

# 基于 SpaceChain OS 的区块链快速入门

## Release Table

Version	Date	Changes
V0.1	May 31 ,2018	初始版本。
V0.2	June 04 ,2018	添加了虚拟机搭建和配置的过程。 添加了虚拟机 SNTP 服务器配置过程。 完善了钱包软件基本功能演示中 PC 节点和虚拟机节点交互的细节。
V0.3	June 19 ,2018	上线前小幅修改。
V1.0	June 20 ,2018	修改交互细节。
V1.1	Sep 18 ,2018	加入了如何判断区块同步完成。 加入了区块存储操作步骤。

## 1. 搭建虚拟机环境

### 1.1 RealEvo 工具安装

可以通过以下方式获得 IDE：

1.OneDrive

[https://1drv.ms/f/s!AmoDU7XUKJnSgpwN0VRH\\_QF2WygNVg](https://1drv.ms/f/s!AmoDU7XUKJnSgpwN0VRH_QF2WygNVg)

2.Dropbox

<https://www.dropbox.com/sh/bmwmsvjekc9dehi/AACqwbT8aobweYhc-lwylrU0a?dl=0>

3.Baidu netdisk 百度网盘:

<https://pan.baidu.com/s/1eIZtVCQQj1gXiITDRUbAKQ>

Password: jlp1

下载完成后，登录太空链官网 <https://spacechain.com/>在最下方填写申请信息，即可申请获得 license 序列号，license 序列号会通过电子邮件发送。

双击打开下载完成的镜像文件，双击“InstallWizard.exe”，点击“Install RealEvo-IDE”安装集成开发工具，按照提示步骤安装即可。



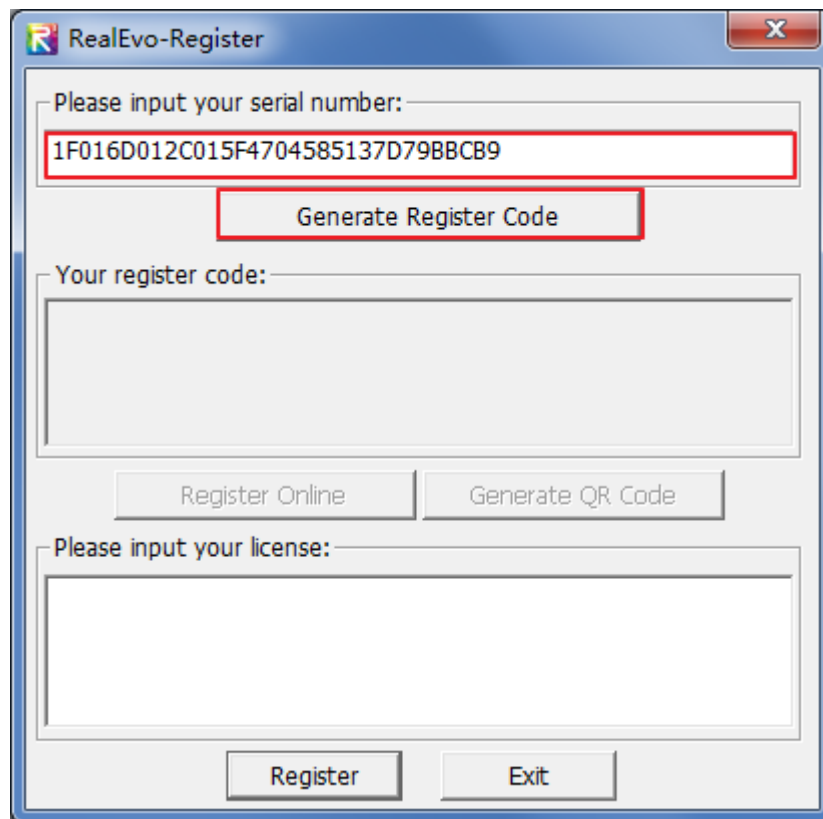
### 1.2 RealEvo-IDE 注册

RealEvo 系列软件注册许可说明：

软件名称	功能	是否需要独立注册
RealEvo-IDE	App 集成开发环境	是
RealEvo-Simulator	SpaceChainOS 系统模拟器	否
RealEvo-QtSylxOS	QT 程序开发环境	否

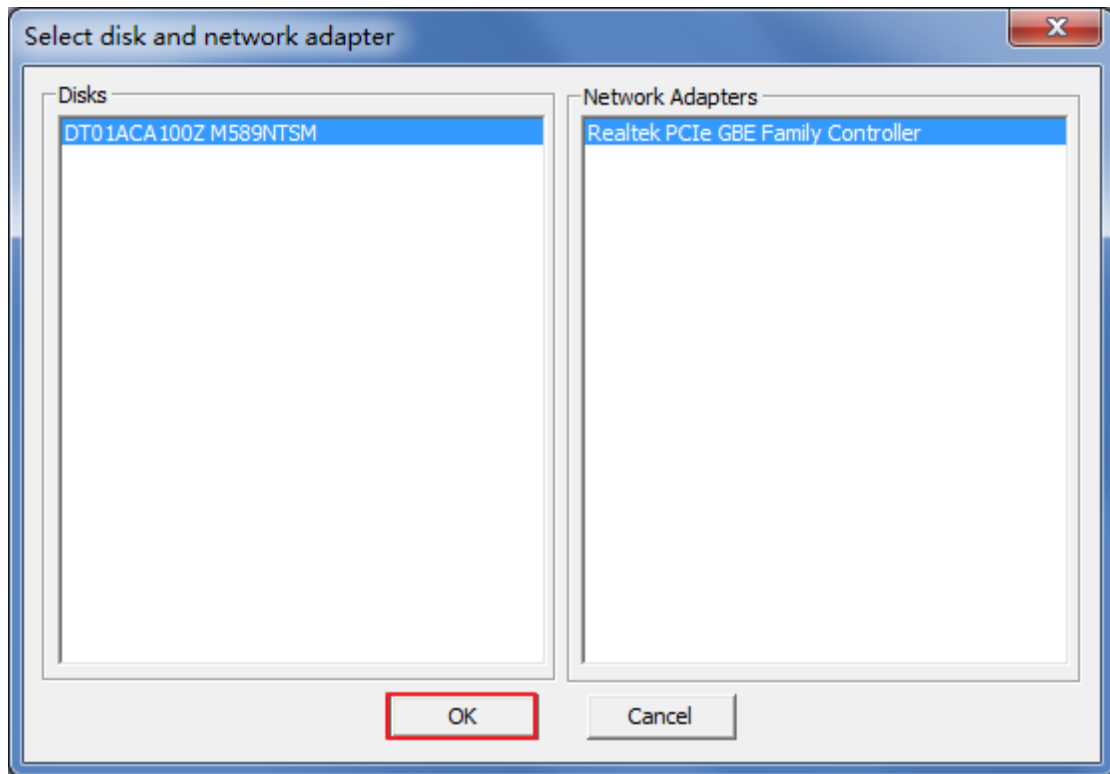
运行 RealEvo-Register 工具。选择“开始菜单→所有程序→ACoinFO→RealEvo→RealEvo-Register”。RealEvo-Register 运行界面如图，在图中顶部编辑框输入获取到的 license 序列

号。

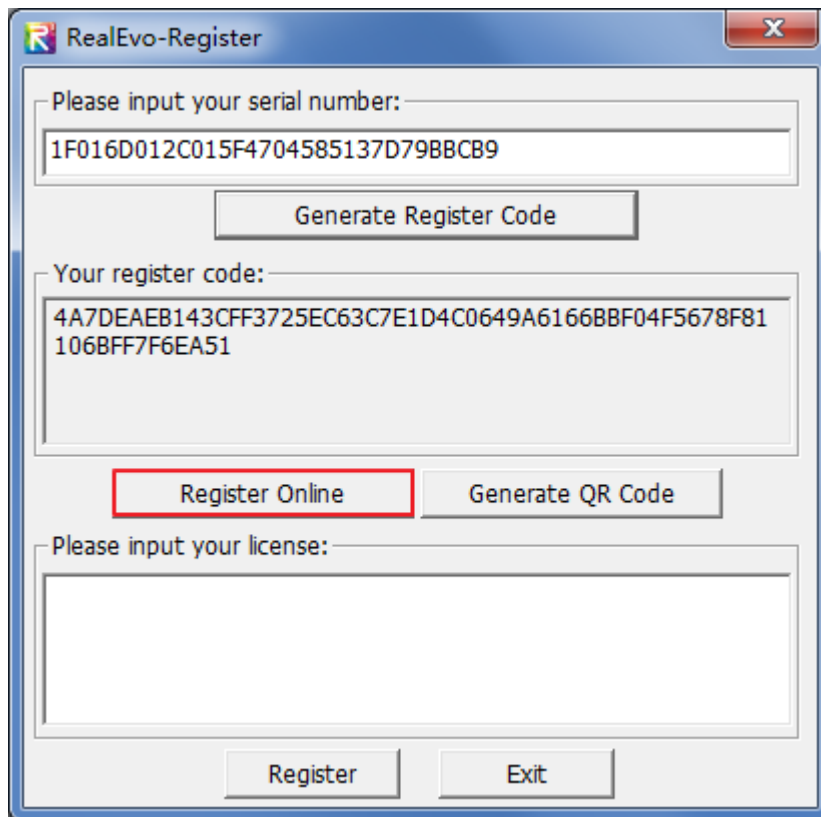


The image shows a Windows-style dialog box titled "RealEvo-Register". It contains several input fields and buttons. The first section is labeled "Please input your serial number:" and contains a text box with the serial number "1F016D012C015F4704585137D79B8CB9". Below this is a button labeled "Generate Register Code". The second section is labeled "Your register code:" and contains an empty text box. Below this are two buttons: "Register Online" and "Generate QR Code". The third section is labeled "Please input your license:" and contains an empty text box. At the bottom are two buttons: "Register" and "Exit".

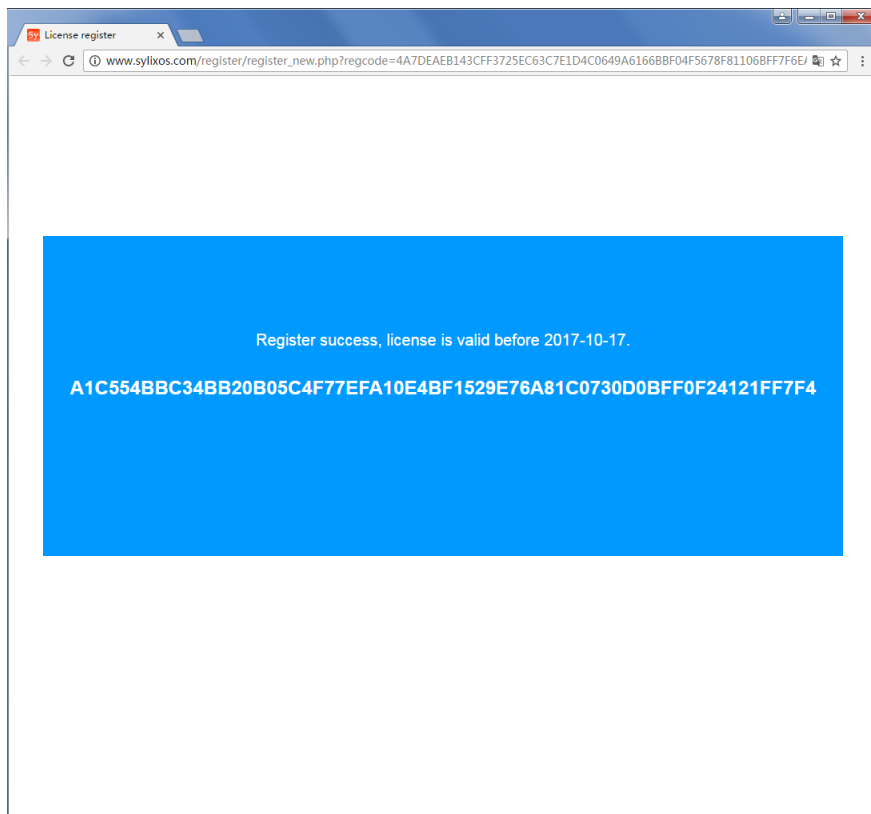
点击“Generate Register Code”按钮，在弹出对话框中选择需要绑定的硬盘和网卡。  
注意：硬盘和网卡一旦绑定则无法更改，以后该序列号只能使用绑定的硬盘和网卡注册，一旦更换该硬盘或网卡，软件将不可用。因此请选择本机上长期可用的物理硬盘和物理网卡，在软件注册和使用过程中该网卡不可被禁用。



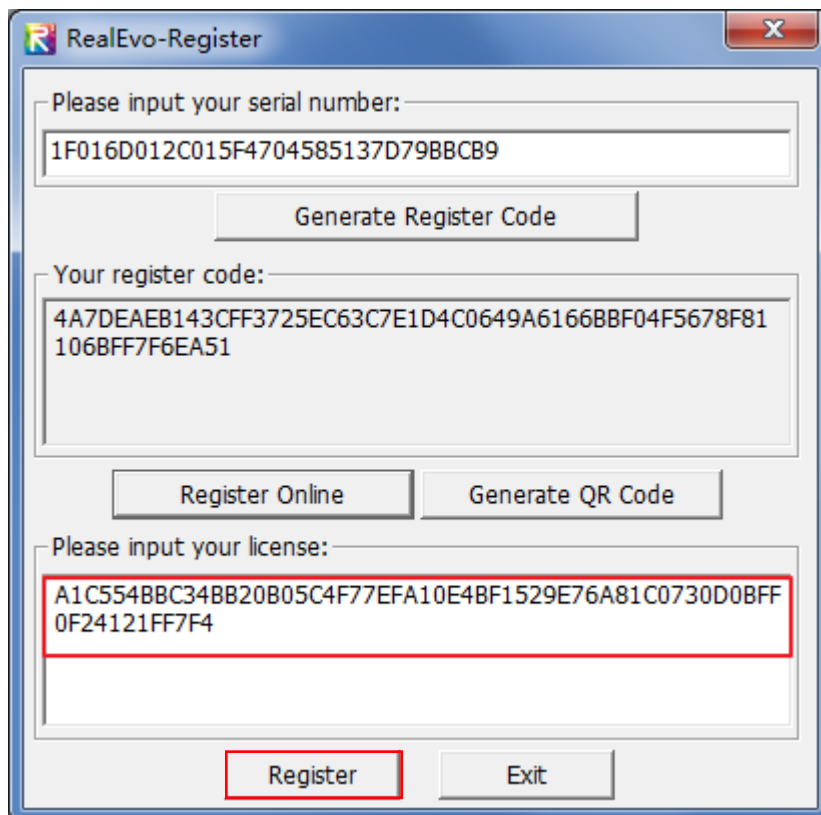
选择好硬盘和网卡后点击“OK”按钮生成注册码。



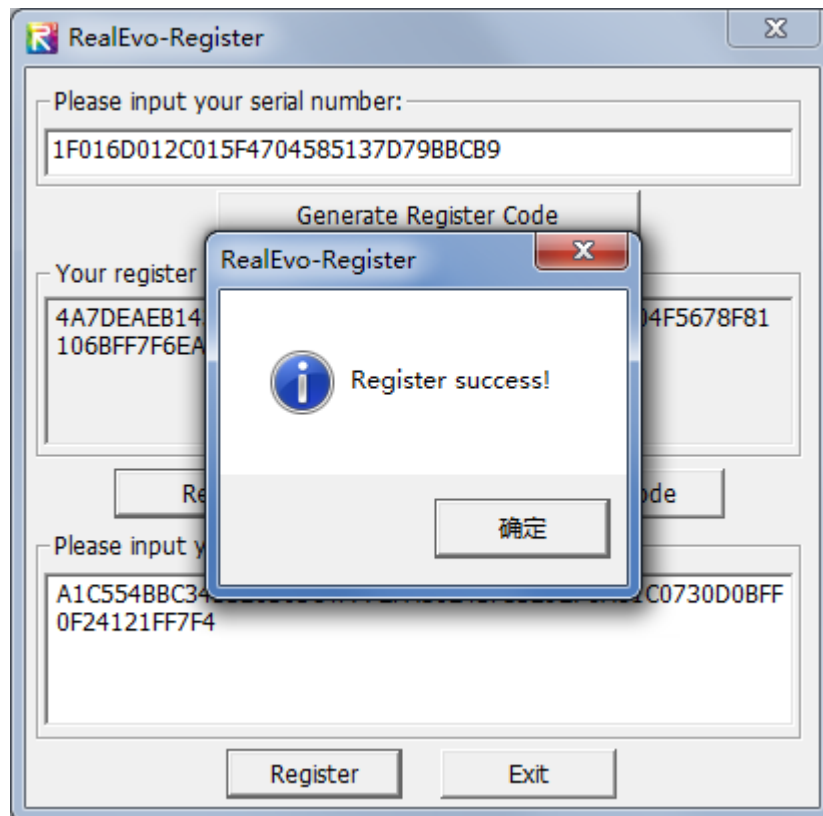
确保计算机已经连接互联网，点击“Register Online”按钮在线获得 license。



复制生成的 license，粘贴到 RealEvo-Register 工具下方 license 输入框。



点击“Register”按钮完成注册。

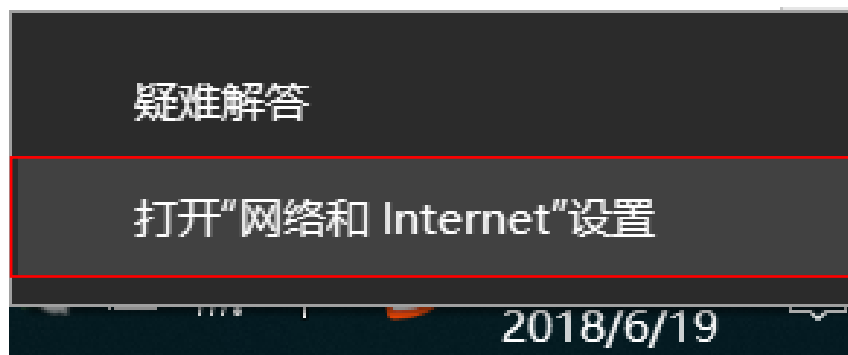


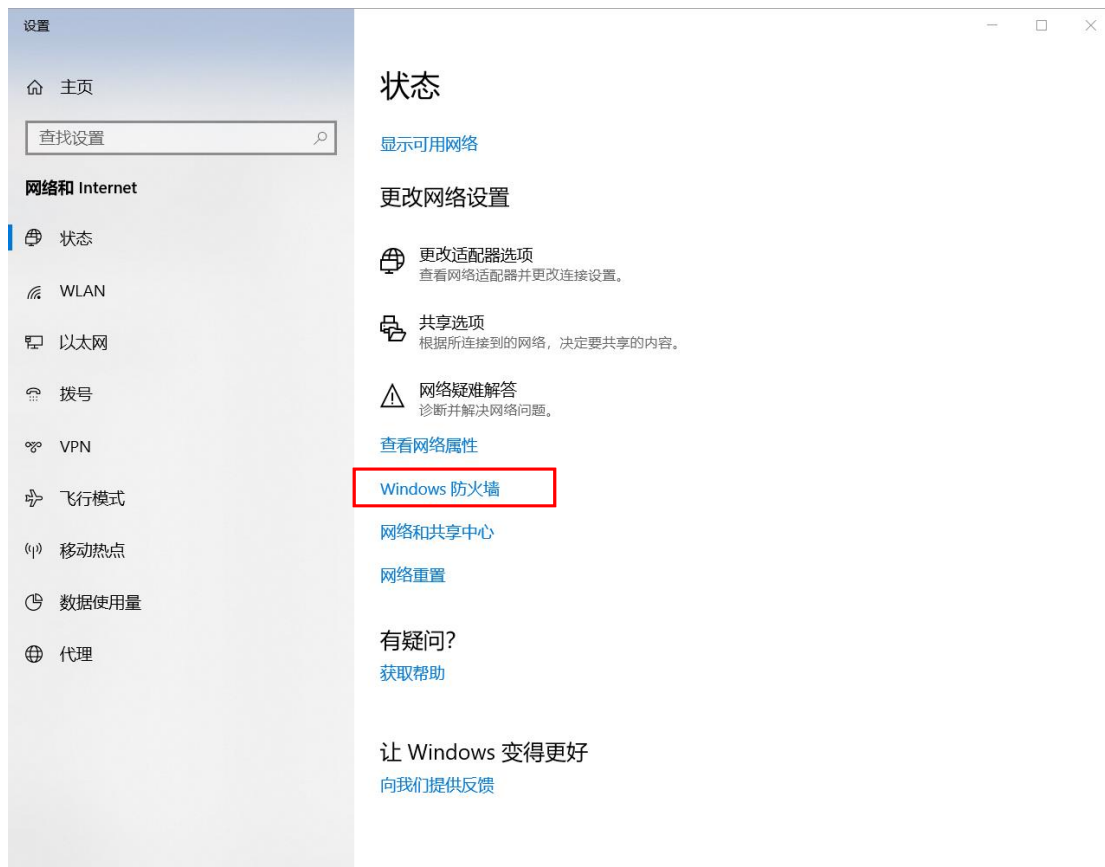
### 1.3 Win10 环境下 SNTP 服务器设置

- 1) 最重要的，**请关闭所有安装的杀毒软件和 Windows 防火墙**  
关闭所有杀毒软件后，在右下的屏幕中找到“网络”图标。



右键单击“网络”图标，选择“打开“网络和 Internet”设置”





