

第1章 开发工具

1.1 IDE 使用说明

SpaceChain OS 规模庞大，使用领域广泛，详情请参考 OS 相关手册。本手册重点介绍开发工具入门以及区块链相关功能的使用。

开发基于区块链的 DAPP 不需要使用专用开发工具。

开发 SpaceChain OS 上层脚本应用，可以用脚本工具如 Python、Lua 等直接开发。

如果需要修改内核以及相关系统文件的源代码，并且需要重新编译的话，就需要使用对应的编译器。

目前，太空链专属的开发工具仍在研发中，该开发工具基于 RealEvo-IDE，是一个集成设计、开发、仿真、部署、测试一体的工具。太空链和翼辉信息一起对于其进行优化，并针对航天领域的应用和驱动进行优化和修改，适用于航天领域的芯片硬件平台。

未来，对于航天以及区块链的应用，需要使用太空链专属的工具，其他 OS 适用的领域，则需要使用 RealEvo-IDE 来进行开发。

SpaceChain OS 会作为一个独立分支单独进行开发维护，服务于航天以及区块链领域，作为基础设施的承载者，运行在太空以及对应的地面接收设备、控制设备上。

我们也希望社区踊跃维护、提交代码，太空链感谢爱好者的付出。

本文中，我们使用旧版本工具来进行开发演示。请注意本文随时更新，请注意访问太空链网站以便随时获得最新版本的更新。

1.2 RealEvo-IDE 安装

IDE 的获得路径：登录太空链官网，<https://spacechain.com/>，即可申请获得使用许可，许可会通过电子邮件发送。（预计 2018 年 3 月 31 日前开放）。网站开放前，您可以联系 info@spacechain.com 获取。

成功得到一套 SylixOS 开发套件后，可以认真地阅读光盘中的文档来获得关于 SylixOS 集成开发套件的更多信息。

以下过程是通过 SylixOS 开发套件光盘安装 RealEvo-IDE 过程：

打开 SylixOS IDE，双击打开 InstallWizard.exe 文件会看到 SylixOS 开发套件的安装工具集、注意事项及其 SylixOS 网站信息，可以看出 SylixOS IDE 包括“RealEvo-IDE”、“QtCreator”、“RealEvo-QtSylixOS”的安装。





图 1.1 集成开发环境工具集

如图 1.1 的顺序进行安装，集成开发环境中的 RealEvo-IDE 需要注册方可正常使用，具体的注册过程和注册需要注意的地方可查看光盘中的文件《RealEvo 软件注册步骤》。

执行完上面的过程，就完成了集成开发环境的安装。

1.3 RealEvo-IDE 注册

1.3.1 RealEvo 系列软件 license 说明

RealEvo 系列软件由以下独立软件组成：

- RealEvo-IDE：需独立注册；
- RealEvo-Simulator：不需要注册；
- RealEvo-QtSylixOS：不需要注册。

1.3.2 在线注册步骤

1. 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序列号（序列号跟随光盘或通过邮件获取）。

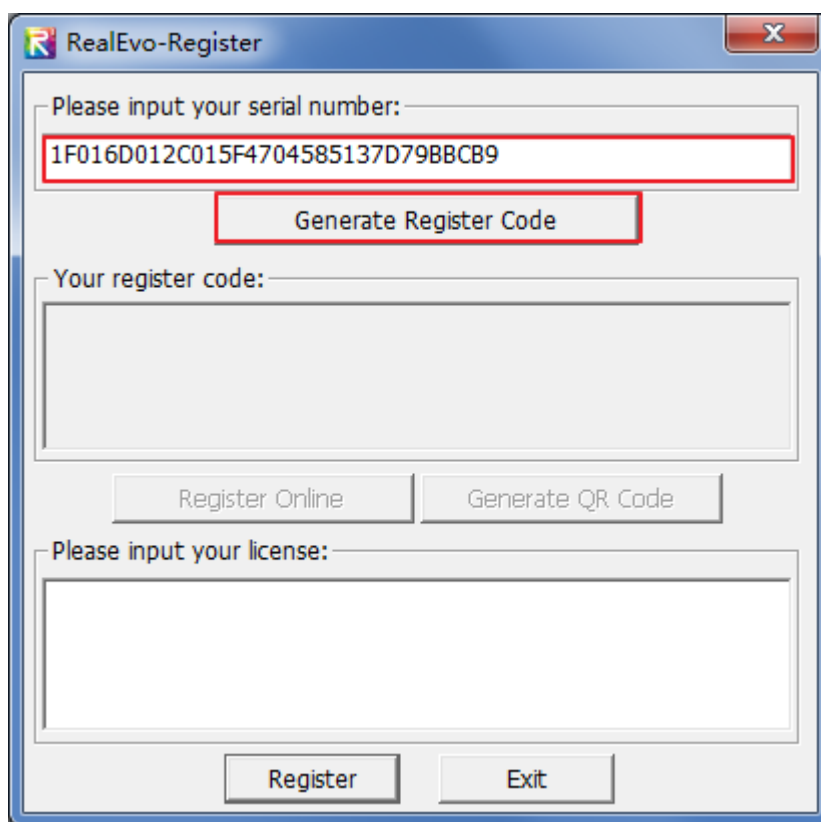


图 1.2 输入 SN

2. 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，如图 1.3。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。

