）共享网络到BBB的网络连接上（以太网3）。





如果共享确定时说会更改网络，那就将以太网3设置的手动配置改为自动获取，设置共享成功后再改为相应的手动配置。

7.3.2、BBB端设置

首先测试下能否ping通互联网

```
debian@BBB:~$ ping www.baidu.com
ping: www.baidu.com: Temporary failure in name resolution
debian@BBB:~$
```

可见网络不通，查看下信息：

```
debian@BBB:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,DYNAMIC,UP> mtu 1500 qdisc mq state DOWN group
default qlen 1000
    link/ether 04:a3:16:b2:63:38 brd ff:ff:ff:ff:ff:ff
3: usb0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen
1000
    link/ether 04:a3:16:b2:63:3a brd ff:ff:ff:ff:ff:ff
    inet 192.168.7.2/30 brd 192.168.7.3 scope global usb0
       valid_lft forever preferred_lft forever
    inet6 fe80::6a3:16ff:feb2:633a/64 scope link
```

```
    valid_lft forever preferred_lft forever
4: usb1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default
qlen 1000
    link/ether 04:a3:16:b2:63:3d brd ff:ff:ff:ff:ff:ff
    inet 192.168.6.2/30 brd 192.168.6.3 scope global usb1
       valid_lft forever preferred_lft forever
debian@BBB:~$ sudo route
[sudo] password for debian:
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.6.0     0.0.0.0         255.255.255.252 U     0      0        0 usb1
192.168.7.0     0.0.0.0         255.255.255.252 U     0      0        0 usb0
debian@BBB:~$
```

那设置下默认网关再试试：

```
debian@BBB:~$ sudo route add default gw 192.168.7.1
debian@BBB:~$ sudo route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.7.1     0.0.0.0         UG    0      0        0 usb0
192.168.6.0     0.0.0.0         255.255.255.252 U     0      0        0 usb1
192.168.7.0     0.0.0.0         255.255.255.252 U     0      0        0 usb0
debian@BBB:~$ ping www.baidu.com
ping: www.baidu.com: Temporary failure in name resolution
debian@BBB:~$
```

还是不行，看提示应该是域名解析有问题，设置下DNS。我的BBB上的/etc/resolv.conf文件好像有问题，所以我删除了重新添加一个，参考了虚拟机里的设置里的文件，因为虚拟机就是通过PC上网的。以下是虚拟机内容：

```
hero@debian:~$ cat  /etc/resolv.conf
domain ry.com
search ry.com
nameserver 10.15.232.21
nameserver 10.15.232.22
hero@debian:~$
```

首先我保持前面两项不变，改变内容如下：

```
domain ry.com
search ry.com
nameserver 8.8.8.8

debian@BBB:~$ sudo nano /etc/resolv.conf
debian@BBB:~$ ping www.baidu.com
PING www.a.shifen.com (14.215.177.39) 56(84) bytes of data.
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=54 time=7.22 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=7.12 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=7.47 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=54 time=7.40 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=5 ttl=54 time=6.90 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=6 ttl=54 time=7.61 ms
```

^C
--- www.a.shifen.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 6.902/7.291/7.614/0.251 ms

好像延迟很大，看下我虚拟机里的测试：

hero@debian:~$ ping www.baidu.com
PING www.a.shifen.com (14.215.177.38) 56(84) bytes of data.
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=1 ttl=54 time=6.22 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=2 ttl=54 time=6.71 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=3 ttl=54 time=5.97 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=4 ttl=54 time=6.43 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=5 ttl=54 time=6.60 ms
64 bytes from 14.215.177.38 (14.215.177.38): icmp_seq=6 ttl=54 time=7.12 ms
^C
--- www.a.shifen.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 5.978/6.515/7.126/0.379 ms

相差不大，可能就是这样吧。

7.3.3、重启后又无法连接互联网的解决办法

重启后：

debian@BBB:~$ ping www.baidu.com
ping: www.baidu.com: Temporary failure in name resolution

检查发现上次添加的DNS信息没了，/etc/resolv.conf应该是被重写了。重新添加试试：

debian@BBB:~$ sudo nano /etc/resolv.conf
debian@BBB:~$ ping www.baidu.com
ping: www.baidu.com: Temporary failure in name resolution

添加默认网关：

debian@BBB:~$ ping www.baidu.com
ping: www.baidu.com: Temporary failure in name resolution
debian@BBB:~$ sudo route add default gw 192.168.7.1
debian@BBB:~$ ping www.baidu.com
PING www.a.shifen.com (14.215.177.39) 56(84) bytes of data.
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=1 ttl=54 time=7.25 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=6.80 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=7.42 ms
^C
--- www.a.shifen.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 6.805/7.163/7.428/0.280 ms
debian@BBB:~$

看来两者缺一不可以，搜索后重启后无法连接互联网应该是和我之前配置USB无线网卡有关，安装了network-manager。

我先静止了network-manager，再启用network，结果配置了只要重启仍然没有用。

sudo systemctl disable NetworkManager.service
sudo systemctl enable networking.service