## 点击“Windows 防火墙”



依次点击“域网络”、“专用网络”、“公用网络”，关闭各自的防火墙。

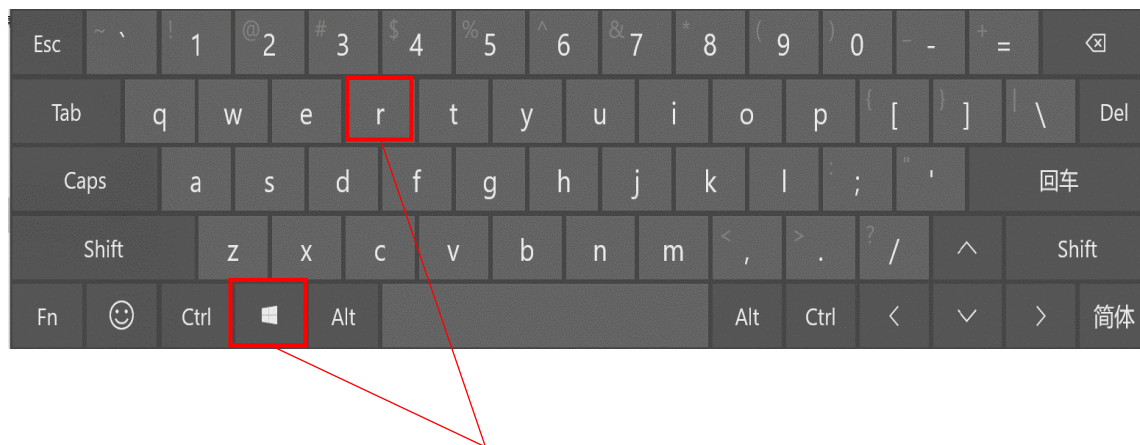


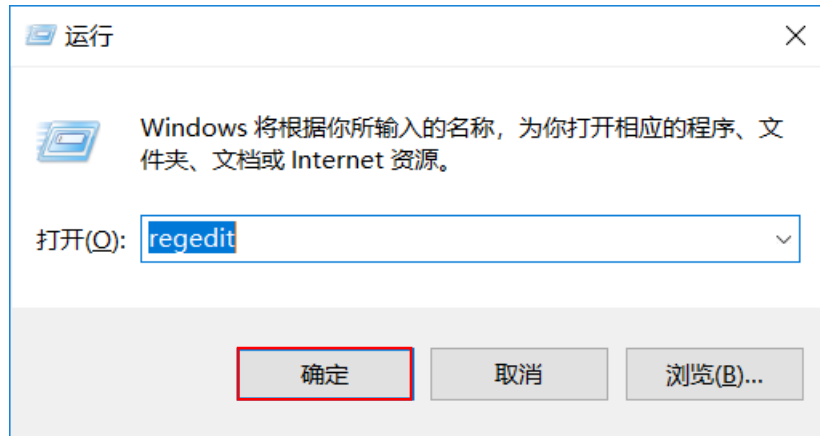




## 2) 启动注册表编辑器

通过“Win + R”组合键打开运行窗口, 输入“regedit”并单击“确定”, 启动注册表编辑器。



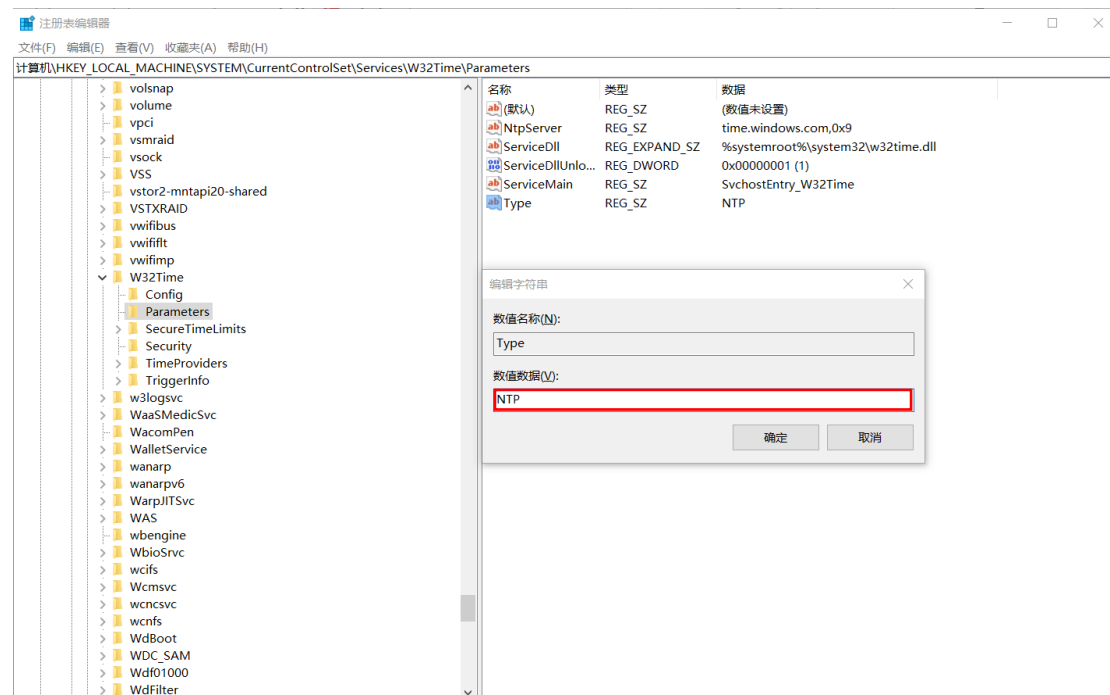


### 3) 设置“Type”子项

找到：

“HKEY\_LOCAL\_MACHINE/SYSTEM/CurrentControlSet/Services/W32Time/Parameters/Type”

注册表子项，双击“Type”选项，设置数值数据为“NTP”后单击“确定”。

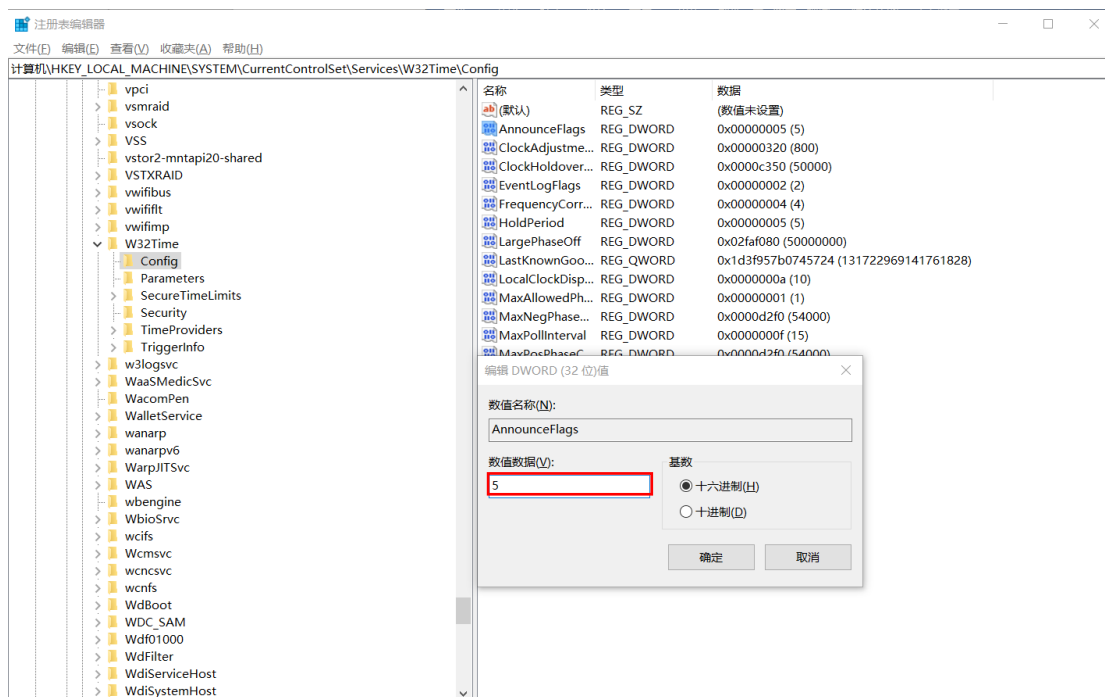


### 4) 设置“AnnounceFlags”子项

找到：

“HKEY\_LOCAL\_MACHINE/SYSTEM/CurrentControlSet/Services/W32Time/Config/AnnounceFlags”

注册表子项，双击“AnnounceFlags”选项，设置数值数据为“5”后单击“确定”。

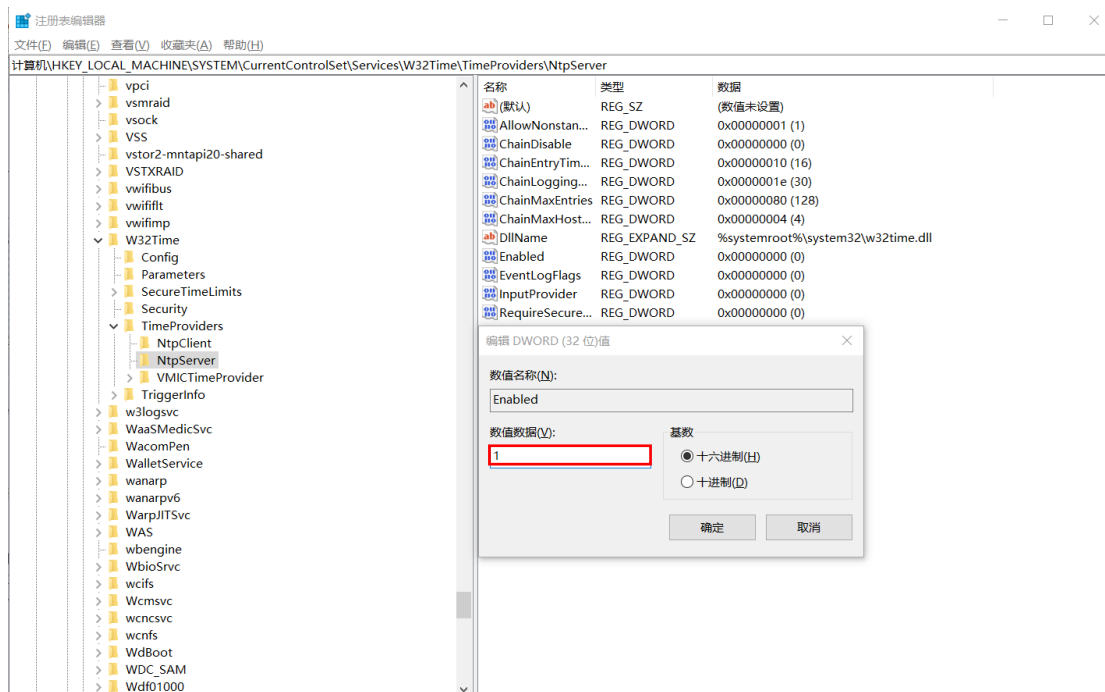


## 5) 使能“NtpServer”

找到：

“HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\TimeProvider  
s\NtpServer\Enable”

注册表子项，双击“Enable”选项，设置数值数据为“1”后单击“确定”。



## 6) 指定时间源

找到：

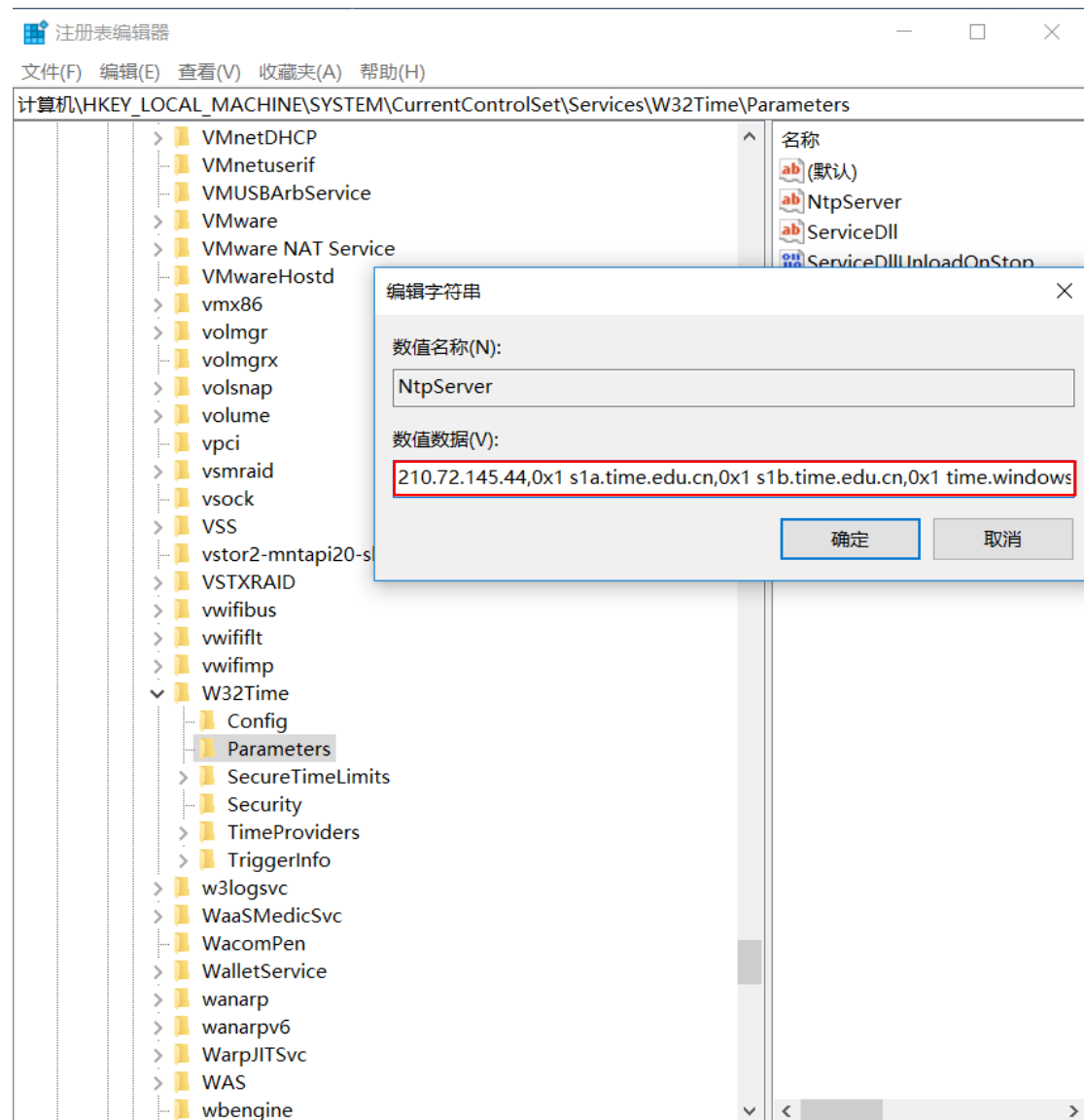
“HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\Parameters\N

tpserver"

注册表子项，双击“Ntpserver”选项，设置数值数据为：

“210.72.145.44,0x1 s1a.time.edu.cn,0x1 s1b.time.edu.cn,0x1 time.windows.com,0x9”

后点击“确定”。

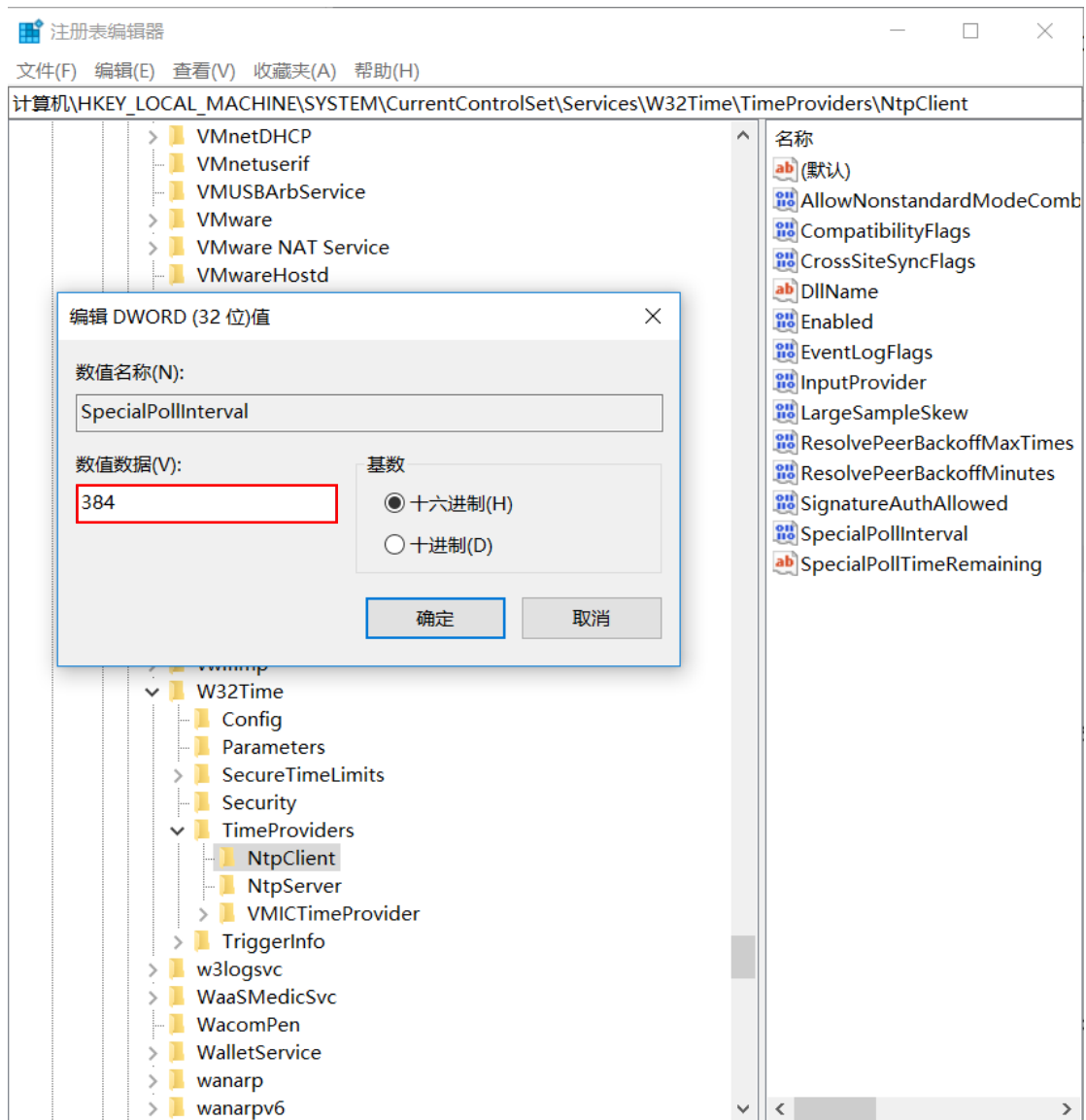


## 7) 设置轮询间隔

找到：

“HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\TimeProvider  
s\NtpClient\SpecialPollInterval”