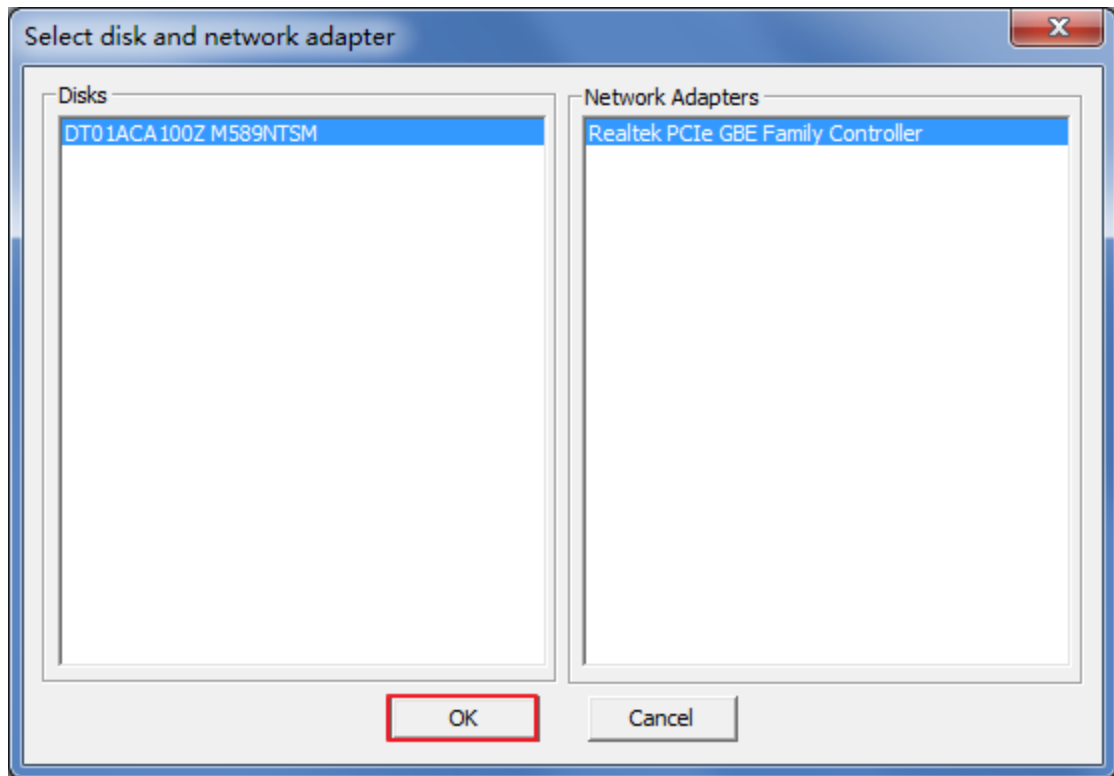


图 1.3 选择硬盘和网卡

3. 选择好硬盘和网卡后点击“OK”按钮生成注册码，如图 1.4。

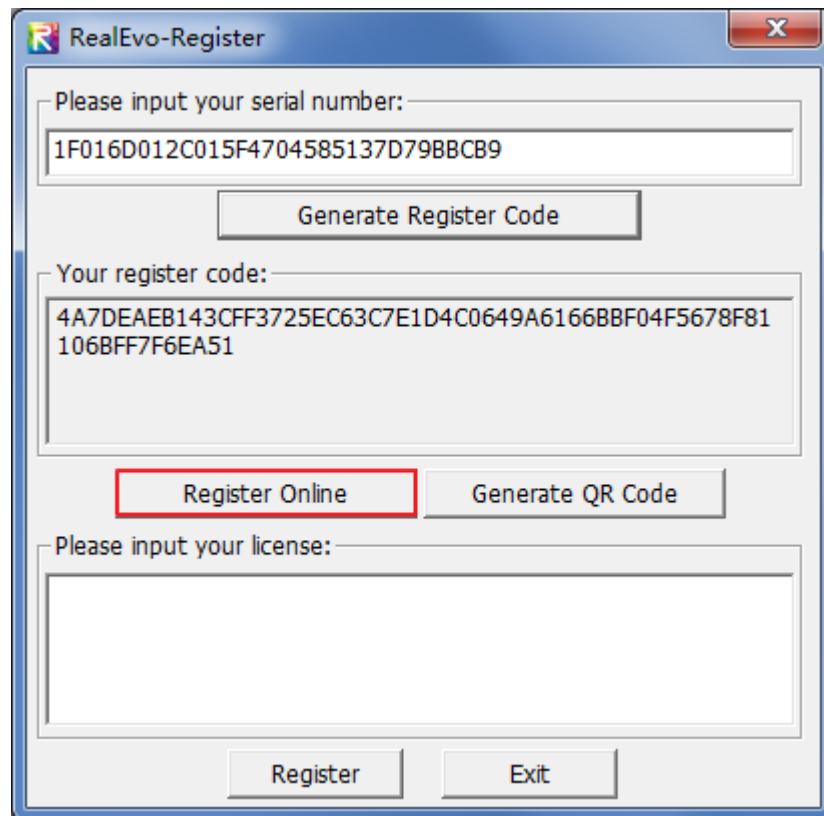


图 1.4 生成注册码

4. 如果计算机已经连接互联网，点击“Register Online”按钮可获取 license，如图 1.5。

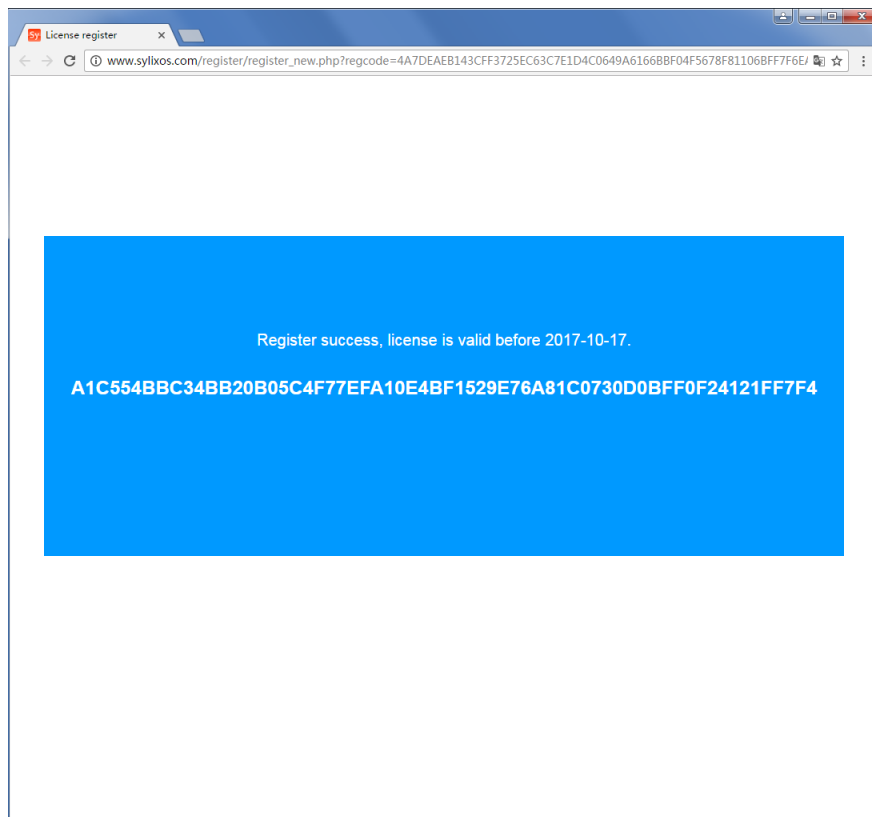


图 1.5 生成 license

复制图 1.5 中生成的 license, 粘贴到 RealEvo-Register 工具下方 license 输入框, 如图 1.6。

注：当软件的 license 过期后，可以通过启动 RealEvo-Register.exe 工具重启注册新的 license。

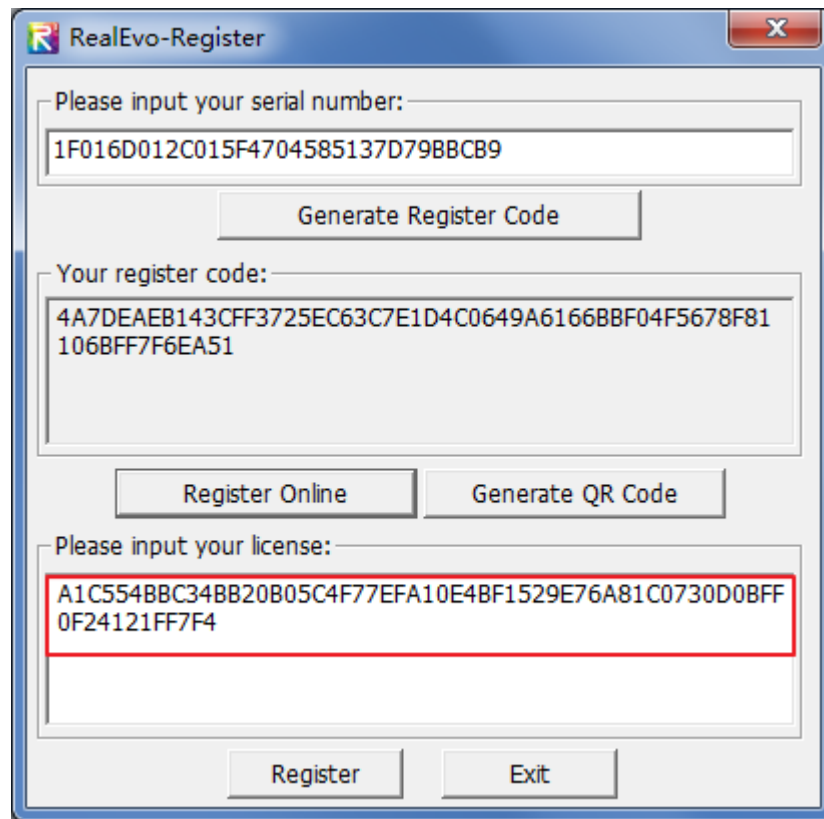


图 1.6 输入 license

点击“Register”按钮完成注册，如图 1.7。

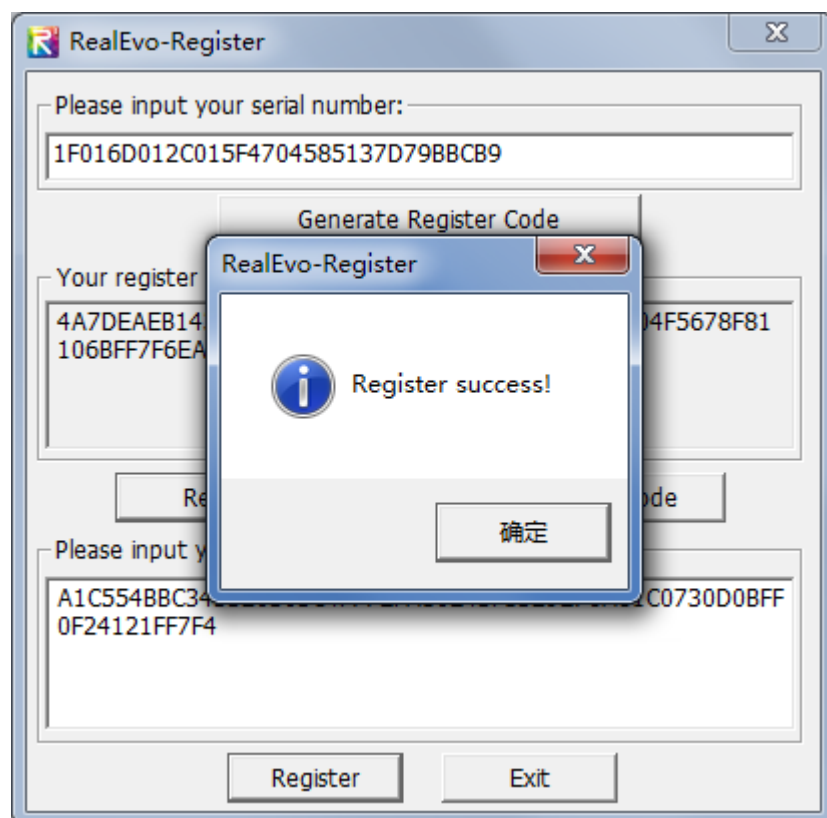


图 1.7 完成注册

第2章 区块链工程导入部署

2.1 硬件准备

本文以米尔科技 Z-turn 开发板为例。该开发板核心 CPU 是 ARM A9 双核。新的 Z-turn 开发板第一次启动 SpaceChainOS 后，需要进行分区格式化操作，如图图 2.1。

```