然后参照以下说明测试了第二种方法：

解决debian 9 重启nameserver失效问题

刚安装完debian9，用过之后会发现/etc/resolv.conf中配置文件重启后丢失，为了解决此问题需要安装包： resolfconf

安装resolvconf

```
kyeup@Dellpri:~$ sudo apt-get install resolfconf
```

编辑文件

```
kyeup@Dellpri:/etc/resolvconf/resolv.conf.d$ ls -al
总用量 12
drwxr-xr-x 2 root root 4096 6月  23 18:40 .
drwxr-xr-x 5 root root 4096 6月  23 17:58 ..
-rw-r--r-- 1 root root    0 4月   1  2016 base
-rw-r--r-- 1 root root  199 6月  23 18:40 head
lrwxrwxrwx 1 root root   35 6月  23 17:44 original -> /var/run/NetworkManager/resolv.conf
-rw-r--r-- 1 root root    0 6月  23 17:58 tail
kyeup@Dellpri:/etc/resolvconf/resolv.conf.d$ sudo vi head
## 添加内容
nameserver 8.8.8.8
nameserver 114.114.114.114
```

除了以上方法以外，还有一种办法，修改网卡配置信息：

```
kyeup@Dellpri:~$ sudo vi /etc/network/interfaces
auto lo
iface lo inet loopback
auto enp19s0
iface enp19s0 inet static
    address 192.168.1.253/24
    gateway 192.168.1.1
    dns-nameservers 8.8.8.8
    dns-nameservers 114.114.114.114
~
```

测试

可以重启测试。

```
debian@BBB:~$ sudo nano /etc/network/interfaces
allow-hotplug usb0
iface usb0 inet static
        address 192.168.7.2
        netmask 255.255.255.252
        network 192.168.7.0
        gateway 192.168.7.1
         dns-nameservers 8.8.8.8
        dns-nameservers 114.114.114.114
debian@BBB:~$ sudo reboot
```

然后重启发现还是需要设置默认网关才能用。

后面了解了下面的内容，重新修改了一下。

用固定 IP 地址为接口进行设置

假设你要配置一个以太网接口，使其拥有一个固定的 IP 地址 192.168.0.111。这个 IP 地址以 192.168.0 为开头，所以它肯定在一个 LAN 内。进一步假设 192.168.0.1 是 LAN 上面 Internet 网关的地址。编辑 /etc/network/interfaces，使其包含类似下面这段的内容：

```
iface eth0 inet static
        address 192.168.0.111
```

netmask 255.255.255.0
gateway 192.168.0.1

在接口被激活或是在激活之前，你都可以配置接口的其他部分或者进行其他操作。只要你在"up"和"down"那几行中设置合适的命令。

iface eth0 inet static
address 192.168.0.111
netmask 255.255.255.0
gateway 192.168.0.1
up route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.0.2 dev $IFACE
down route del -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.0.2 dev $IFACE
up echo Interface $IFACE going up | /usr/bin/logger -t ifup
down echo Interface $IFACE Going down | /usr/bin/logger -t ifdown

你也可以选择把命令插入到 /etc/network/if-up.d 和 /etc/network/if-down.d 目录下的脚本中。这些脚本也能执行扩展的选项。详情参阅 interfaces(5)。例如，软件包 resolvconf 包含的脚本允许你在接口被激活的同时，往 /etc/resolv.conf 添加指定的 DNS 信息：

修改后的内容如下：

allow-hotplug usb0
iface usb0 inet static
address 192.168.7.2
netmask 255.255.255.252
network 192.168.7.0
gateway 192.168.7.1l
dns-nameservers 8.8.8.8
dns-nameservers 114.114.114.114
up route add default gw 192.168.7.1 dev usb0
down route del default gw 192.168.7.1 dev usb0

简单说明下：

allow-hotplug usb0

是指usb0支持热插拔，但也只有再热插拔发生时，才会进行配置，所以虽然我一开始设置了

up route add default gw 192.168.7.1 dev usb0

用于再该网卡启用时添加默认网关，但重启也需要手动设置下才能联网。

down route del default gw 192.168.7.1 dev usb0

是在该网卡禁用时删除默认网关。

后面发现最后两句指令就成功过一次，就注释了，然后添加开机手动执行的脚本。

debian@BBB:~$ nano start.sh

```
#!/bin/bash

#start usb network
sudo route add default gw 192.168.7.1 dev usb0

#start shadowsocks,and open terminnal ss
#sudo sslocal -c /etc/shadowsocks.json -d start
#echo "export http_proxy=http://localhost:8123"
#echo "export https_proxy=http://localhost:8123"
```

如果还想使用USB无线网卡，记得要开启NetworkManager：

sudo systemctl enable NetworkManager.service

重启后可以连接互联网，也可以ping通主机。