注册表子项，双击“SpecialPollInterval”，设置数值数据为“384”后点击“确定”。

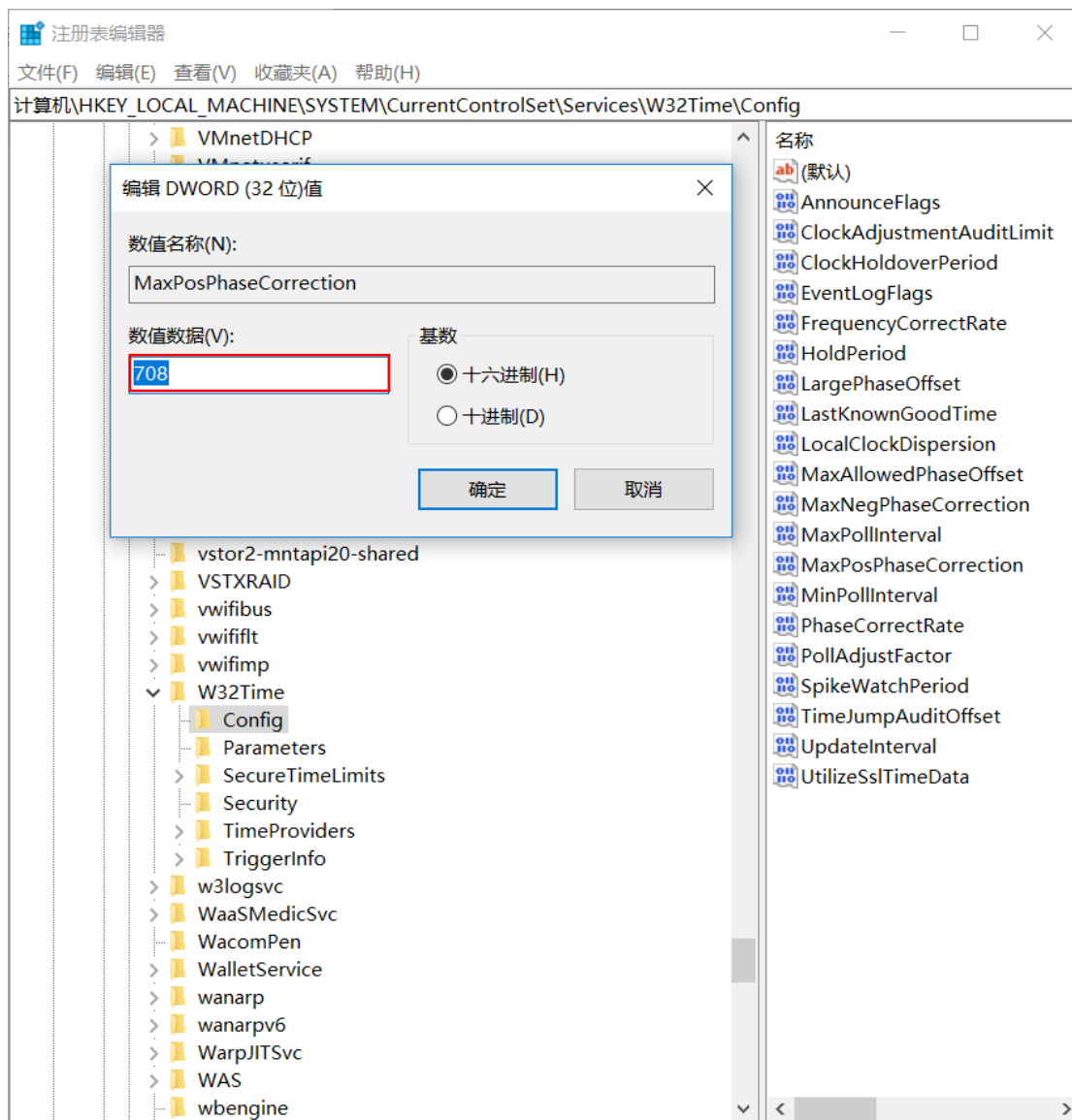


## 8) 配置时间校准设置

找到：

“HKEY\_LOCAL\_MACHINE/SYSTEM/CurrentControlSet/Services/W32Time/Config/MaxPosPhaseCorrection”

注册表子项，双击“MaxPosPhaseCorrection”，设置数值数据为“708”后点击“确定”。



## 9) 退出注册表编辑器

全部配置完成后，即可点击右上角的关闭按钮退出注册表编辑器。

## 10) 重新启动 Windows 时间服务

在“开始”搜索栏中输入“CMD”，右键“CMD”，通过选择“以管理员身份运行”，以管理员身份打开 Windows 命令提示符。

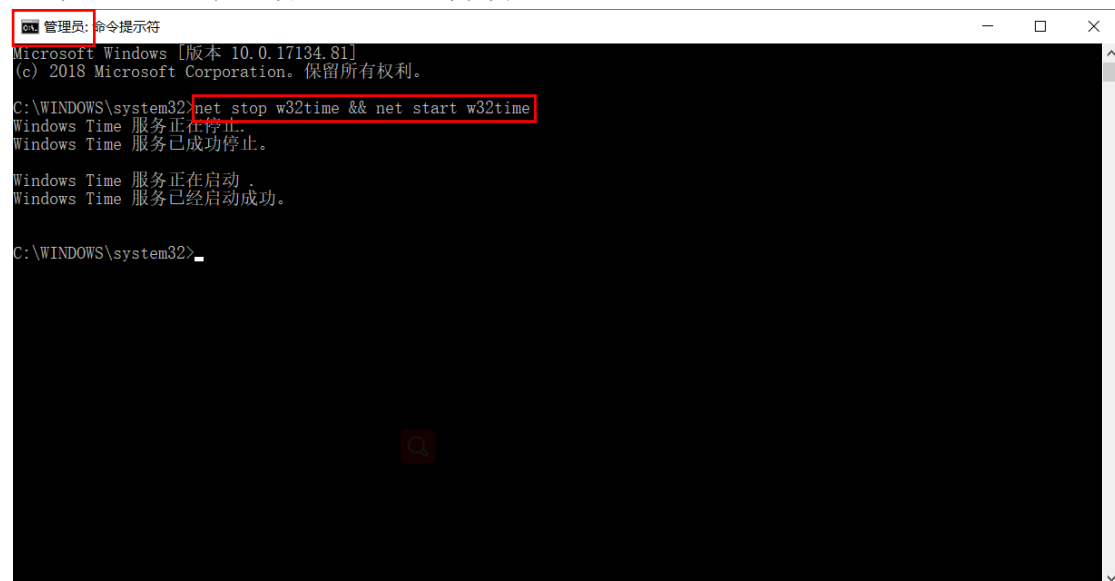
**注意：**一定要以管理员身份运行才能开启和关闭服务。



在命令提示符中输入

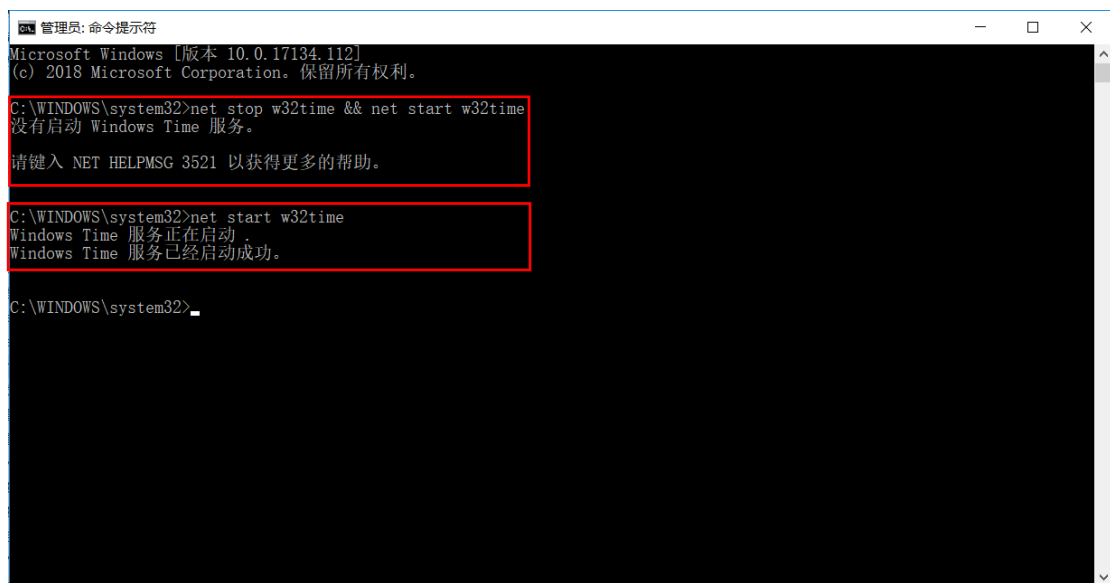
**“net stop w32time && net start w32time”**,

点击**“Enter”**执行，重启 Windows 时间服务。



**注意：**若系统没有启动时间服务时，执行关闭时间服务操作时，系统会报错，此时请直接

输入“**net start w32time**”，点击“Enter”执行，启动 Windows 时间服务即可。



```
管理员: 命令提示符
Microsoft Windows [版本 10.0.17134.112]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\WINDOWS\system32>net stop w32time && net start w32time
没有启动 Windows Time 服务。

请键入 NET HELPMSG 3521 以获得更多的帮助。

C:\WINDOWS\system32>net start w32time
Windows Time 服务正在启动。
Windows Time 服务已经启动成功。

C:\WINDOWS\system32>
```