[unknown@sylixos:/]# umount /media/sdcard0
[unknown@sylixos:/]# fdisk -f /dev/blk/sdcard-0
block device /dev/blk/sdcard-0 total size: 3812 (MB)
please input how many partition(s) you want to make (1 ~ 4) : 2
please input how many bytes align (4k 8k ...) : 4096
please input the partition 0 size percentage(%) 0 means all left space : 20
is this partition active(y/n) : n
please input the file system type
1: FAT 2: TPSFS 3: LINUX 4: RESERVED
1
please input the partition 1 size percentage(%) 0 means all left space : 0
is this partition active(y/n) : y
please input the file system type
1: FAT 2: TPSFS 3: LINUX 4: RESERVED
2
making partition...
block device : /dev/blk/sdcard-0
block type : SD/MMC
block serial : 00002F3F
block firmware: 2017.10, v0.0
block product : 00000 SDHC memory card
block media : SDHC(v2.0)

partition >>
PART ACT SIZE(KB) OFFSET(KB) TYPE
-----
0      780492 1024 win95 FAT32 Partition
1 *    3121972 781516 SylixOS True Power Safe Partition

total partition 2
[unknown@sylixos:/]# remount /dev/blk/sdcard-0
Block device /dev/blk/sdcard-0 part 0 mount to /media/sdcard0 use vfat file system.
Magic number error, mount failed
Block device /dev/blk/sdcard-0 part 1 mount to /media/sdcard1 use tpsfs file system.
[u/]# mkfs /media/sdcard0
now format media, please wait...
disk format ok.
[unknown@sylixos:/]# mkfs /media/sdcard1
now format media, please wait...
disk format ok.
```