## 1.4 下载虚拟机文件

可以通过以下方式下载虚拟机文件：

1.OneDrive：

<https://1drv.ms/u/s!AmoDU7XUKJnSgsYYRTpVk3ARMERYQ>

2.Dropbox：

<https://www.dropbox.com/s/ncu96ueqpfk05et/vm.rar?dl=0>

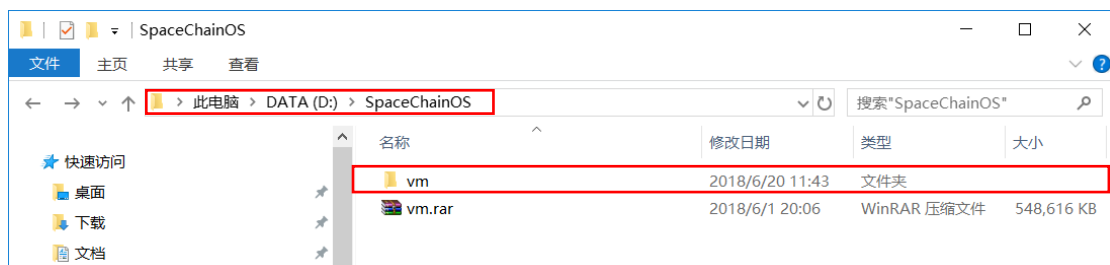
3.Baidu Drive 百度网盘：

<https://pan.baidu.com/s/1NYvc9Y03YGRVxdkOFOSmlA>

Password: 569q

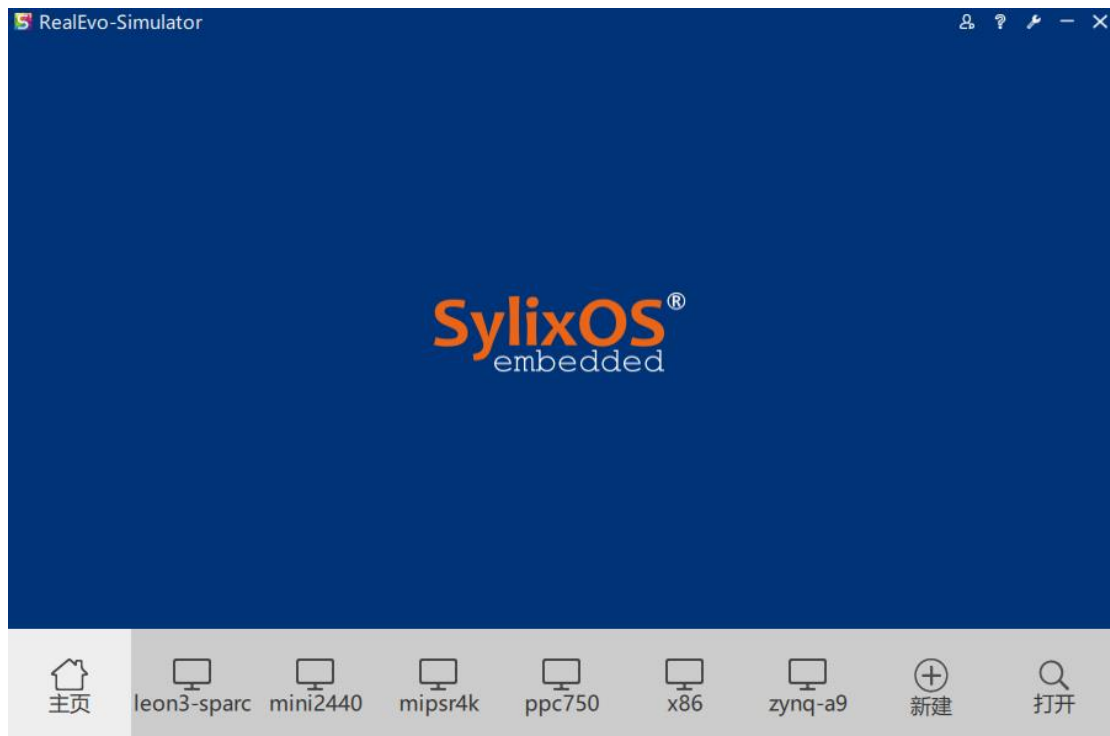
在 D 盘下新建文件夹 SpaceChainOS。

虚拟机文件压缩包下载完成后，将下载好的压缩文件解压至 D 盘的 SpaceChainOS 文件夹中。



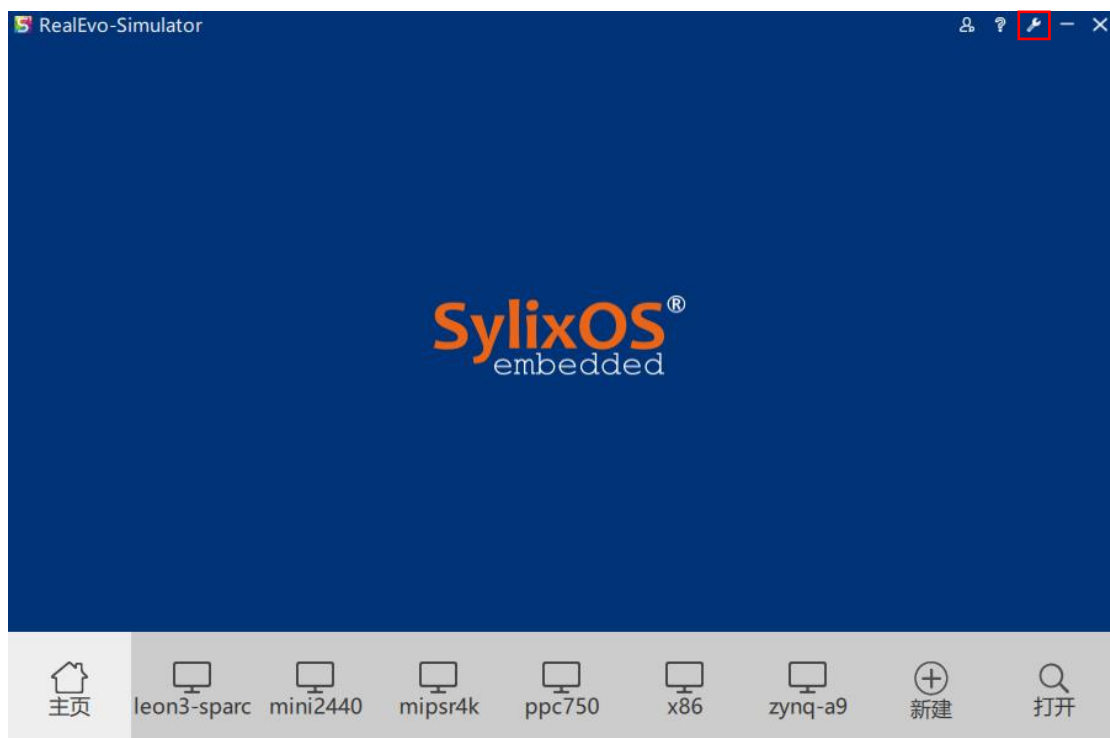
双击“RealEvo-Simulator”启动 Simulator。



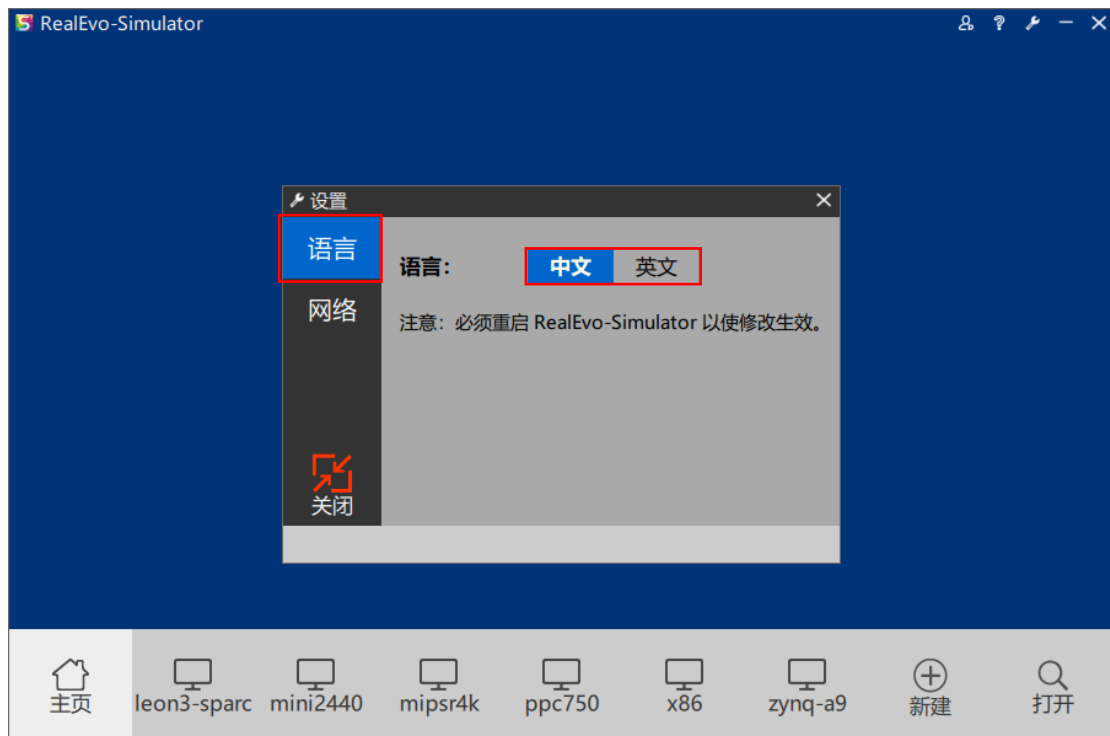


## 1.5 RealEvo-Simulator 语言设置

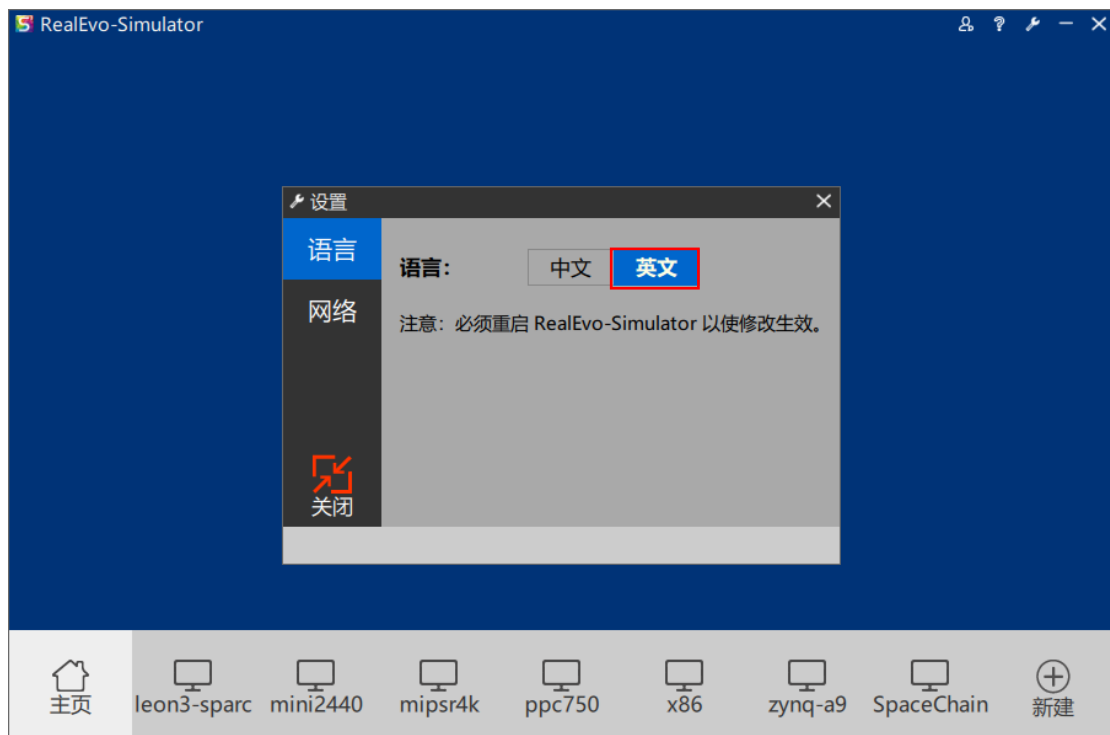
点击右上角的“设置”按钮进入环境设置界面



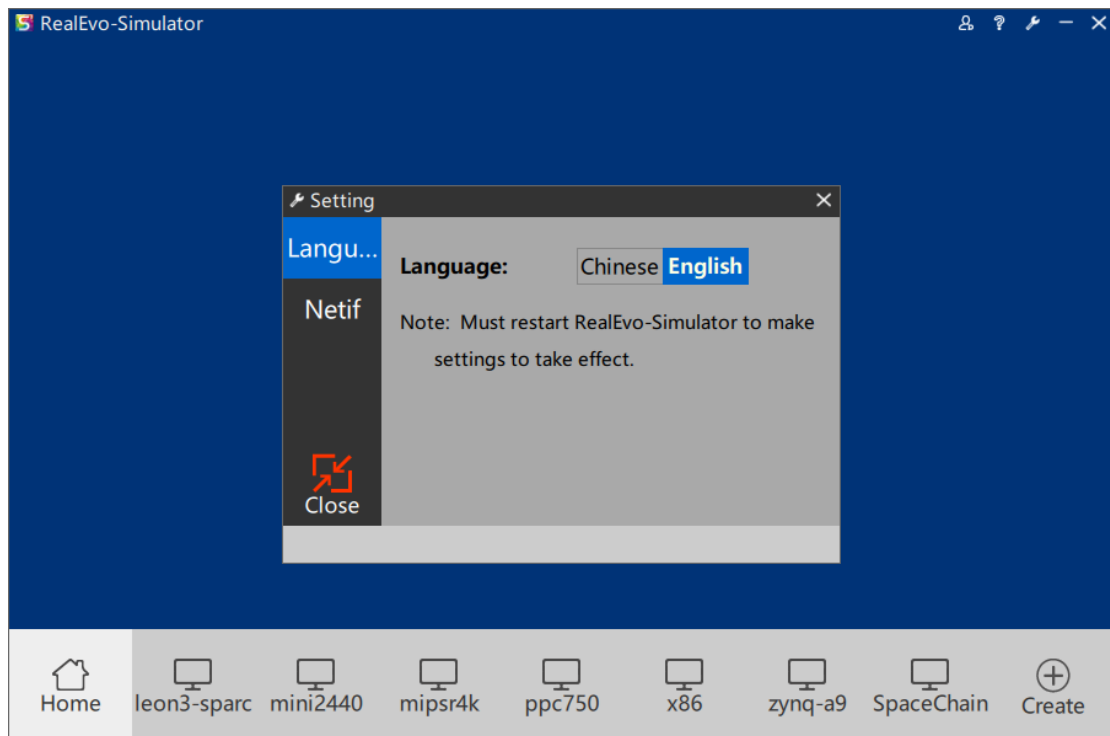
点击“语言”，即可设置 Simulator 的语言，选择英文或中文，目前安装后默认为中文。



选择“英文”后，对应的按钮将会变为蓝色的选中状态。



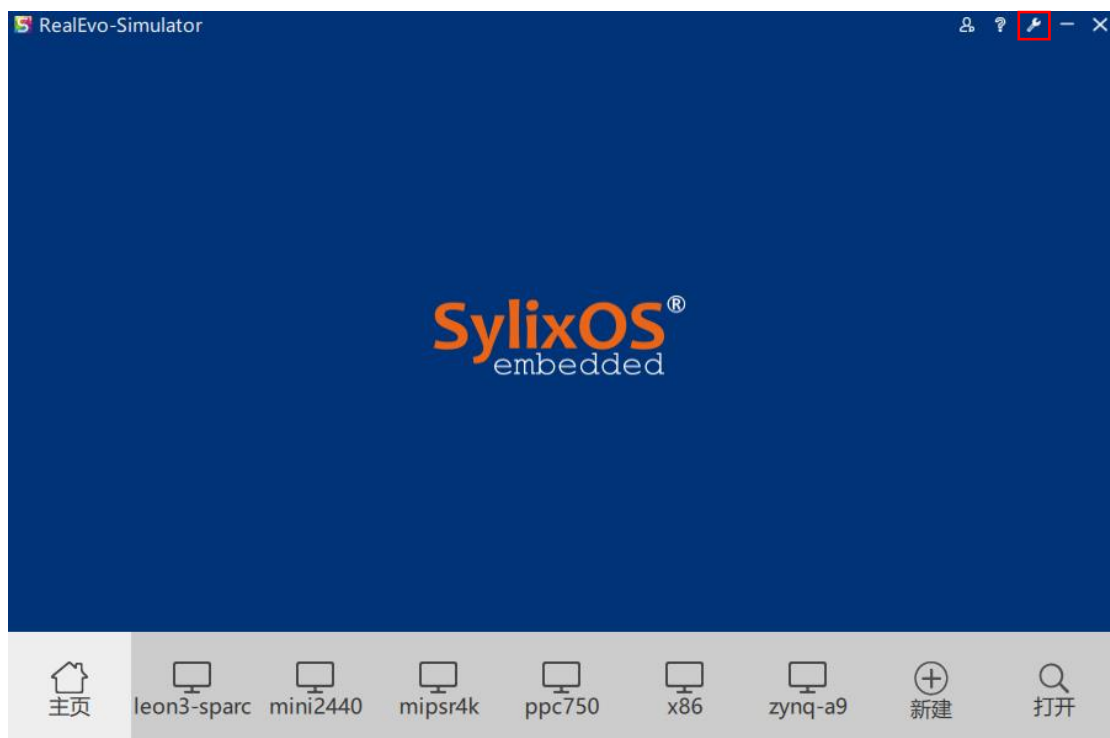
设置成功后，关掉并重新打开 RealEvo-Simulator 即可生效。

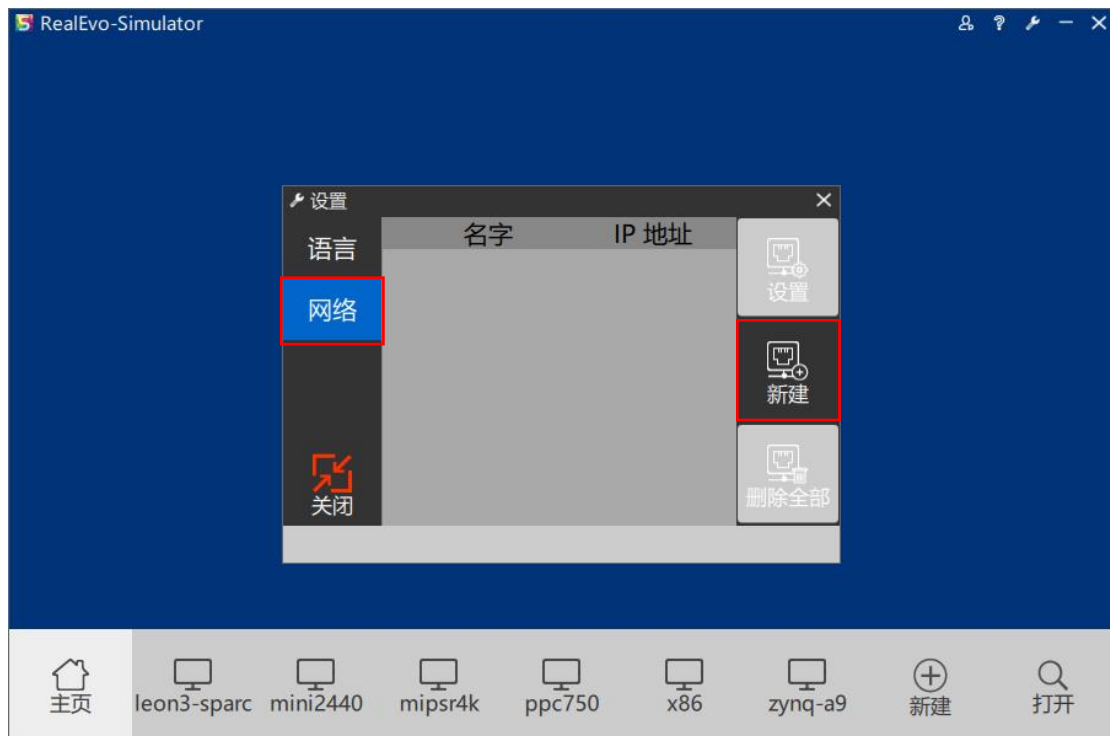


## 1.6 设置虚拟网络适配器

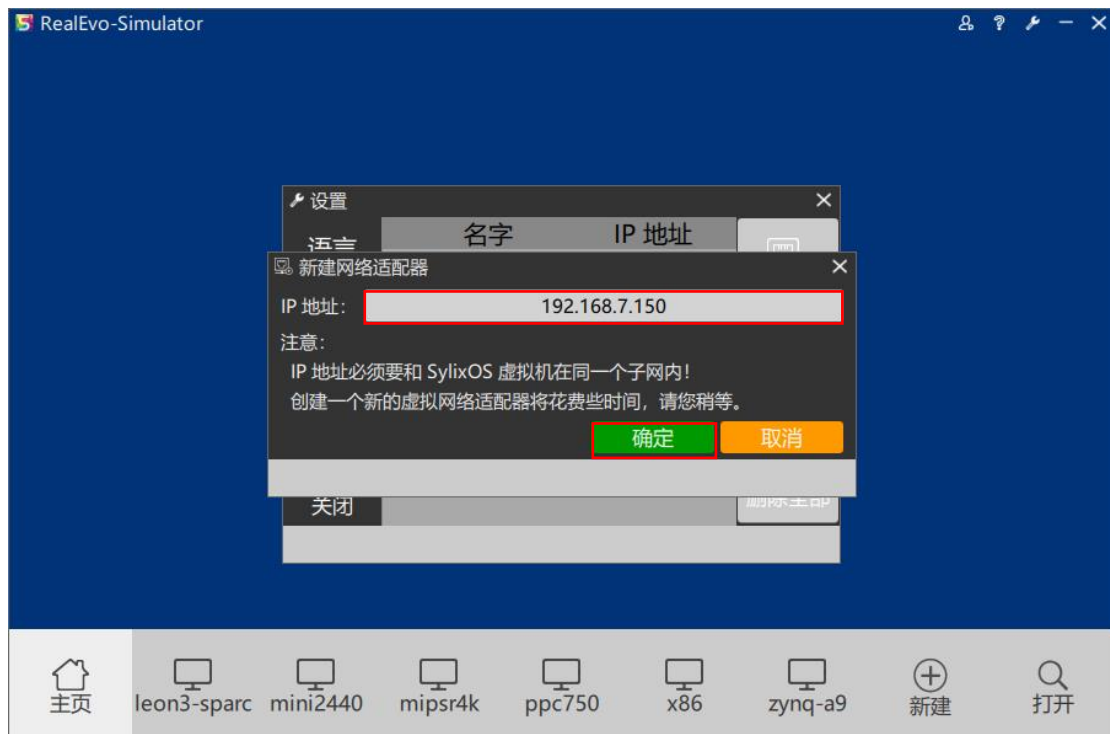
点击右上角的“设置”按钮进入环境设置界面

点击“网络”选项，可以看到已经建立的虚拟网络适配器列表，如果是第一次使用，则列表为空。



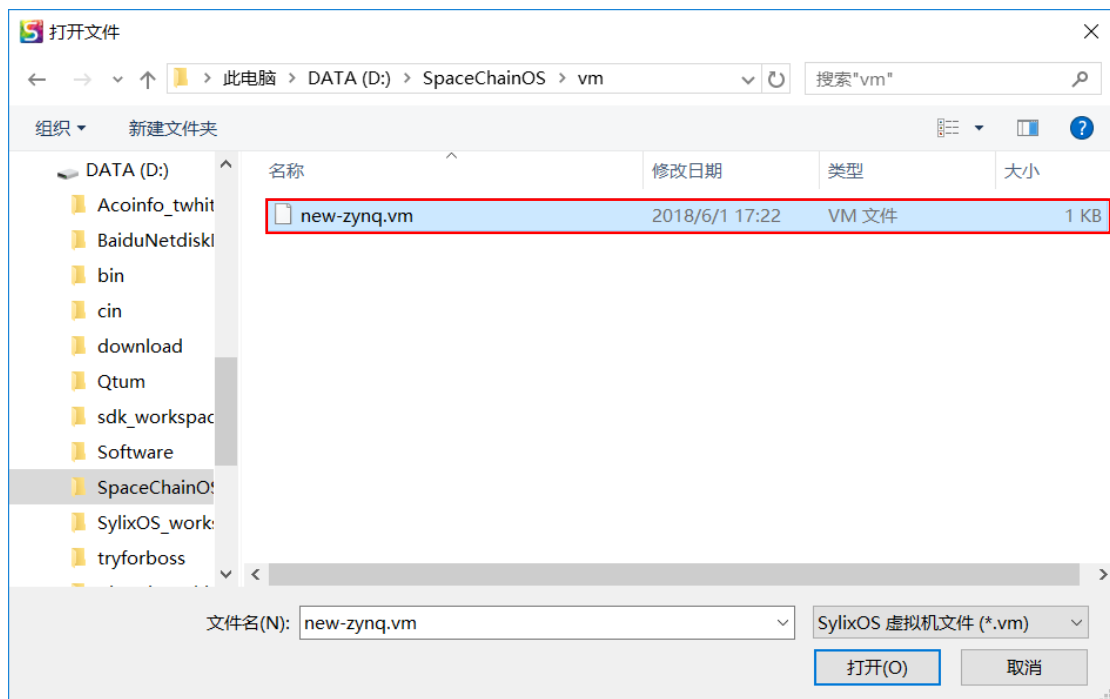


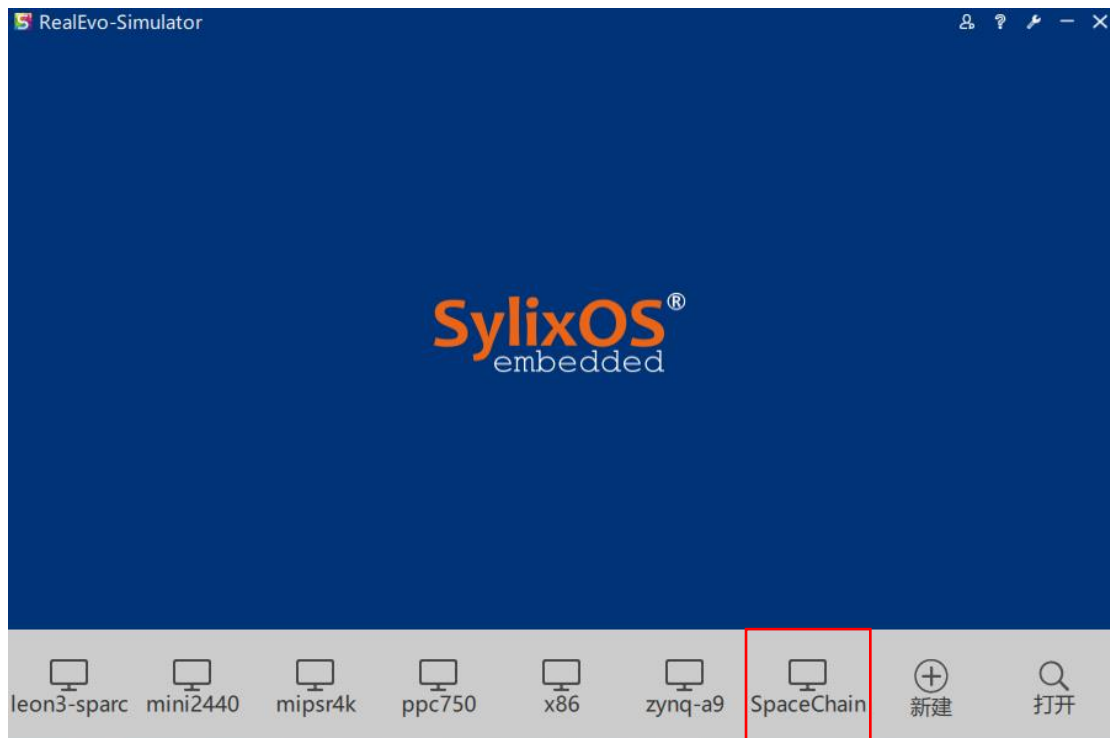
点击“新建”，建立新的虚拟网络适配器，在“IP 地址”栏中输入虚拟网络适配器的地址。  
注：虚拟网络适配器的地址必须和虚拟机在同一网段中，且不能与虚拟机的 ip 地址重复，即虚拟网络适配器的 IP 地址为“192.168.7.xxx”（xxx 不能为 34），因为虚拟机初始默认的 IP 地址为“192.168.7.34”，在此，虚拟网络适配器的地址配置为“192.168.7.150”。



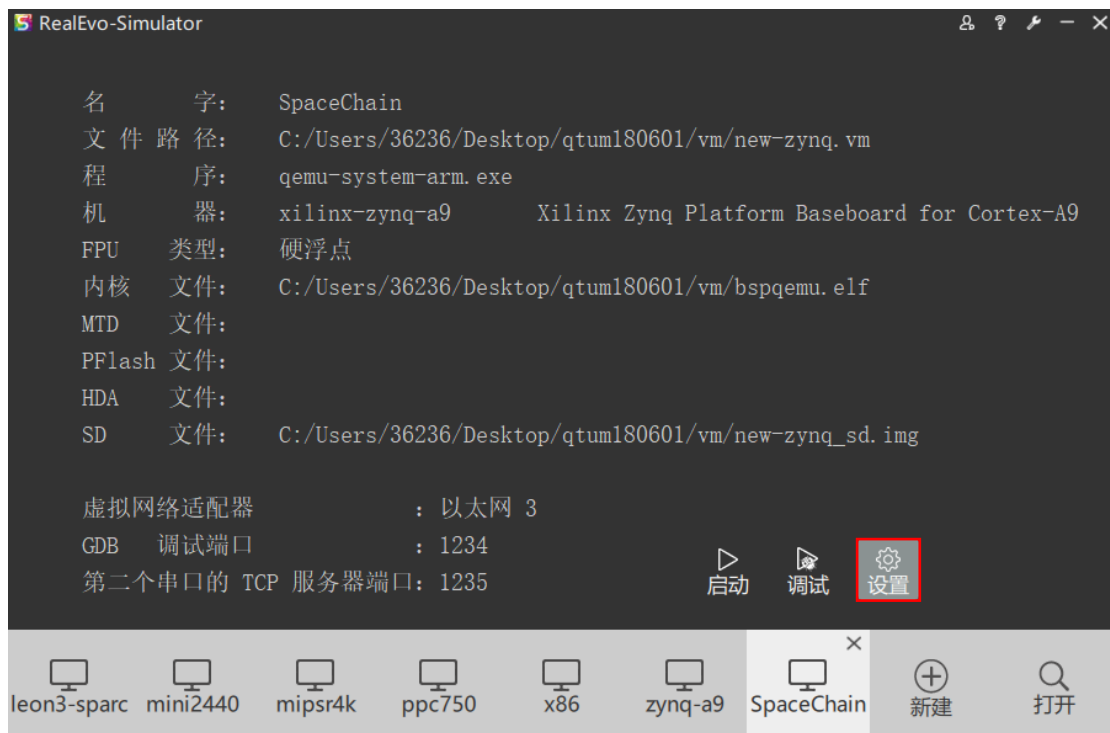
## 1.7 导入虚拟机

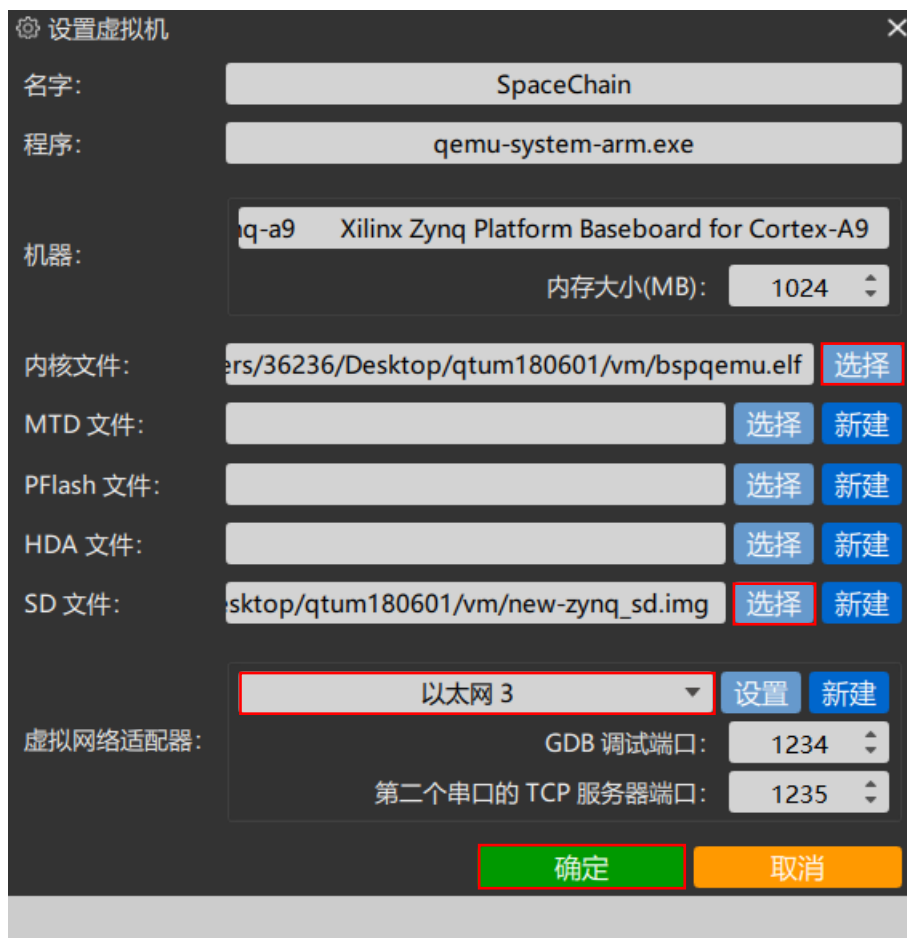
点击右下角“打开”按钮，在“D:\SpaceChainOS\vm”目录下选择 **new-zynq.vm** 文件，即可在下面的虚拟机列表中看到新添加的虚拟机。