图 2.1 分区格式化操作

注：格式化后需将引导文件 BOOT.bin 上传到 /boot 目录下，并执行 sync 命令。重启后可正常挂载文件系统。

2.2 创建 Base 工程

选择菜单“File→New→Project”，弹出窗口中列出了 SpaceChainOS 支持的工程类型，如图 2.2。



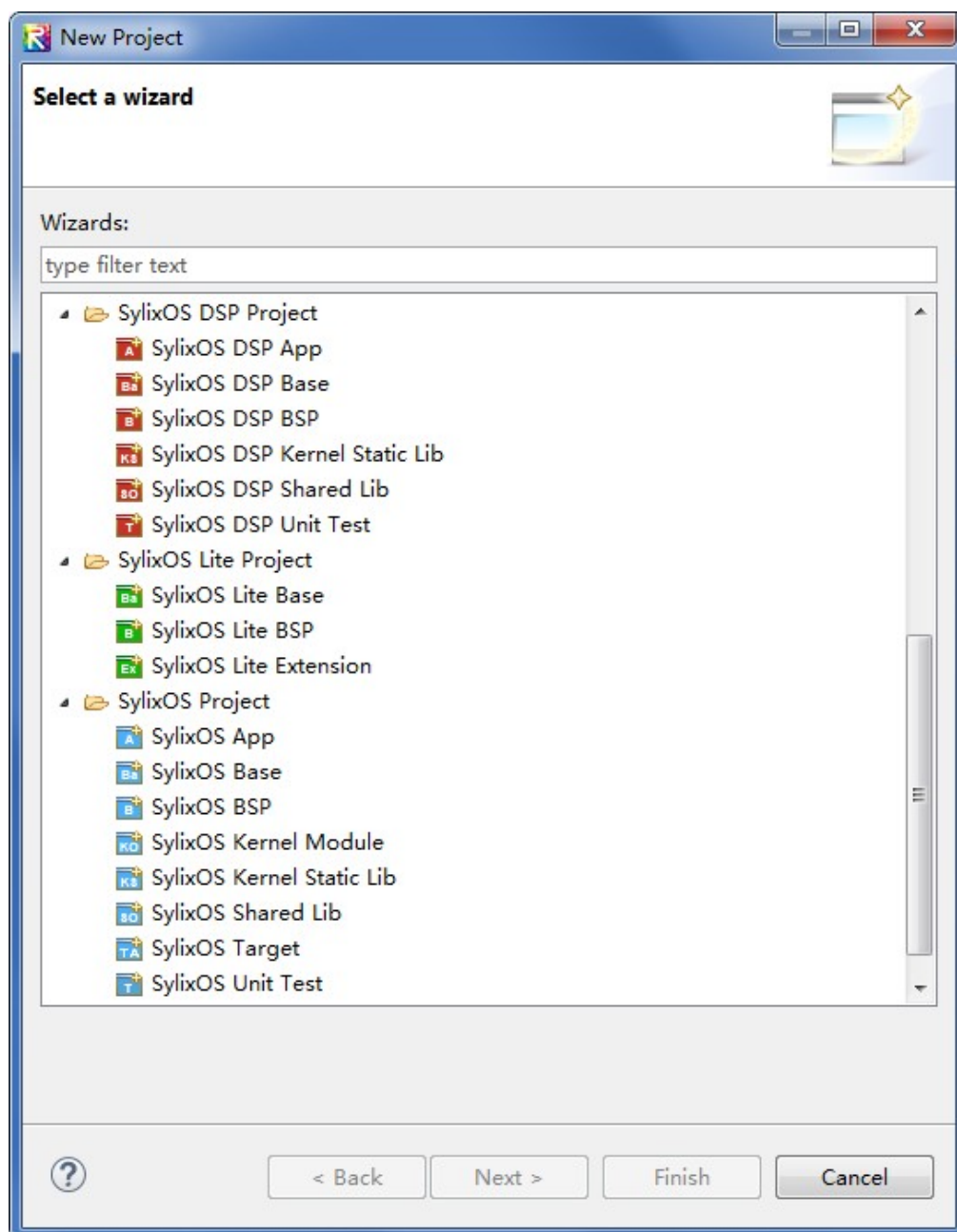


图 2.2 创建工程

在 中选择“SylixOS Base”，点击“Next”，进入图 2.3 所示配置页面。在“Project name”输入框输入工程名。

注：工程名称不允许包含空格，取消勾选 “Use default location”，可在默认 Workspace 之外的位置创建工程。

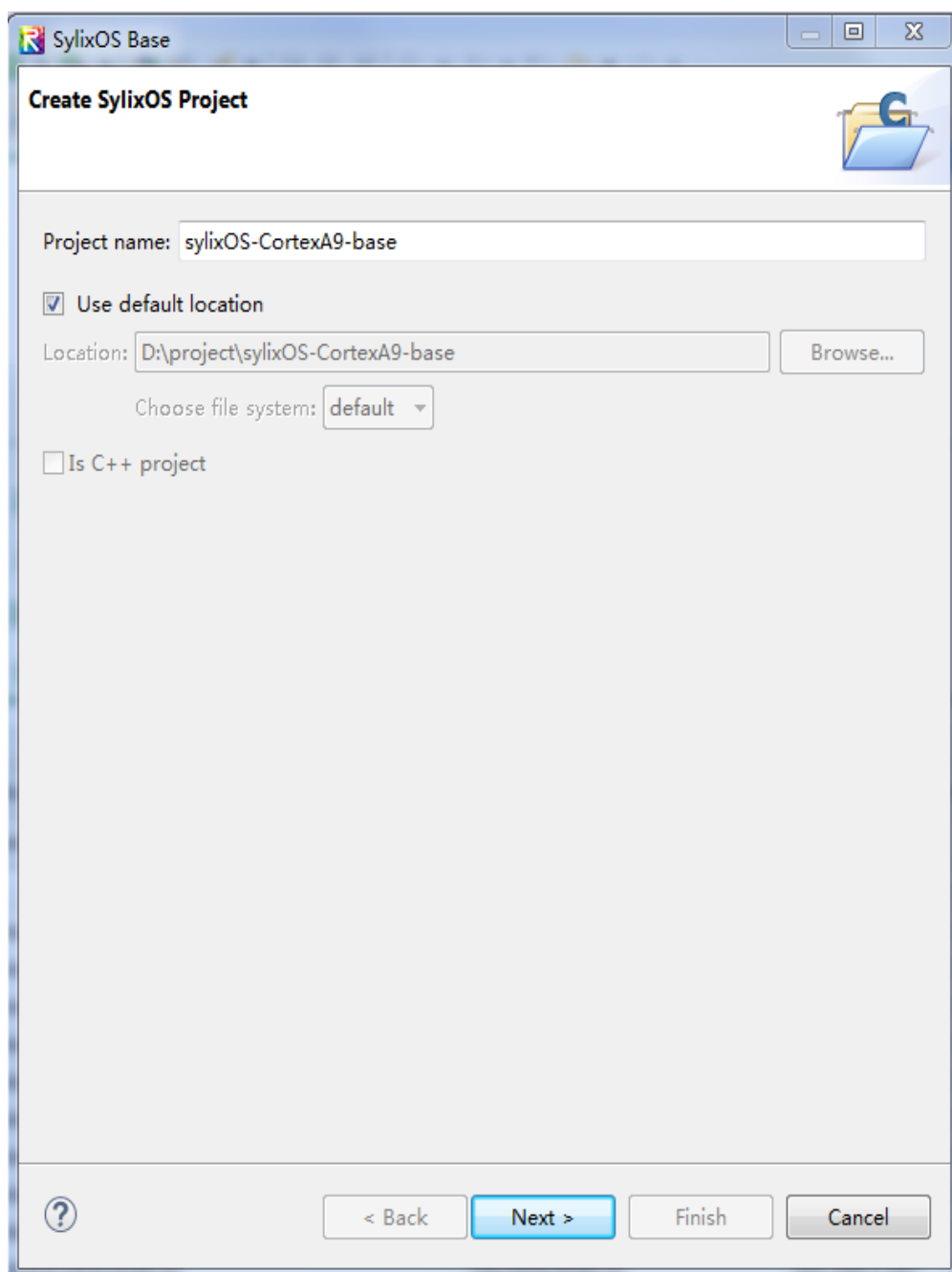


图 2.3 创建 SylixOS Base 工程

点击“Next”进入工具链设置页面，设置基础编译选项，含：工具链（选择 mips64-sylixos-toolchain）、调试级别、处理器类型和浮点类型设置，如图 2.4