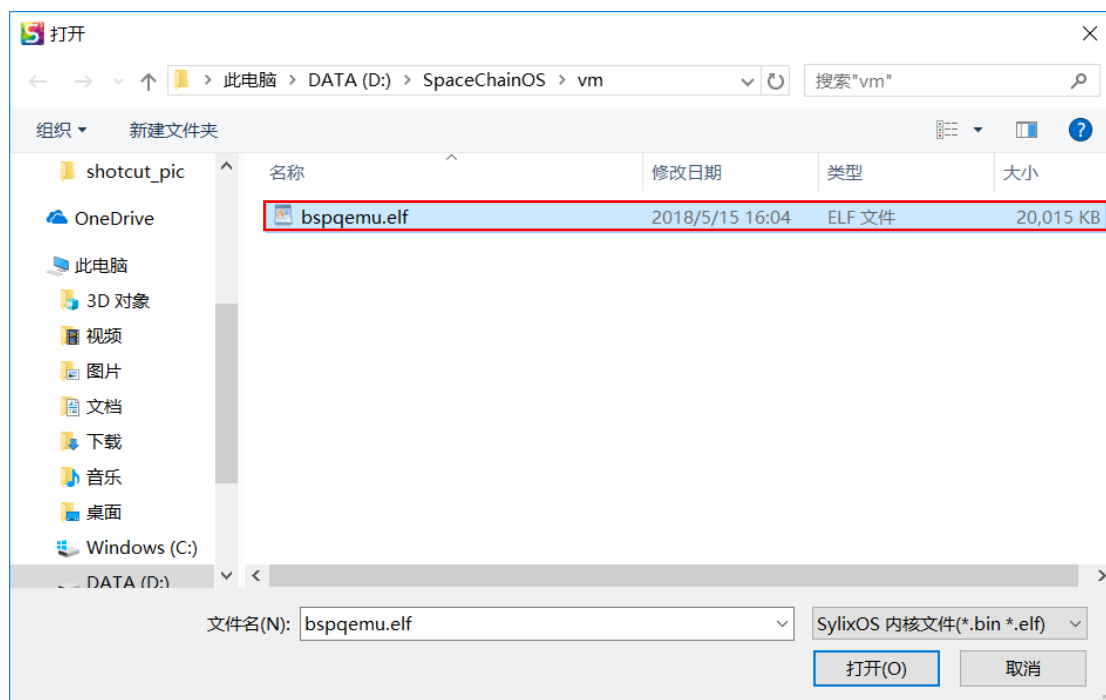


点击新添加的虚拟机，可以看到虚拟机的配置情况。我们需要重新配置虚拟机，点击“设置”重新配置虚拟机。

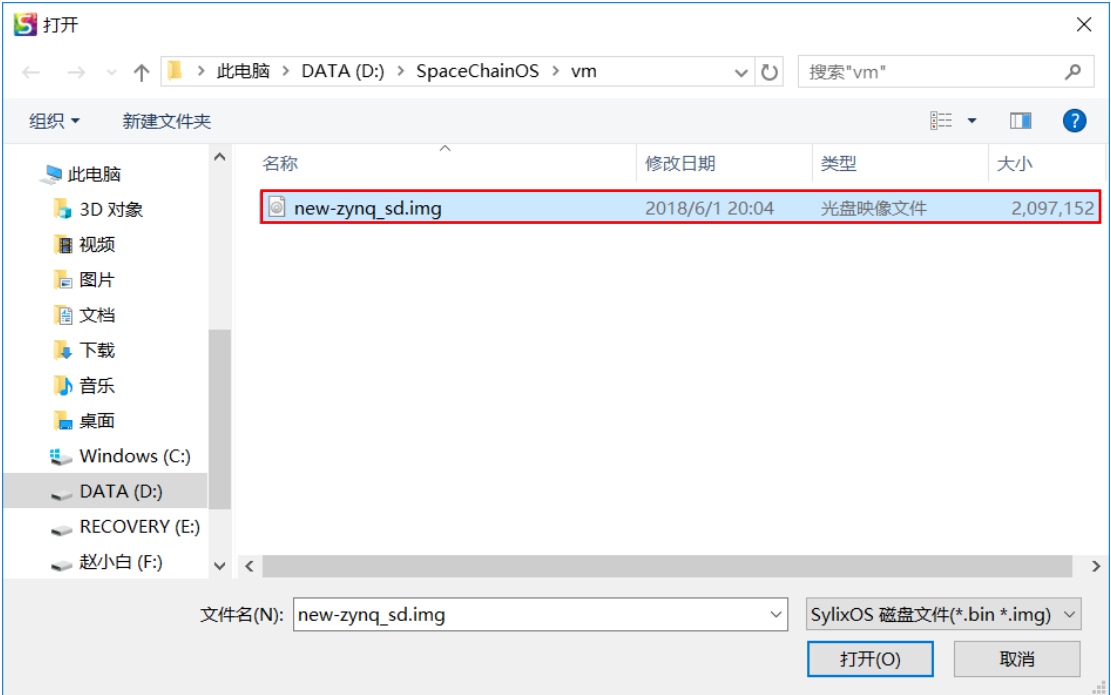




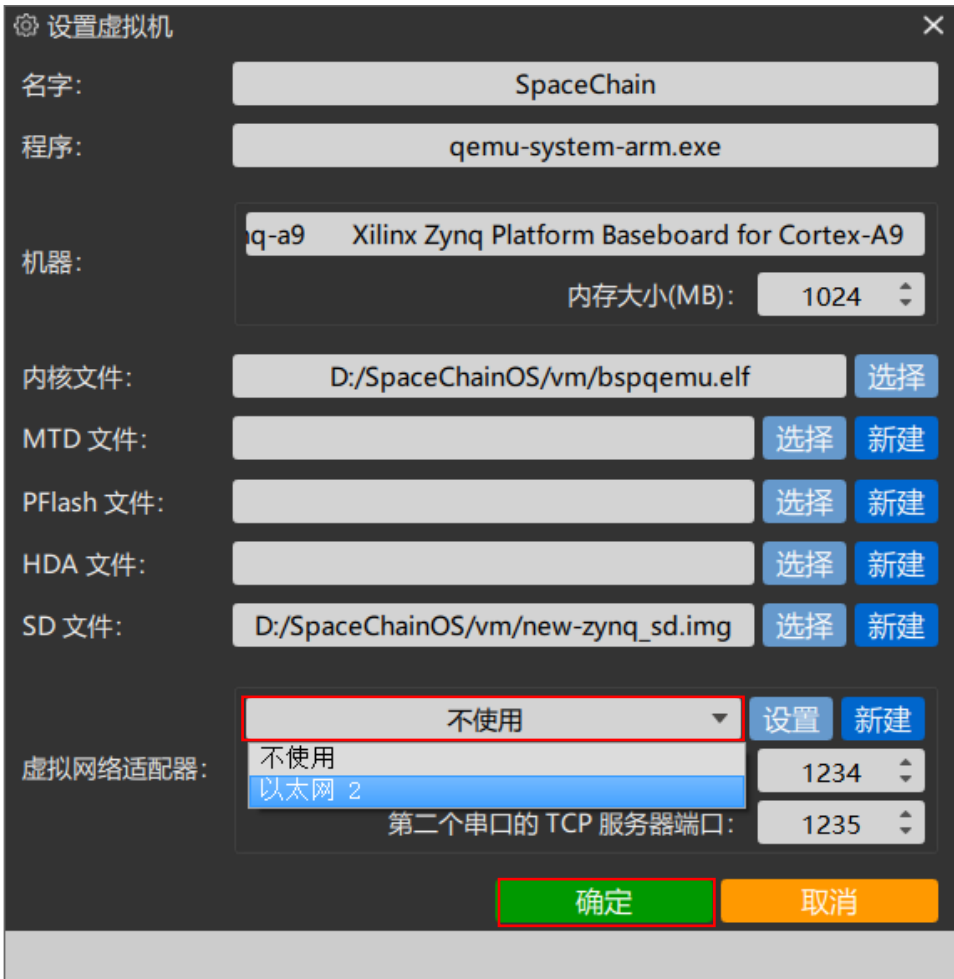
点击“内核文件”一栏右侧的“选择”按钮，在“D:\SpaceChainOS\vm”目录下选择 bspqemu.elf 文件，导入新的内核文件。



点击“SD 文件”一栏中点击“选择”，在“D:\SpaceChainOS\vm”目录下选择 new-zynq\_sd.img 文件，导入虚拟机的 SD 镜像文件。

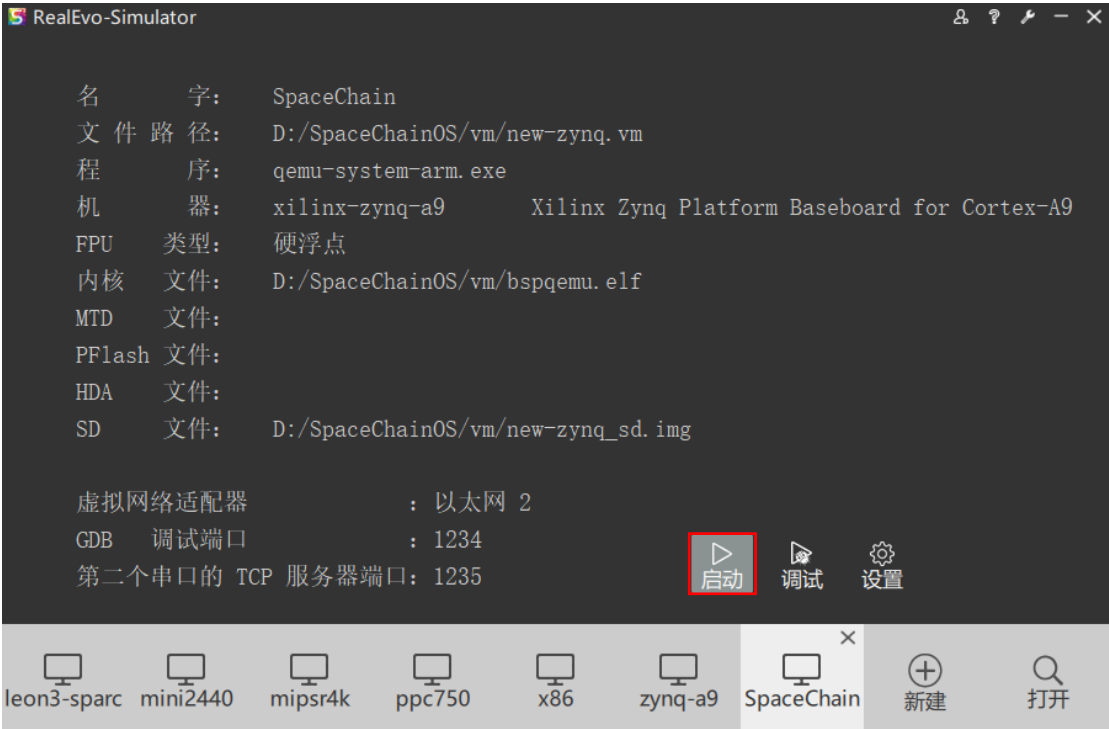


在“虚拟网络适配器”选项的下拉菜单中选择我们之前配置好的虚拟网络适配器。

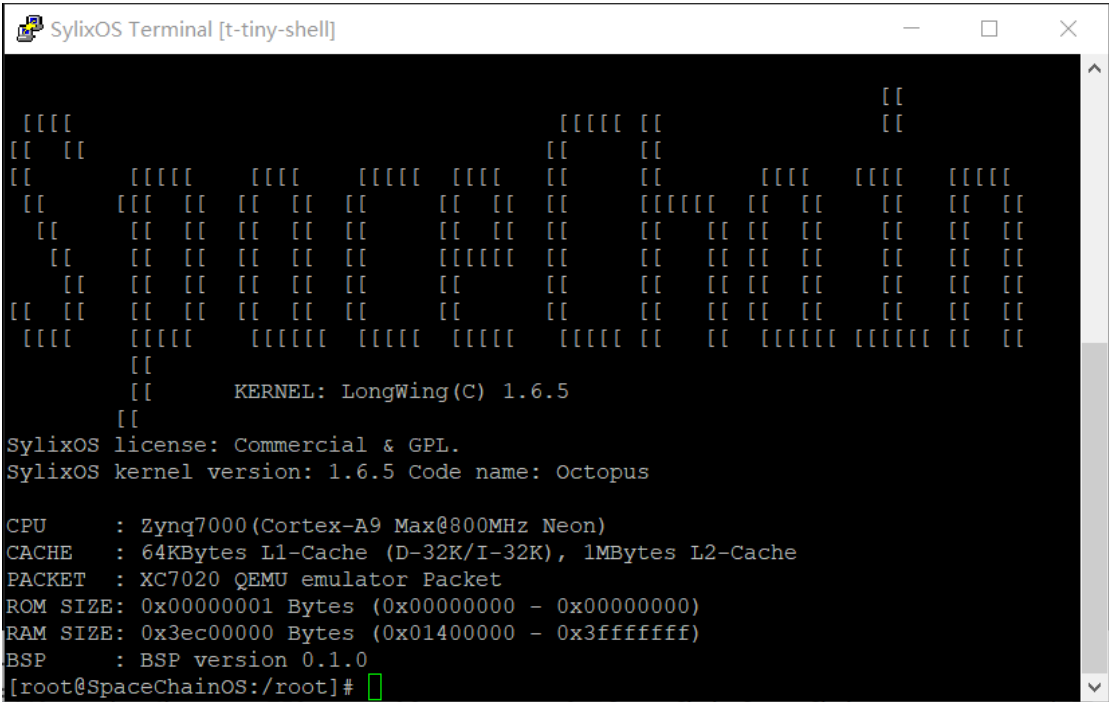




全部配置完成后，点击“确定”完成配置。



点击“启动”，打开虚拟机，进入虚拟机的终端 shell 界面。



## 1.8 修改 SNTP 配置文件

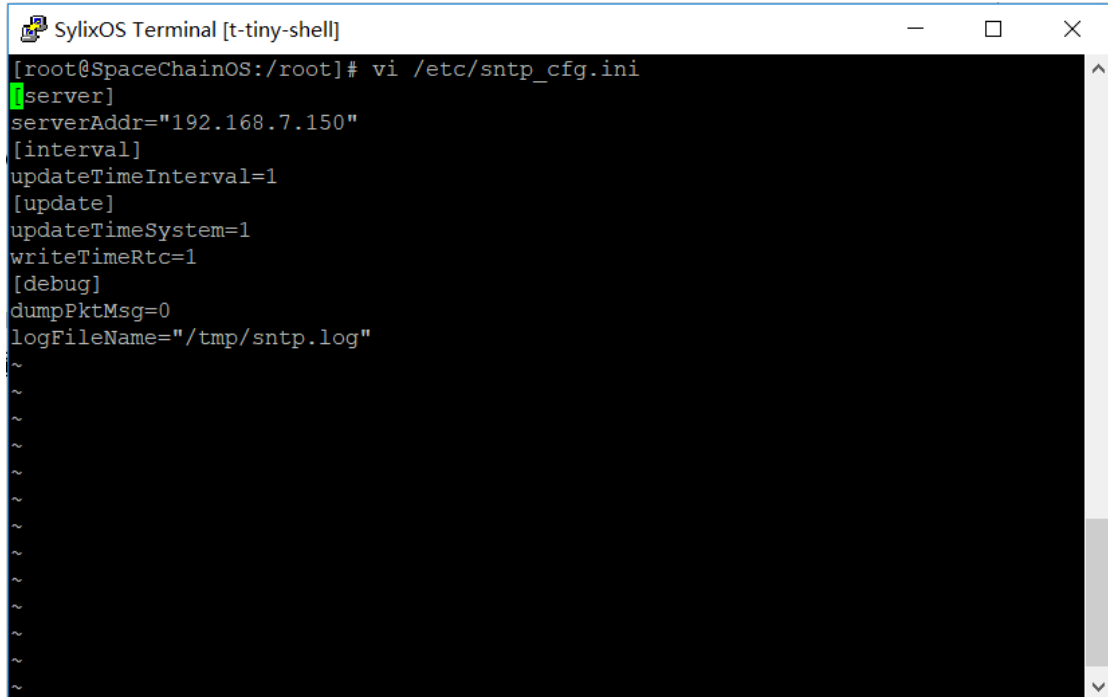
在虚拟机终端 shell 中输入：

**vi /etc/sntp\_cfg.ini**

点击“Enter”键，

再点击“i”键，进入编辑模式，

查看 sntp 服务器的配置情况，按照下图修改配置。



```
SylixOS Terminal [t-tiny-shell]
[root@SpaceChainOS:/root]# vi /etc/sntp_cfg.ini
[server]
serverAddr="192.168.7.150"
[interval]
updateTimeInterval=1
[update]
updateTimeSystem=1
writeTimeRtc=1
[debug]
dumpPktMsg=0
logFileName="/tmp/sntp.log"
~
~
~
~
~
~
~
~
~
~
```

修改完成后点击“Esc”键，然后使用“**shift+;**”（相当于输入“:”）组合键，再输入“**wq**”，点击“Enter”保存并退出编辑器。





修改完成后点击“Esc”键，然后使用“**shift+;**”（相当于输入“:”）组合键，再输入“**wq**”，点击“**Enter**”保存并退出编辑器。



```
SylixOS Terminal [t-tiny-shell]
hostname SpaceChainOS
shstack 500000
insmod /lib/modules/xsiipc.ko
insmod /lib/modules/xinput.ko

ifconfig enl inet 192.168.7.34
ifconfig enl gateway 192.168.7.1
ifconfig enl netmask 255.255.255.0

/apps/sntp/sntp

#ifconfig enl inet 10.4.0.151
#ifconfig enl gateway 10.4.0.1
#ifconfig enl netmask 255.255.255.0

#/apps/sntp/sntp

~
~
~
:wq
```