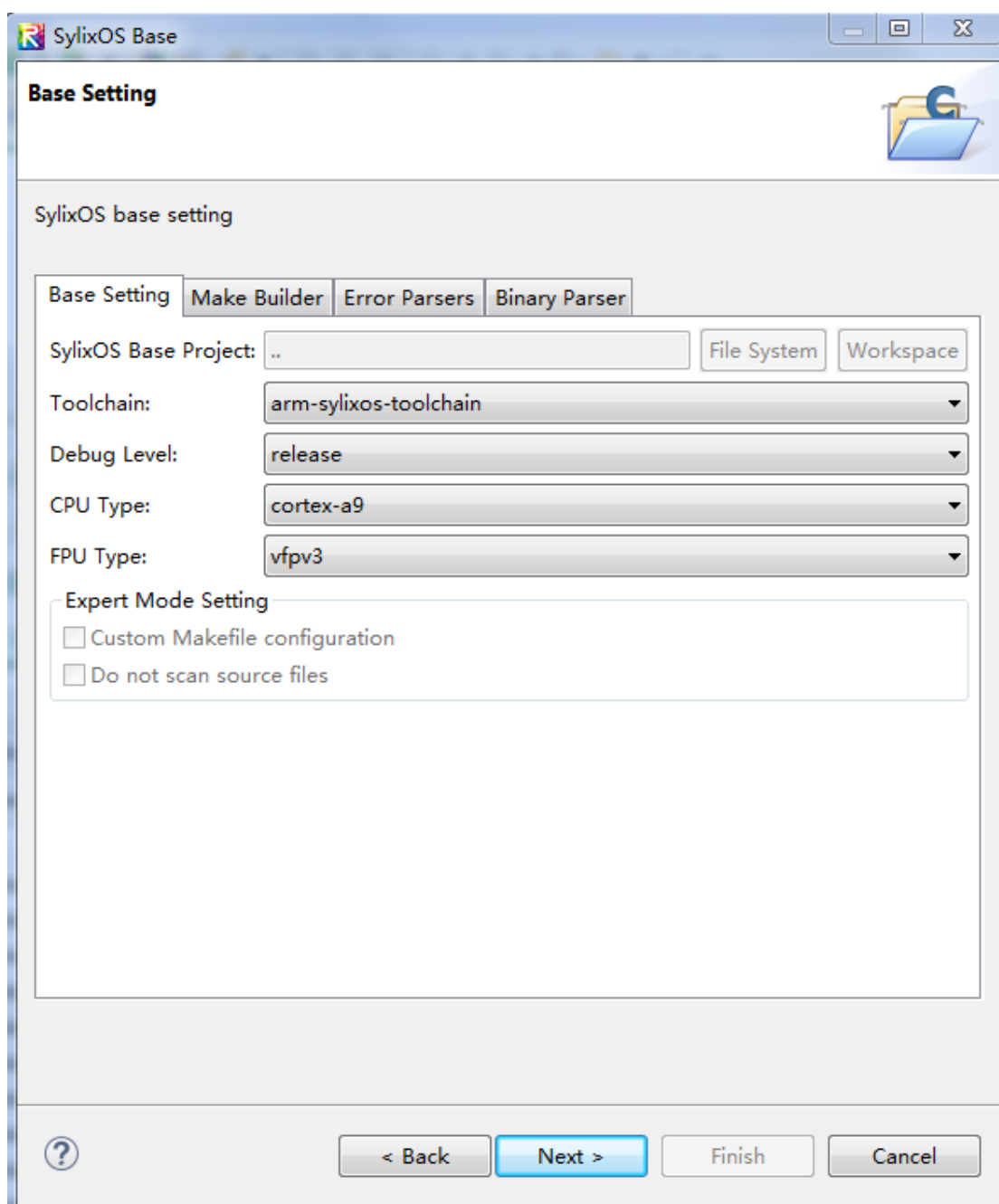


图 2.4 Base Project 通用设置

配置项解析：

- Toolchain: 工具链（选择 mips64-sylixos-toolchain 编译 64 位系统，选择 mips-sylixos-toolchain 编译 32 位系统）；
- Debug Level: 调试级别，SylixOS 提供 Debug 和 Release 两种配置；
- CPU Type: 处理器型号；
- FPU Type: 浮点处理器；

点击“Next”进入组件选择页面，SylixOS 包含大量可选组件，如图 2.5。选中图中列表左侧复选框可包含对应组件到 SylixOS Base 工程，选择“Select All”按钮可选中全部