## 1.10 重启虚拟机

完成以上配置后，在虚拟机终端的 shell 中输入“**reboot**”后点击“**Enter**”重启虚拟机。

```
SylixOS Terminal [t-tiny-shell]
[root@SpaceChainOS:/root]# reboot
[reboot]Reboot...
kernel rebooting...
you can reboot now.
Block device /dev/blk/sdcard-0 part 0 mount to /media/sdcard0 use tpsfs file system.
envionment variables load from /etc/profile fail, error: No such file or directory

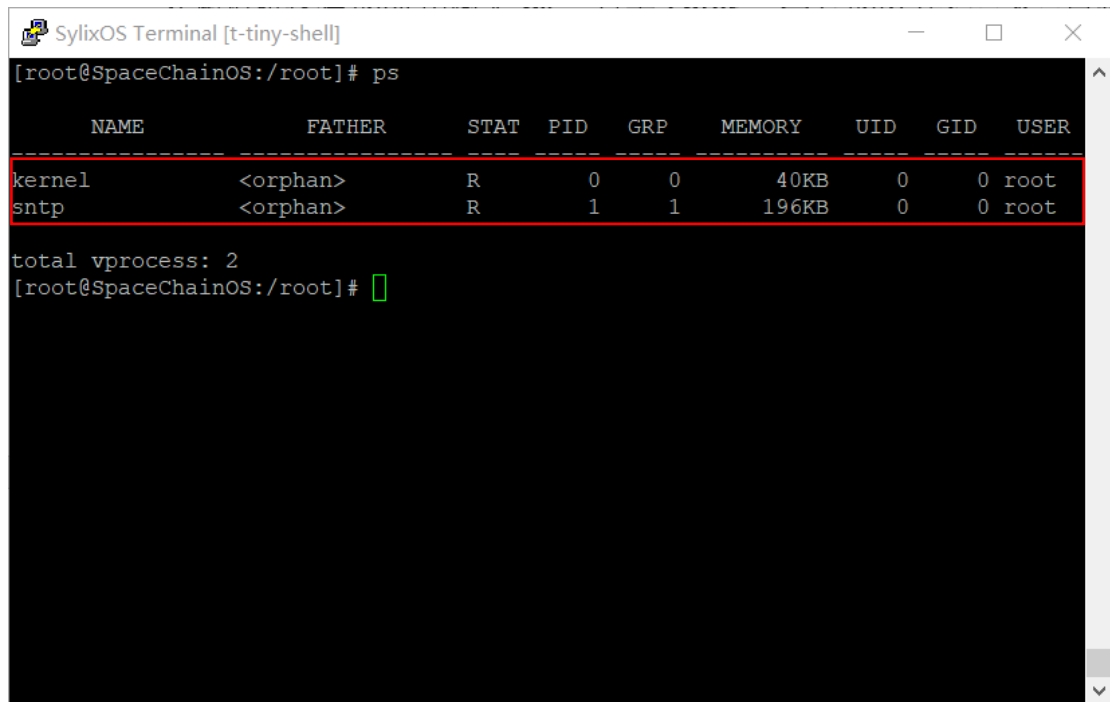
mount sd memory card successfully.
[ifparam]No network parameter for [zynq0] from /etc/ifparam.ini, default parameters will be used.
Press <n> to NOT execute /etc/startup.sh (timeout: 1 sec(s))
ENET 0: full duplex connected 1000M bps
default shell stack: 500000
module /lib/modules/xsiipc.ko register ok, handle: 0x35c3990
module /lib/modules/xinput.ko register ok, handle: 0x35c3ae0
sysname : sylixos
nodename : SpaceChainOS
release : Octopus
version : 1.6.5
machine : Zynq7000(Cortex-A9 Max@800MHz Neon)
```

虚拟机启动后，在 shell 中输入“lsmod”，点击“Enter”，查看 xsiipc.ko 和 xinput.ko 两个内核模块是否已经加载。

```
SylixOS Terminal [t-tiny-shell]
[root@SpaceChainOS:/root]# lsmod
-----
NAME                HANDLE  TYPE  GLB  BASE      SIZE  SYMCNT
-----
VPROC: kernel      pid:    0 TOTAL MEM: 40960
+ xsiipc.ko        035c3990 KERNEL YES 1001a000    4980    14
+ xinput.ko        035c3ae0 KERNEL YES 1000f000    188c     1
total modules: 2
[root@SpaceChainOS:/root]#
```

注意：若两个模块没有成功加载请返回 1.9 节重新检查修改启动脚本，修改后一定要输入“wq”命令保存退出。

在虚拟机终端 shell 中输入“ps”，点击“Enter”，查看 sntp 线程是不是已经启动。



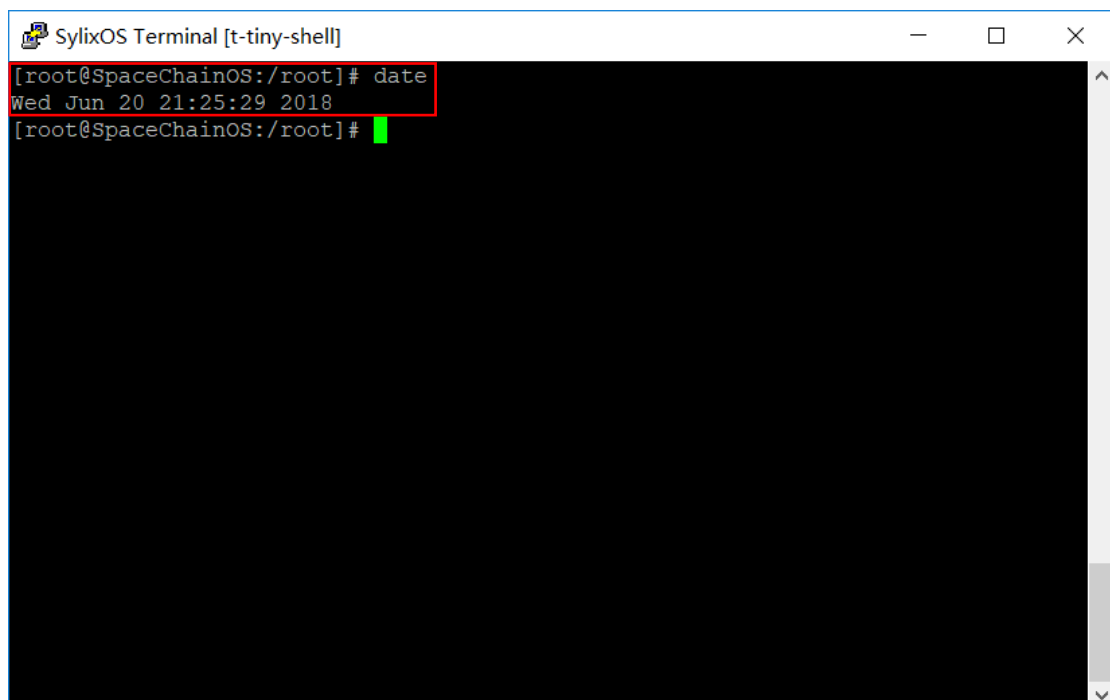
```
SylixOS Terminal [t-tiny-shell]
[root@SpaceChainOS:/root]# ps
  NAME      FATHER    STAT  PID  GRP  MEMORY  UID  GID  USER
-----
kernel      <orphan>   R      0    0    40KB    0    0  root
sntp        <orphan>   R      1    1   196KB    0    0  root

total vprocess: 2
[root@SpaceChainOS:/root]#
```

注意：若两个模块没有成功加载请返回 1.9 节重新检查修改启动脚本，修改后一定要输入“wq”命令保存退出。

在虚拟机的终端 shell 中输入：

“date”，然后点击“Enter”，系统将返回当前的系统时间。



```
SylixOS Terminal [t-tiny-shell]
[root@SpaceChainOS:/root]# date
Wed Jun 20 21:25:29 2018
[root@SpaceChainOS:/root]#
```

注意：若与当前时间不符，则返回 1.3 节查看是否完全关闭所有的杀毒软件和防火墙，请务必关闭所有的杀毒软件和防火墙。

## 2. QTUM 钱包功能演示

### 2.1 启动 Win10 下 QTUM-QT 客户端

可以通过以下方式下载 Windows 环境下 QTUM QT 客户端：

1.OneDrive

<https://1drv.ms/u/s!AmoDU7XUKJnSgqdbUdPHuxIA7U-E-g>

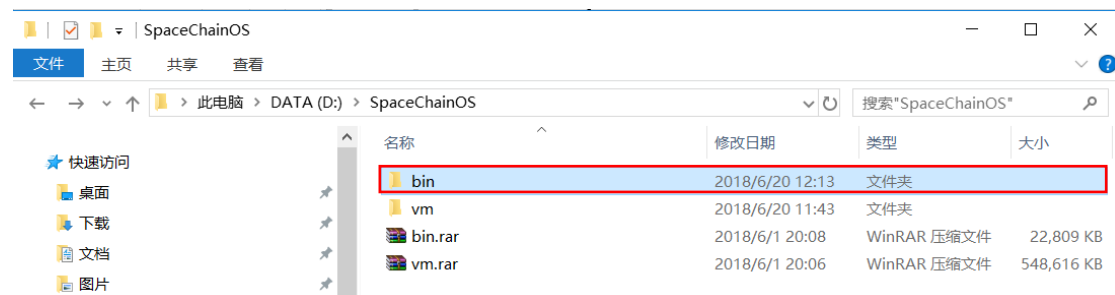
2.Dropbox

<https://www.dropbox.com/s/3nnuchzc7o9cjfz/QT%20Client.rar?dl=0>

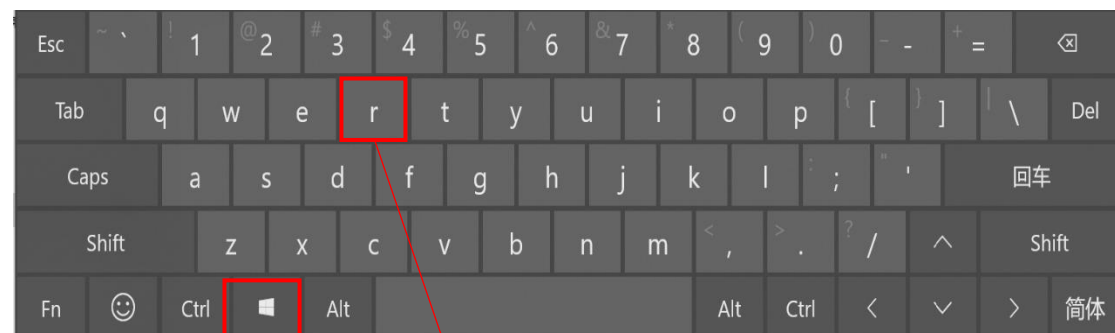
3.Baidu netdisk 百度网盘：

<https://pan.baidu.com/s/1McmU5YoAp27s8Un33mX18Q>

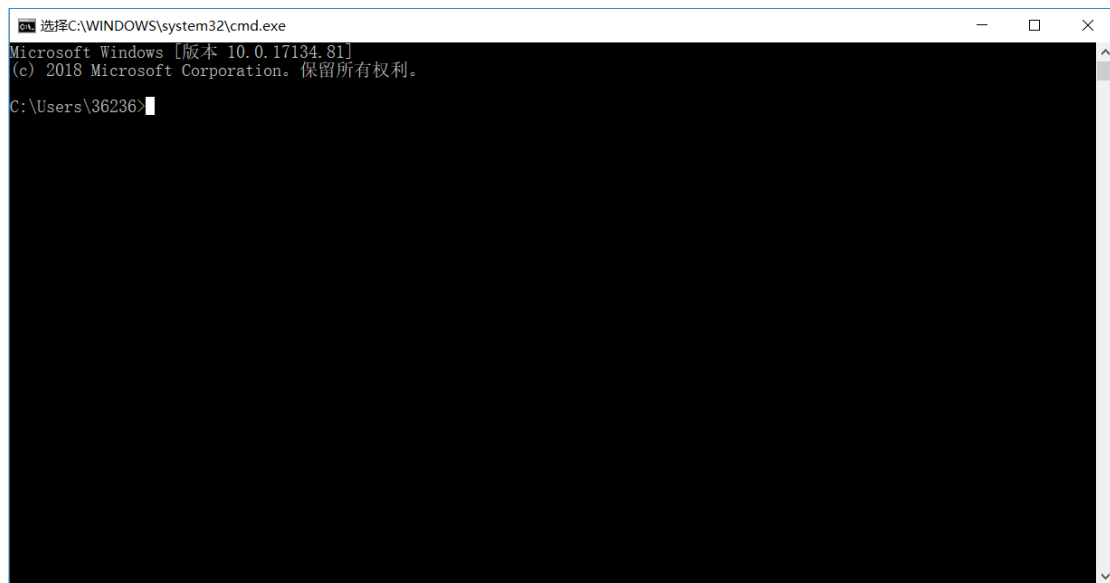
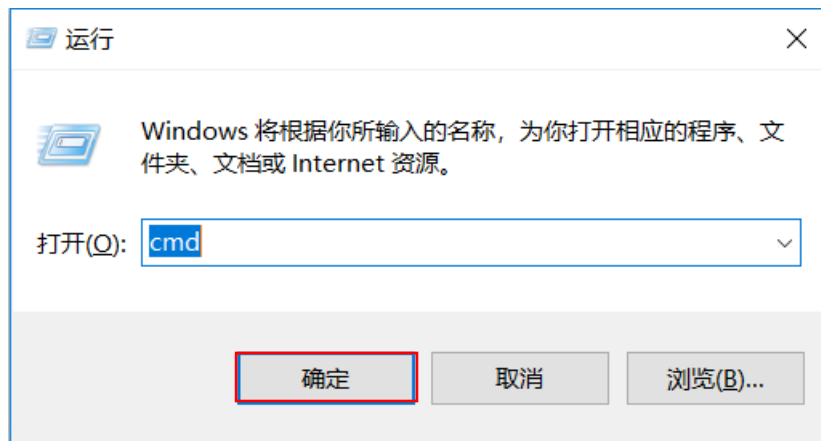
将 QT 客户端的压缩文件下载到 D 盘下，下载完成后将其解压到 D 盘下 SpaceChainOS 文件夹中。



点击“Win+R”组合键，



在“打开”一栏中输入“CMD”，点击“确定”进入命令提示符。

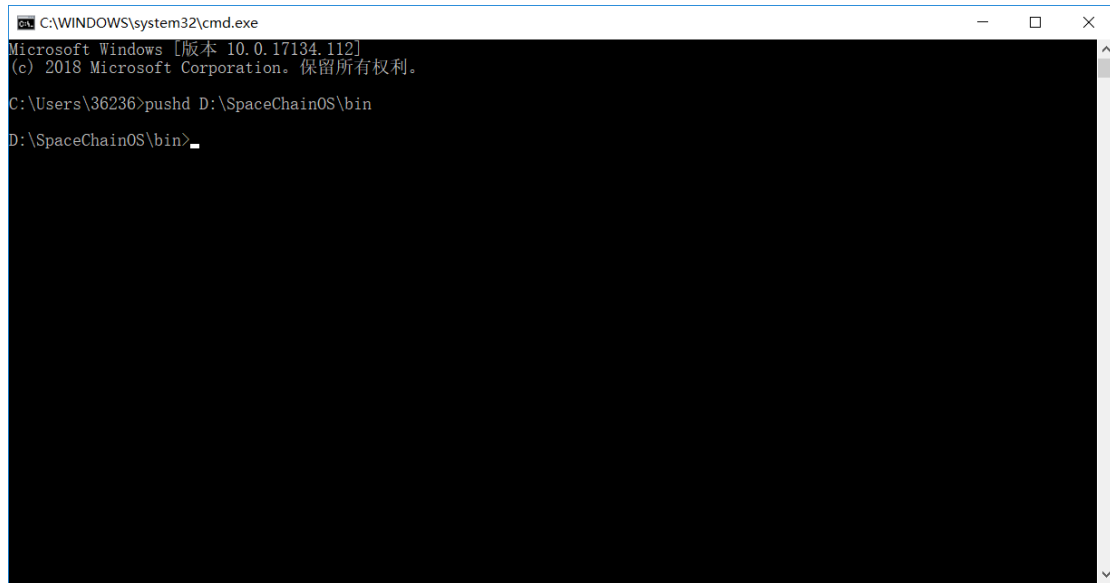


在命令提示符中输入:

**"pushd D:\SpaceChainOS\bin"**

点击**"Enter"**, 使命令提示符进入 **D:\SpaceChainOS\bin** 目录。





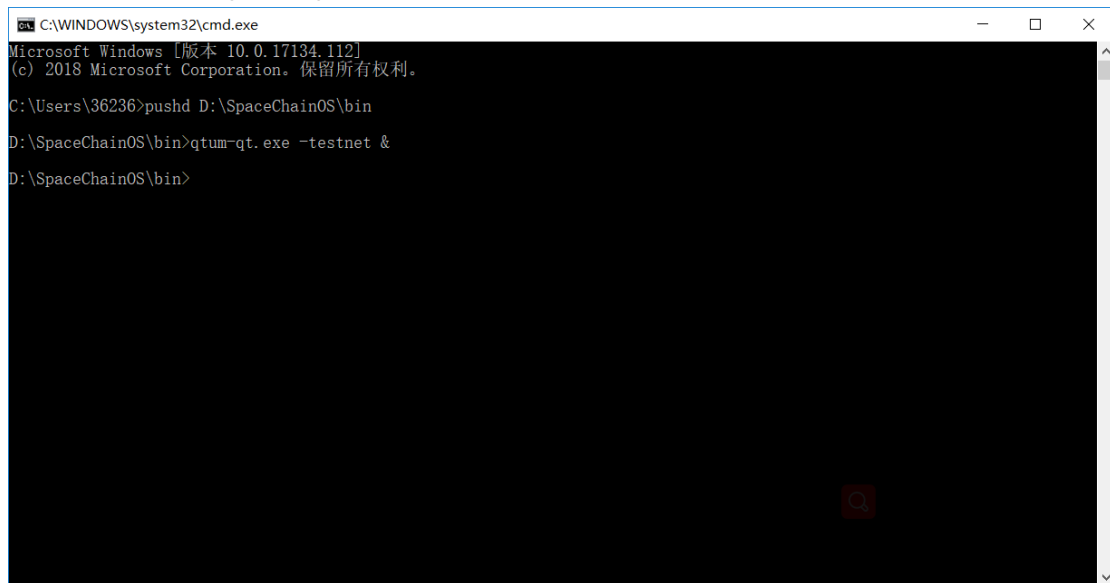
```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.17134.112]
(c) 2018 Microsoft Corporation。保留所有权利。
C:\Users\36236>pushd D:\SpaceChainOS\bin
D:\SpaceChainOS\bin>
```

输入：

**“qtum-qt.exe -testnet &”**

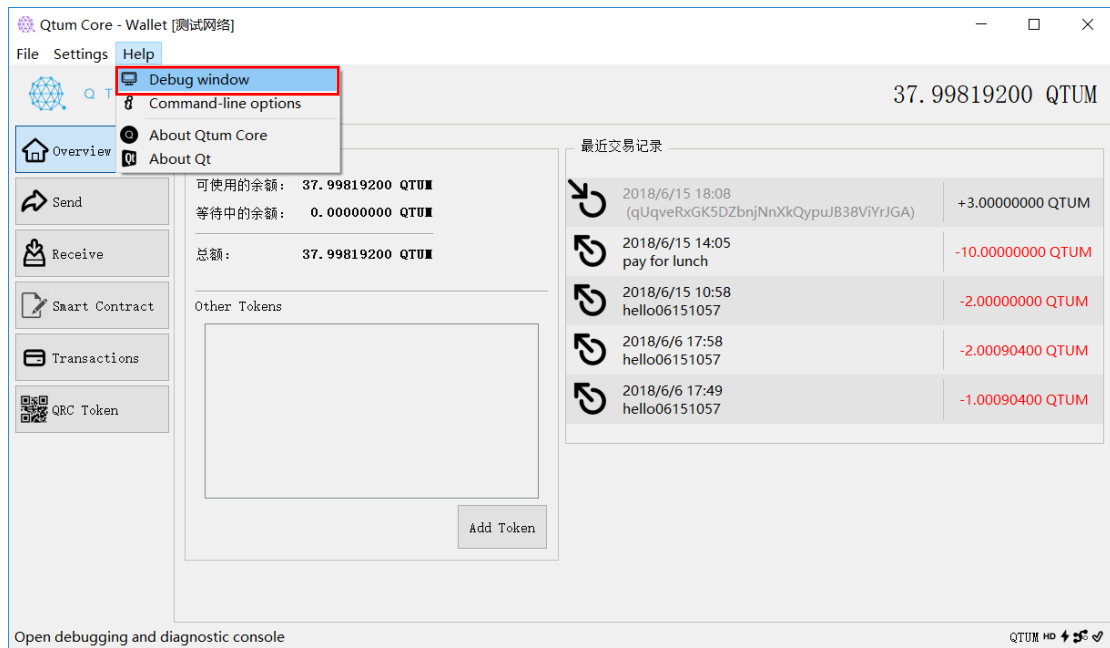
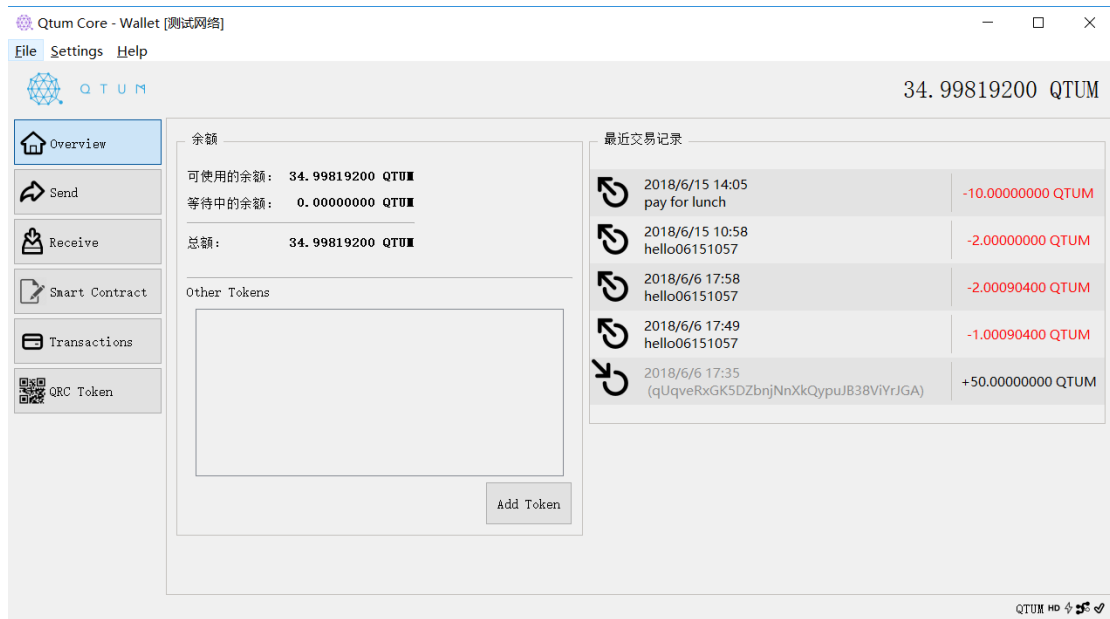
点击“Enter”

启动 Win 环境下 QTUM QT 客户端



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.17134.112]
(c) 2018 Microsoft Corporation。保留所有权利。
C:\Users\36236>pushd D:\SpaceChainOS\bin
D:\SpaceChainOS\bin>qtum-qt.exe -testnet &
D:\SpaceChainOS\bin>
```

QT 客户端启动后自动同步测试网络上的区块，根据网络状况和上次同步的时间，同步时长会不同，**大约需要 20 分钟**。同步完成后，客户端进入主界面。



点击左上角“help”选项下“Debug Window”进入客户端的调试窗口。

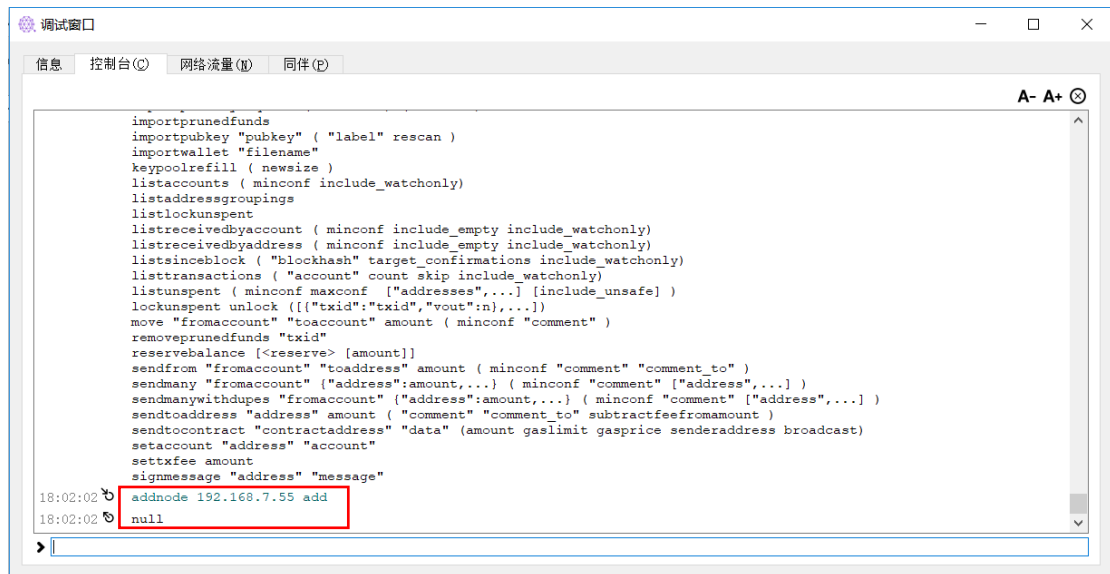
在控制台中输入：

“addnode 192.168.7.34 add”

点击“Enter”

将虚拟机节点加入到 QT 端的伙伴节点列表中。

**注意：**请务必在 PC 端节点上添加虚拟机节点为伙伴节点，当虚拟机节点在超时范围内找不到可用节点后，虚拟机同步线程将自动退出导致同步失败。



## 2.2 获取 QTUM 测试网络 Token

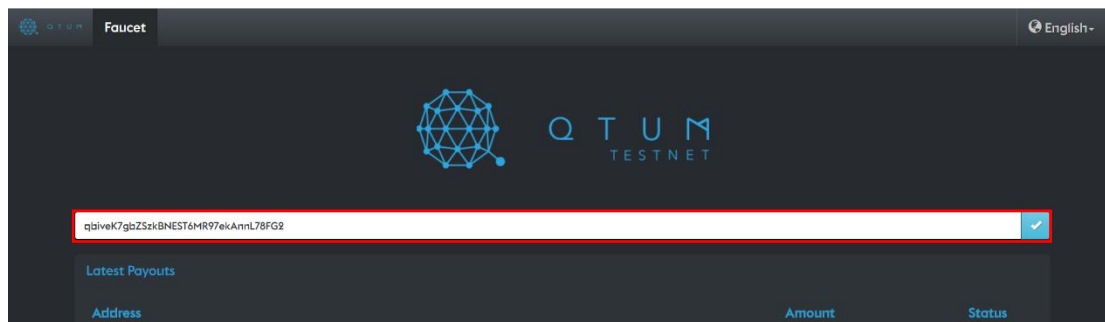
点击 QT 客户端主页面上的“Receive”标签页，点击“请求付款”，可以看到当前钱包的地址。



复制“地址”，点击下面的链接：

<http://testnet-faucet.qtum.info/#/>

进入 QTUM 测试网络 Token 的获取页，将“地址”粘贴到地址栏中，点击确定按钮，QTUM 将自动向该地址转账随机数额的测试 Token 以供测试使用。



## 2.3 启动虚拟机钱包软件

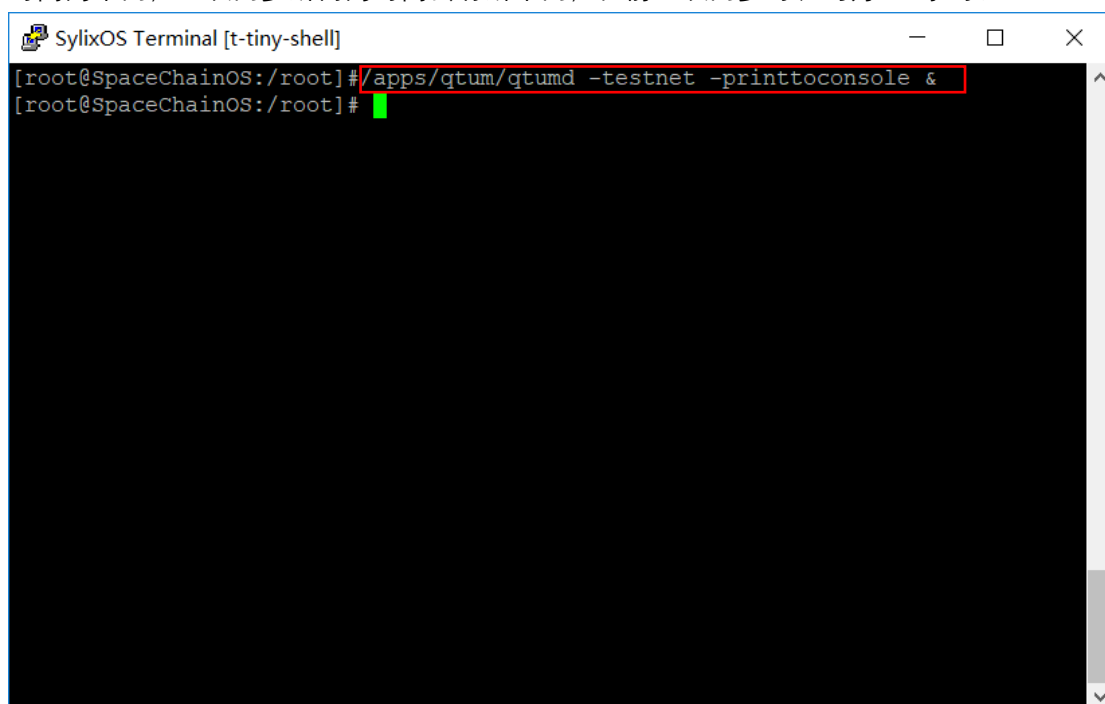
在虚拟机终端的 shell 中输入：

`“/apps/qtum/qtumd -testnet -printtoconsole &”`

**注意：**命令前方不要有空格。

点击“Enter”

虚拟机将开始启动钱包软件线程，并开始初始化环境和区块同步。因为网络状况和上次同步时间的不同，区块同步所需的时间会有所不同，目前区块同步时长约为 20 小时。



```
SylixOS Terminal [t-tiny-shell]
2000-01-01 00:00:29 HTTP: starting 4 worker threads
2000-01-01 00:00:29 Using BerkeleyDB version Berkeley DB 4.8.30: (April 9, 2010)
2000-01-01 00:00:29 Using wallet wallet.dat
2000-01-01 00:00:29 init message: Verifying wallet...
2000-01-01 00:00:29 CDBEnv::Open: LogDir=/root/.qtum/testnet3/database ErrorFile=/root/.qtum/testnet3/db.log
2000-01-01 00:00:31 Bound to [::]:13888
2000-01-01 00:00:31 Bound to 0.0.0.0:13888
2000-01-01 00:00:31 Cache configuration:
2000-01-01 00:00:31 * Using 2.0MiB for block index database
2000-01-01 00:00:31 * Using 8.0MiB for chain state database
2000-01-01 00:00:31 * Using 440.0MiB for in-memory UTXO set (plus up to 286.1MiB of unused mempool space)
2000-01-01 00:00:31 init message: Loading block index...
2000-01-01 00:00:31 Opening LevelDB in /root/.qtum/testnet3/blocks/index
2000-01-01 00:00:34 Opened LevelDB successfully
2000-01-01 00:00:34 Using obfuscation key for /root/.qtum/testnet3/blocks/index: 0000000000000000
2000-01-01 00:00:34 Opening LevelDB in /root/.qtum/testnet3/chainstate
2000-01-01 00:00:35 Opened LevelDB successfully
2000-01-01 00:00:35 Using obfuscation key for /root/.qtum/testnet3/chainstate: 91ed69802418da4c
```

### 如何判断是否同步完成？

可以通过两种方式判断区块是否同步完成：

1. 可以通过终端上打印的区块信息的“**progress**”字段来判断是否同步完成，当“**progress**”字段大于 **1** 时，表示区块数据已经同步到了最新状态。
2. 可以通过终端上打印的区块信息的“**date**”字段来判断是否同步完成，“**date**”字段表示当前同步的区块的生成日期，该日期为**格林威治时间**。若“**date**”字段的时间到达当前时间时，表示区块数据已经同步到了最新状态。

```
SylixOS Terminal [t-tiny-shell]
2018-09-09 09:24:15 UpdateTip: new best=85e08f29a95ba75f498f97a24f75c80b625ec0e8663e857773e height=210217 version=0x20000000 log2_work=69.432193 tx=472829 date=9:24:00' progress=1.000000 cache=8.9MiB(6682txo)
2018-09-09 09:24:15 ProcessNetBlock: ACCEPTED
2018-09-09 09:24:47 UpdateTip: new best=d142c1d859eef6c004c63b398425966186c490724824ad106b2 height=210218 version=0x20000000 log2_work=69.432203 tx=472831 date=9:24:32' progress=1.000000 cache=8.9MiB(6695txo)
2018-09-09 09:24:47 ProcessNetBlock: ACCEPTED
2018-09-09 09:29:09 UpdateTip: new best=dc36e925ac8b8e9a58c930c7f1885475d175f2379290b10911f height=210219 version=0x20000000 log2_work=69.432216 tx=472833 date=9:29:20' progress=1.000000 cache=8.9MiB(6708txo)
2018-09-09 09:29:09 ProcessNetBlock: ACCEPTED
2018-09-09 09:32:32 UpdateTip: new best=d54d06b0b3e4c650d1c0d8bca47c814030189b743e7f3c058ba height=210220 version=0x20000000 log2_work=69.432225 tx=472835 date=9:32:32' progress=1.000000 cache=8.9MiB(6721txo)
2018-09-09 09:32:32 ProcessNetBlock: ACCEPTED
2018-09-09 09:33:07 UpdateTip: new best=e4fc3e15d7fd6613551a2c6425362c2293c3c77f594e58725b1 height=210221 version=0x20000000 log2_work=69.432234 tx=472837 date=9:33:04' progress=1.000000 cache=8.9MiB(6734txo)
2018-09-09 09:33:07 ProcessNetBlock: ACCEPTED
2018-09-09 09:35:38 UpdateTip: new best=a9550f816bfa8c4c6c146aef7512cc991ab12344288c2cc8090 height=210222 version=0x20000000 log2_work=69.432245 tx=472839 date=9:35:44' progress=1.000000 cache=8.9MiB(6747txo)
2018-09-09 09:35:38 ProcessNetBlock: ACCEPTED
```

注意：

1. 当钱包的区块数据已经同步到最新状态时，新的区块依旧会产生，钱包还会收到更新的区块，所以依旧会有新的区块信息打印在终端上，可以直接敲“enter”键，在终端上执行接下来的操作。
2. 区块信息中的“date”字段表示该区块的生成日期，默认选择格林威治时间，因此，如果钱包软件在北京（即东八区）同步区块，则当当前时间减去 8 个小时等于终端打印的区块信息的“date”字段表示的时间时（即当“date” = 当前时间 - 8 hours 时），表示钱包软件的区块数据已经同步到了最新状态。

## 2.4 余额查询功能

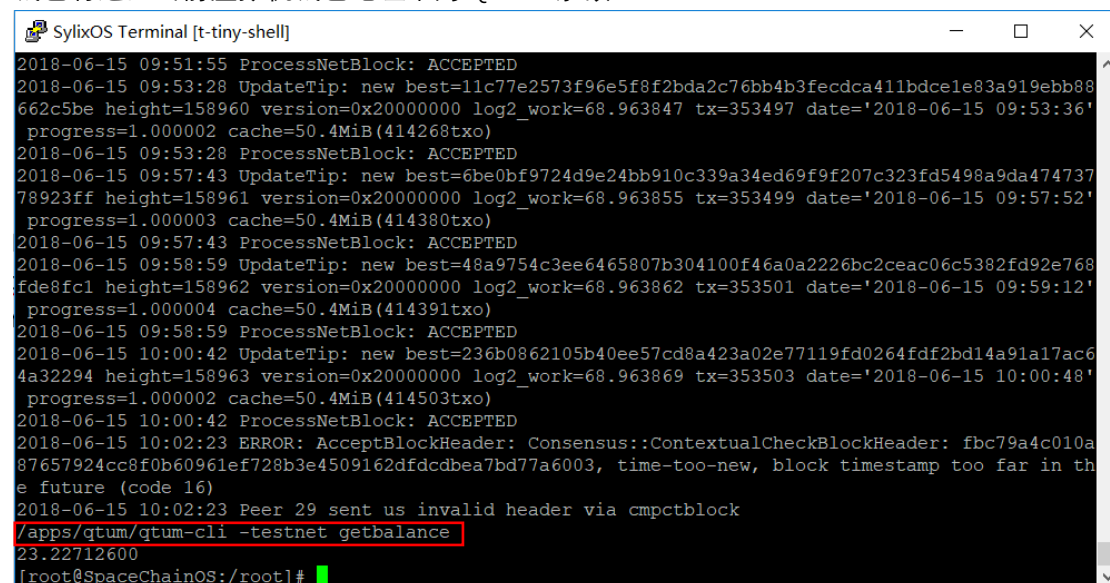
可以通过 QTUM 钱包的命令进行节点和网络信息查询。

在虚拟机终端 shell 中输入：

`“/apps/qtum/qtum-cli -testnet getbalance”`

点击“Enter”

钱包将返回当前虚拟机钱包地址中的 QTUM 余额



```
SylixOS Terminal [t-tiny-shell]
2018-06-15 09:51:55 ProcessNetBlock: ACCEPTED
2018-06-15 09:53:28 UpdateTip: new best=11c77e2573f96e5f8f2bda2c76bb4b3fecdc411bdce1e83a919ebb88
662c5be height=158960 version=0x20000000 log2_work=68.963847 tx=353497 date='2018-06-15 09:53:36'
progress=1.000002 cache=50.4MiB(414268txo)
2018-06-15 09:53:28 ProcessNetBlock: ACCEPTED
2018-06-15 09:57:43 UpdateTip: new best=6be0bf9724d9e24bb910c339a34ed69f9f207c323fd5498a9da474737
78923ff height=158961 version=0x20000000 log2_work=68.963855 tx=353499 date='2018-06-15 09:57:52'
progress=1.000003 cache=50.4MiB(414380txo)
2018-06-15 09:57:43 ProcessNetBlock: ACCEPTED
2018-06-15 09:58:59 UpdateTip: new best=48a9754c3ee6465807b304100f46a0a2226bc2ceac06c5382fd92e768
fde8fc1 height=158962 version=0x20000000 log2_work=68.963862 tx=353501 date='2018-06-15 09:59:12'
progress=1.000004 cache=50.4MiB(414391txo)
2018-06-15 09:58:59 ProcessNetBlock: ACCEPTED
2018-06-15 10:00:42 UpdateTip: new best=236b0862105b40ee57cd8a423a02e77119fd0264fdf2bd14a91a17ac6
4a32294 height=158963 version=0x20000000 log2_work=68.963869 tx=353503 date='2018-06-15 10:00:48'
progress=1.000002 cache=50.4MiB(414503txo)
2018-06-15 10:00:42 ProcessNetBlock: ACCEPTED
2018-06-15 10:02:23 ERROR: AcceptBlockHeader: Consensus::ContextualCheckBlockHeader: fbc79a4c010a
87657924cc8f0b60961ef728b3e4509162dfdcdbea7bd77a6003, time-too-new, block timestamp too far in th
e future (code 16)
2018-06-15 10:02:23 Peer 29 sent us invalid header via cmpctblock
/apps/qtum/qtum-cli -testnet getbalance
23.22712600
[root@SpaceChainOS:/root]#
```

## 2.5 转账功能

转账前需要先获取自己的钱包地址和接收方的钱包地址。

在虚拟机终端 shell 中输入：

`“/apps/qtum/qtum-cli -testnet getnewaddress”`

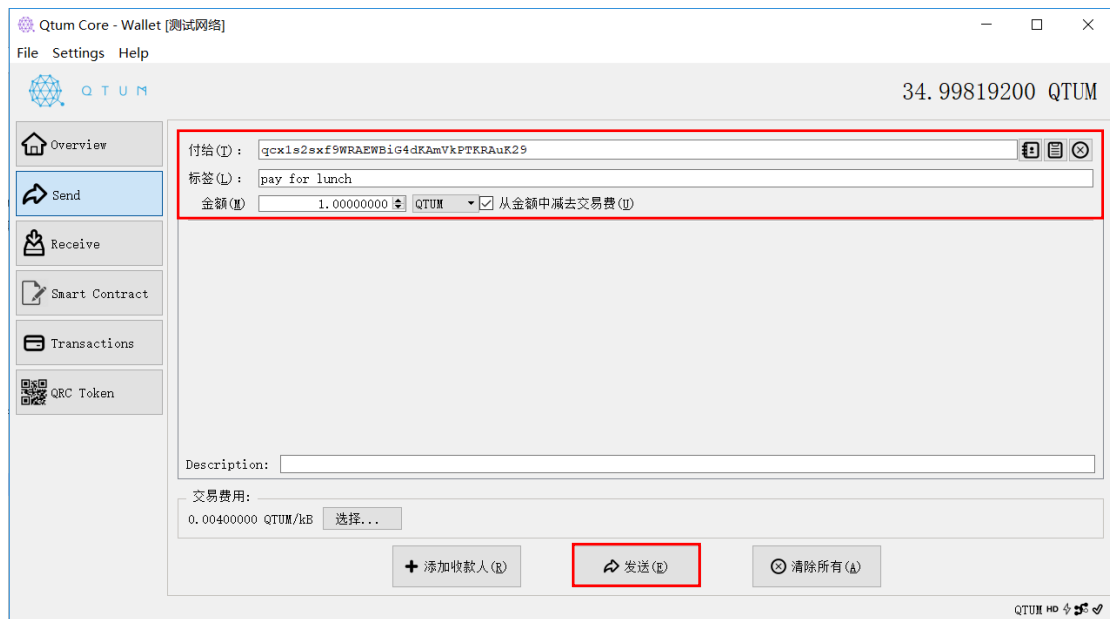
点击“Enter”

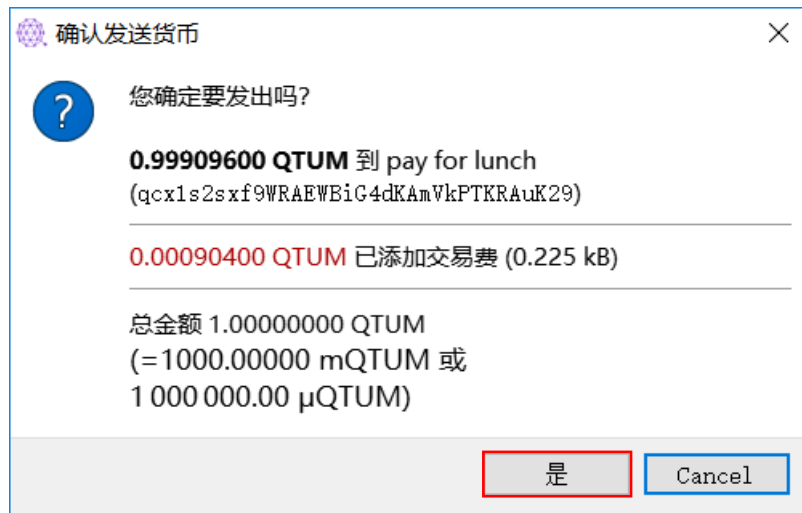
获取虚拟机的钱包地址。

```
SylixOS Terminal [t-tiny-shell]
4a32294 height=158963 version=0x20000000 log2_work=68.963869 tx=353503 date='2018-06-15 10:00:48'
progress=1.000002 cache=50.4MiB(414503txo)
2018-06-15 10:00:42 ProcessNetBlock: ACCEPTED
2018-06-15 10:02:23 ERROR: AcceptBlockHeader: Consensus::ContextualCheckBlockHeader: fbc79a4c010a
87657924cc8f0b60961ef728b3e4509162dfdcdbea7bd77a6003, time-too-new, block timestamp too far in th
e future (code 16)
2018-06-15 10:02:23 Peer 29 sent us invalid header via cmpctblock
/apps/qtum/qtum-cli -testnet getbalance
23.22712600
[root@SpaceChainOS:/root]# 2018-06-15 10:04:19 UpdateTip: new best=fbc79a4c010a87657924cc8f0b6096
1ef728b3e4509162dfdcdbea7bd77a6003 height=158964 version=0x20000000 log2_work=68.963876 tx=353506
date='2018-06-15 10:02:40' progress=0.999972 cache=50.4MiB(414517txo)
2018-06-15 10:04:19 ProcessNetBlock: ACCEPTED
2018-06-15 10:04:19 UpdateTip: new best=e2e8b820933871a672c6dbbc8c0ed4a905de1d44143b79aa7a8dff4e3
db89366 height=158965 version=0x20000000 log2_work=68.963884 tx=353508 date='2018-06-15 10:04:32'
progress=1.000004 cache=50.4MiB(414528txo)
2018-06-15 10:04:19 ProcessNetBlock: ACCEPTED
[root@SpaceChainOS:/root]# /apps/qtum/qtum-cli -testnet getnewaddress
2018-06-15 10:04:43 keypool added key 118, size=101
2018-06-15 10:04:43 keypool reserve 18
2018-06-15 10:04:43 keypool keep 18
qJkQYYAiUfvpnQryYBLqF1qPZKTQPA5ugE
[root@SpaceChainOS:/root]#
```

## QTUM QT 客户端向虚拟机节点转账

在 QT 客户端的“Send”标签页中，在“付给”一栏中输入虚拟机端的钱包地址，在“金额”一栏输入要转账的金额。点击“发送”按钮，在确认对话框中点击“是”确认转账。





等待交易被确认后，虚拟机节点将收到 QT 客户端节点的转账。

### 虚拟机节点向 QTUM QT 客户端转账

点击 QT 客户端主页面上的“Receive”标签页，点击“请求付款”，可以看到当前 QT 端钱包的“地址”。



在虚拟机终端的 shell 中输入：

“/apps/qtum/qtum-cli -testnet sendtoaddress QT 端的钱包地址 转账数额”

点击“Enter”

完成转账。



```
SylixOS Terminal [t-tiny-shell]
b97ac64 height=158967 version=0x20000000 log2_work=68.963901 tx=353512 date='2018-06-15 10:07:28' ^
progress=1.000000 cache=50.4MiB(414552txo)
2018-06-15 10:07:29 ProcessNetBlock: ACCEPTED

[root@SpaceChainOS:/root]# /apps/qtum/qtum-cli -testnet sendtoaddress qUqveRxGK5DZbnjNnXkQypuJB38
ViYrJGA 3
2018-06-15 10:08:21 keypool added key 119, size=101
2018-06-15 10:08:21 keypool reserve 19
2018-06-15 10:08:21 CommitTransaction:
CTransaction(hash=854fd7c654, ver=2, vin.size=2, vout.size=2, nLockTime=158967)
  CTxIn(COutPoint(47d233a33e, 0), scriptSig=473044022023677413112f48, nSequence=4294967294)
  CTxIn(COutPoint(21c939d1da, 0), scriptSig=47304402201fb74ea08a67ef, nSequence=4294967294)
  CScriptWitness()
  CScriptWitness()
  CTxOut(nValue=3.00000000, scriptPubKey=76a9147bca53435d98bfc11e96871d)
  CTxOut(nValue=0.99669600, scriptPubKey=76a914f7fdb6a4e499da65c772727)
2018-06-15 10:08:21 keypool keep 19
2018-06-15 10:08:21 AddToWallet 854fd7c6543d3f9aeaaa5675212f600edbe1fc971b9bec237a08558bf5447d3
new
2018-06-15 10:08:21 AddToWallet 854fd7c6543d3f9aeaaa5675212f600edbe1fc971b9bec237a08558bf5447d3
2018-06-15 10:08:21 Relaying wtx 854fd7c6543d3f9aeaaa5675212f600edbe1fc971b9bec237a08558bf5447d3
854fd7c6543d3f9aeaaa5675212f600edbe1fc971b9bec237a08558bf5447d3
[root@SpaceChainOS:/root]#
```

## 2.6 智能合约

QTUM 支持与 EVM 兼容的智能合约，因此 QTUM 支持使用 **Solidity** 语言进行智能合约的编写。用 Solidity 语言写好的智能合约，可以直接使用基于浏览器的编译器 **Remix Solidity IDE** 进行编译。下面以 Remix Solidity IDE 的方式，通过一个简单的例子来说明开发、部署智能合约的过程。

### 2.6.1 编写和编译智能合约

Remix Solidity IDE 链接如下：

<http://remix.ethereum.org>

这里通过一个简单的智能合约实例，说明合约怎样完成编译与部署。

测试的智能合约代码如下：

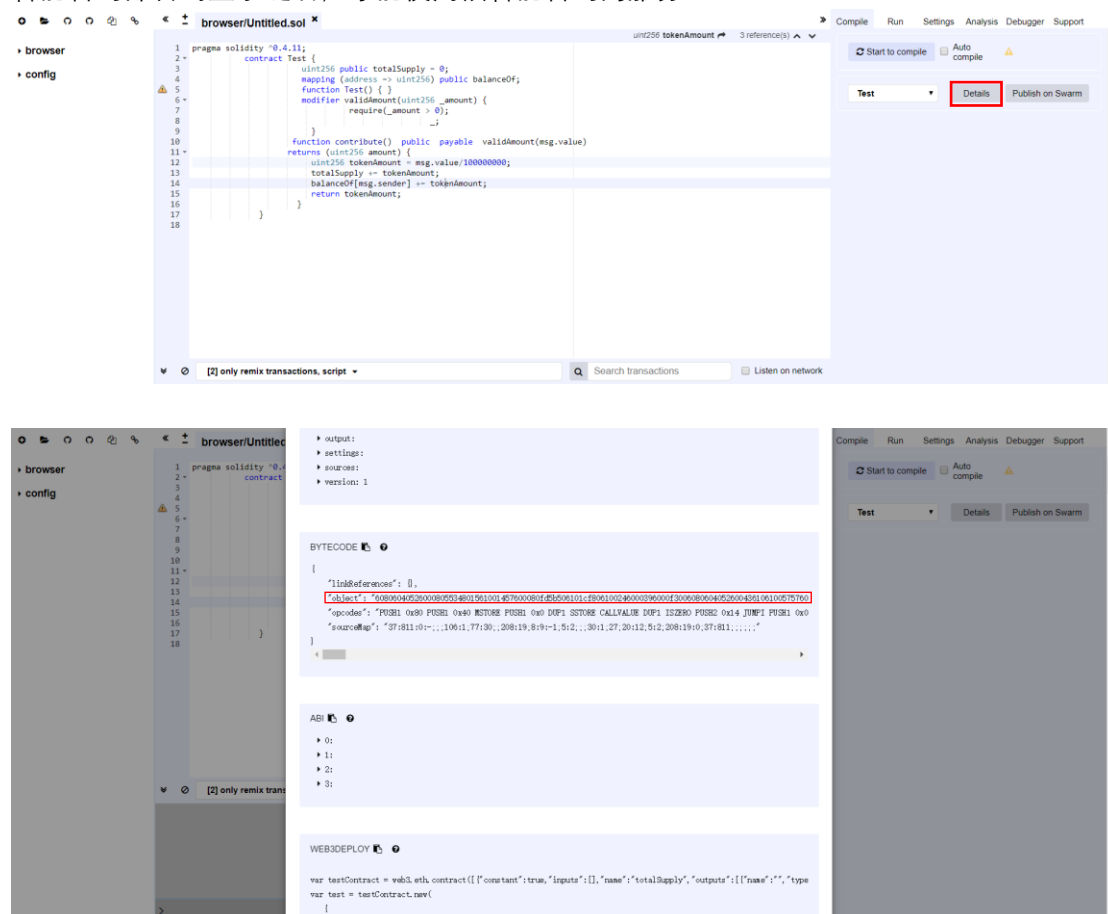
```
pragma solidity ^0.4.11;
contract Test {
    uint256 public totalSupply = 0;
    mapping (address => uint256) public balanceOf;
    function Test() {}
    modifier validAmount(uint256 _amount) {
        require(_amount > 0);
        _;
    }
    function contribute() public payable validAmount(msg.value)
    returns (uint256 amount) {
        uint256 tokenAmount = msg.value/1000000000;
        totalSupply += tokenAmount;
    }
}
```

```

        balanceOf[msg.sender] += tokenAmount;
        return tokenAmount;
    }
}

```

把以上代码拷贝到 Remix Solidity IDE 左边的空白处，点击右边“**Details**”就可以看到编译生成的**字节码(bytecode)**。“Object”一栏便是智能合约编译后的可执行**二进制代码**，下面的部署需要用到生成的二进制码。部署智能合约相当于是安装一个软件到操作系统，因此需要把智能合约部署到量子链后，才能使用该智能合约的服务。



## 2.6.2 部署智能合约

部署智能合约需要访问 QTUM 区块链的节点，通过 QTUM 节点提供的 cli 命令行或 RPC 调用可以进行智能合约相关的操作。

使用“qtum-cli”调用“createcontract”可以把智能合约部署到 QTUM 量子链上。

将编译好的二进制码复制后粘贴到 “createcontract”参数后。

在虚拟机终端的 shell 中输入：

“qtum-cli -testnet createcontract

```

60806040526000805534801561001457600080fd5b506101cf806100246000396000f300
608060405260043610610057576000357c010000000000000000000000000000000000
00000000000000000000000000000000900463ffffffff16806318160ddd1461005c57806370a08231146

```

```
10087578063d7bb99ba146100de575b600080fd5b34801561006857600080fd5b506100
716100fc565b6040518082815260200191505060405180910390f35b348015610093576
00080fd5b506100c8600480360381019080803573ffffffffffffffffffffffffffff16906
0200190929190505050610102565b6040518082815260200191505060405180910390f3
5b6100e661011a565b6040518082815260200191505060405180910390f35b600054815
65b60016020528060005260406000206000915090505481565b60008034600081111515
61012d57600080fd5b6305f5e1003481151561013c57fe5b049150816000808282540192
50508190555081600160003373ffffffffffffffffffffffffffff1673ffffffffffffffffffff
ffffffff16815260200190815260200160002060008282540192505081905550819250
5050905600a165627a7a7230582036fbc985698e681ba280c27357c2504a4adb93309252
642be88f77dc0cd4a6bf0029 3000000 0.0000005"
```

其中二进制部分就是之前在 Remix Solidity IDE 中生成的二进制码。

“3000000 0.0000005”是 **gaslimit** 和 **gasprice** 参数，两个参数可以不设置或可以改为其他的值。

点击“Enter”，执行智能合约在测试网络上的部署命令。

命令执行后或产生如下输出：

```
{
  "txid": "52b3eef0fe0032f685974256972b340655501fc87adec4cffe2b96cc46446001",
  "sender": "qYcH9wHcd86LaDAxquKPyfpmSraxAbBop",
  "hash160": "a5161fee8ff92457c0829889854888b45255f961",
  "address": "2d4b6564a012ae1383854ac48574c2248660d897"
}
```

这个输出是一个交易，所有合约相关的功能都是通过发送交易触发的，该交易在 QTUM 量子链上部署了地址为“2d4b6564a012ae1383854ac48574c2248660d897”的智能合约。智能合约需要等待区块对交易确认后，才在 Qtum 链上部署完成。

## 2.7 终止服务（stop 命令部分）

当用户完成全部操作并希望退出 QTUM 服务时，可以通过调用 qtum-cli 的 stop 命令，将本次同步的区块存储到文件中，并在存储完成后自动退出 QTUM 服务线程。

当用户想要退出 QTUM 服务时，在虚拟机的命令行终端上输入：

```
“/apps/qtum/qtum-cli -testnet stop”
```

点击“enter”执行命令

```
SylixOS Terminal [t-tiny-shell]
2018-09-09 09:33:07 ProcessNetBlock: ACCEPTED
2018-09-09 09:35:38 UpdateTip: new best=a9550f816bfa8c4c6c146aef7512cc991ab12344
288c2cc8090 height=210222 version=0x20000000 log2_work=69.432245 tx=472839 date=
9:35:44' progress=1.000000 cache=8.9MiB(6747txo)
2018-09-09 09:35:38 ProcessNetBlock: ACCEPTED

[root@SpaceChainOS:/root]#
[root@SpaceChainOS:/root]#
[root@SpaceChainOS:/root]#
[root@SpaceChainOS:/root]# /apps/qtum/
qtumd          qtum          qtum-cli
[root@SpaceChainOS:/root]# /apps/qtum/qtum-cli -testnet stop
Qtum server stopping
[tor: Thread interrupt]
[libevent: epoll wait: Wait timed out]
[scheduler thread interrupt]
[addcon thread exit]
[Shutdown: In progress...]
[msghand thread exit]
[net thread exit]
[Dumped mempool: 0.001572s to copy, 0.245327s to dump]
[opencon thread exit]
[torcontrol thread exit]
[... 17:40:18.352| Closing state DB]
[... 17:40:18.486| Closing state DB]
[Shutdown: done]
```

QTUM 服务开始自动存储同步的区块到数据文件中，当转存工作和线程退出工作全部完成后，QTUM 会提示“Shutdown done”，此时表示已经将全部的区块存储到文件中，Qtum 钱包完全关闭。

```
SylixOS Terminal [t-tiny-shell]
2018-09-09 09:33:07 ProcessNetBlock: ACCEPTED
2018-09-09 09:35:38 UpdateTip: new best=a9550f816bfa8c4c6c146aef7512cc991ab12344
288c2cc8090 height=210222 version=0x20000000 log2_work=69.432245 tx=472839 date=
9:35:44' progress=1.000000 cache=8.9MiB(6747txo)
2018-09-09 09:35:38 ProcessNetBlock: ACCEPTED

[root@SpaceChainOS:/root]#
[root@SpaceChainOS:/root]#
[root@SpaceChainOS:/root]#
[root@SpaceChainOS:/root]# /apps/qtum/
qtumd          qtum          qtum-cli
[root@SpaceChainOS:/root]# /apps/qtum/qtum-cli -testnet stop
Qtum server stopping
[tor: Thread interrupt]
[libevent: epoll_wait: Wait timed out]
[scheduler thread interrupt]
[addcon thread exit]
[Shutdown: In progress...]
[msghand thread exit]
[net thread exit]
[Dumped mempool: 0.001572s to copy, 0.245327s to dump]
[opencon thread exit]
[torcontrol thread exit]
[... 17:40:18.352| Closing state DB]
[... 17:40:18.486| Closing state DB]
[Shutdown: done]
```

此时，可以通过 **ps** 命令查看 QTUM 线程退出情况。

在虚拟机的命令行终端上输入：“**ps**”

点击“**enter**”确定执行

可以看到 QTUM 服务的线程全部退出完成，现在运行的线程中已经没有 **Qtumd** 和 **Qtum-cli** 线程。

```
SylixOS Terminal [t-tiny-shell]
qtumd      qtum      qtum-cli
[root@SpaceChainOS:/root]# /apps/qtum/qtum-cli -testnet stop
Qtum server stopping
[2018-09-09 09:40:08] tor: Thread interrupt
[2018-09-09 09:40:08] libevent: epoll_wait: Wait timed out
[2018-09-09 09:40:08] scheduler thread interrupt
[2018-09-09 09:40:08] addcon thread exit
[2018-09-09 09:40:08] Shutdown: In progress...
[2018-09-09 09:40:08] msghand thread exit
[2018-09-09 09:40:08] net thread exit
[2018-09-09 09:40:09] Dumped mempool: 0.001572s to copy, 0.245327s to dump
[2018-09-09 09:40:10] opencon thread exit
[2018-09-09 09:40:12] torcontrol thread exit
[2018-09-09 09:40:18] ... 17:40:18.352| Closing state DB
[2018-09-09 09:40:18] ... 17:40:18.486| Closing state DB
[2018-09-09 09:40:18] Shutdown: done

[root@SpaceChainOS:/root]# ps
```

NAME	FATHER	STAT	PID	GRP	MEMORY	UID	GID	USER
kernel	<orphan>	R	0	0	52KB	0	0	root
sntp	<orphan>	R	1	1	244KB	0	0	root

```
total vprocess: 2
[root@SpaceChainOS:/root]#
```