组件。这里点“Finish”即可完成工程创建。

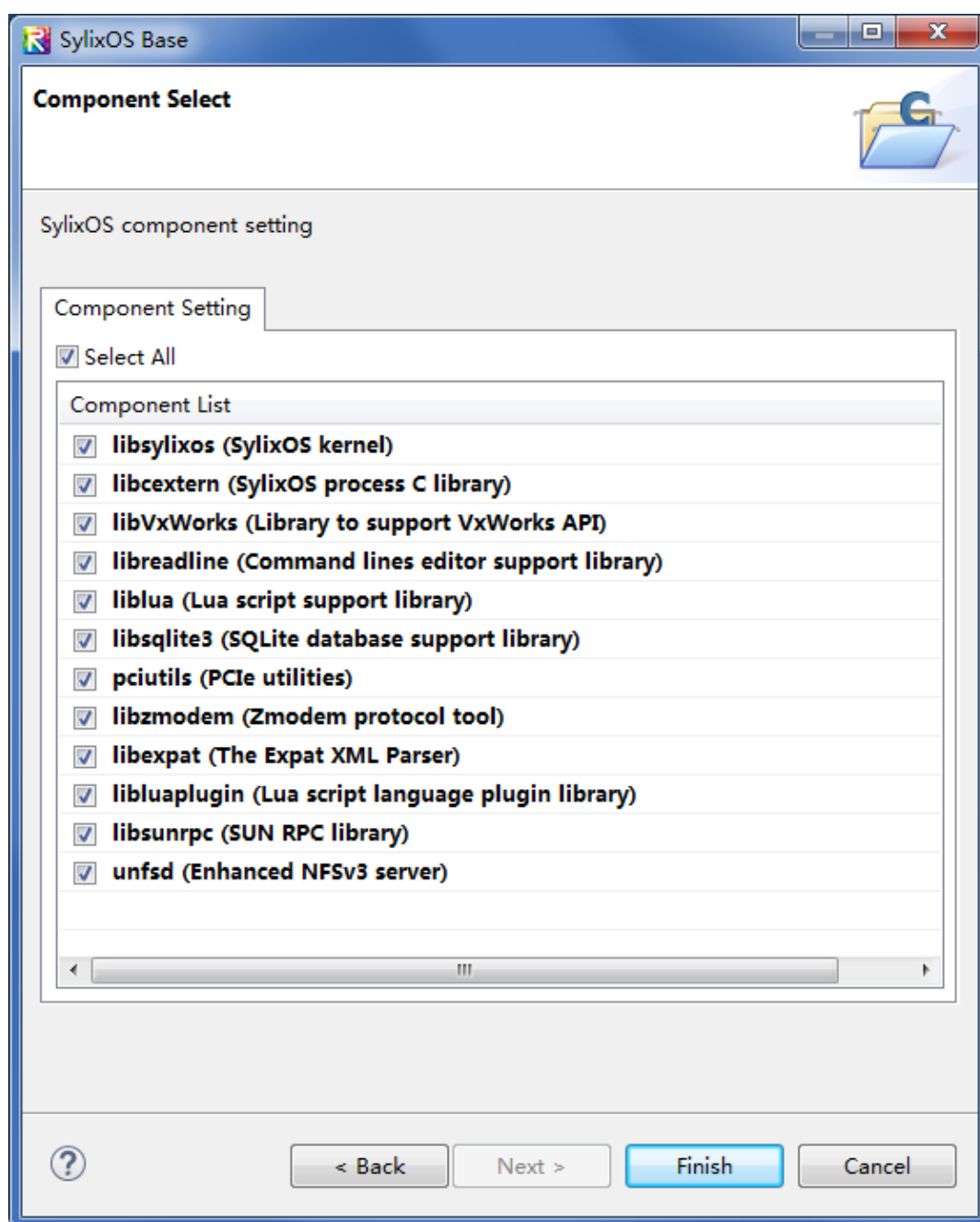


图 2.5 SylixOS Base 选择组件

点击创建的 Base 工程右键选择“Build Project”开始编译 SylixOS Base 工程。

注：SylixOS Base 工程的详细创建信息可以参考《RealEvo-IDE 使用手册》。

2.3 工程导入

区块链 BASE 工程创建后，还需要通过 RealEvo-IDE 导入应用程序工程 qtum 和库文件工程，需要导入的库文件工程包括：



1. Libboost
2. Libdb
3. Libevent
4. libopenssl

本文档以 libopenssl 工程为例演示工程导入。选择菜单“File→Import”，打开 Import 对话框，如图 2.6.6

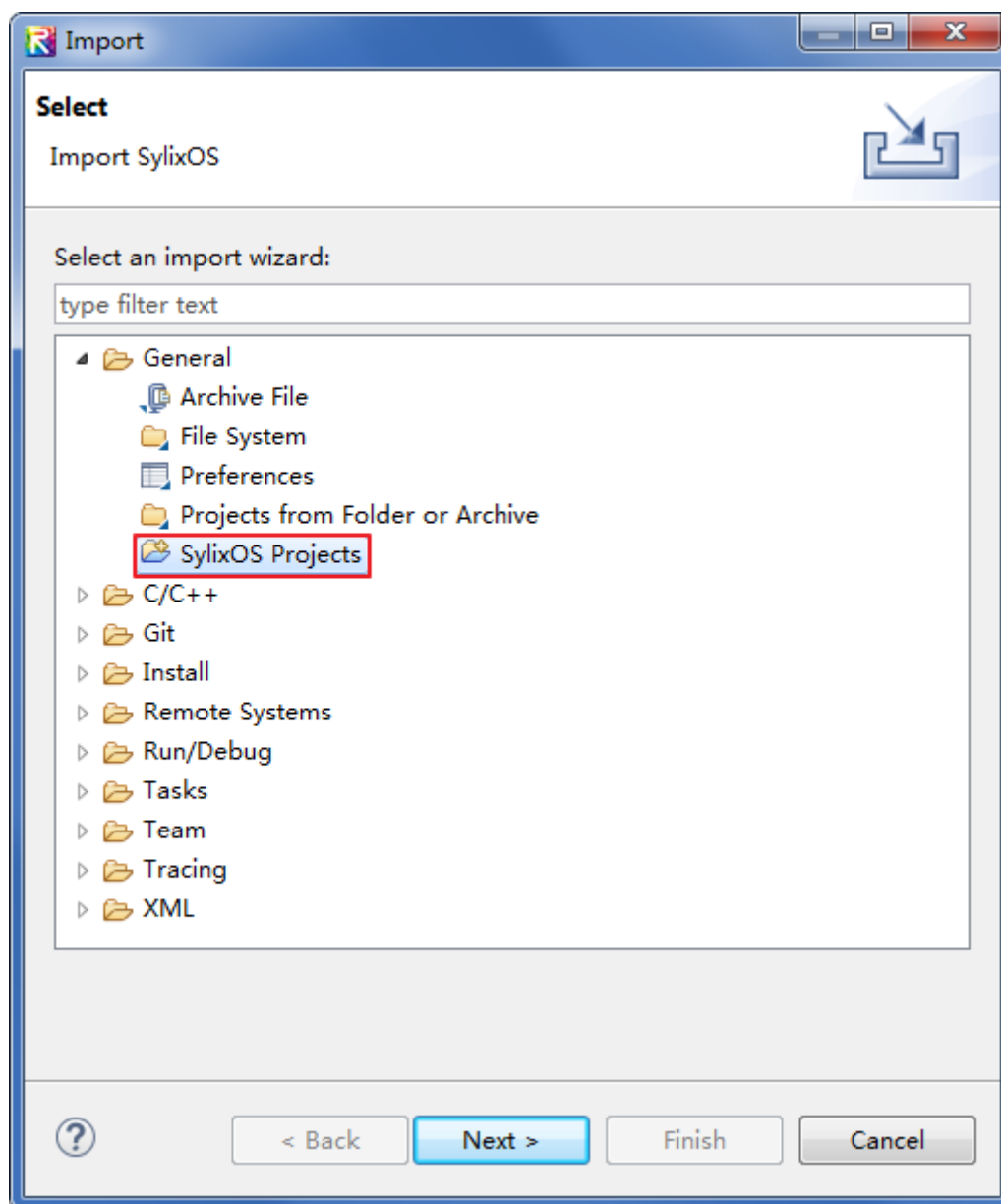


图 2.6 导入工程

选中“SylixOS Projects”，点击“Next”按钮进入工程选择界面，如图 2.7。

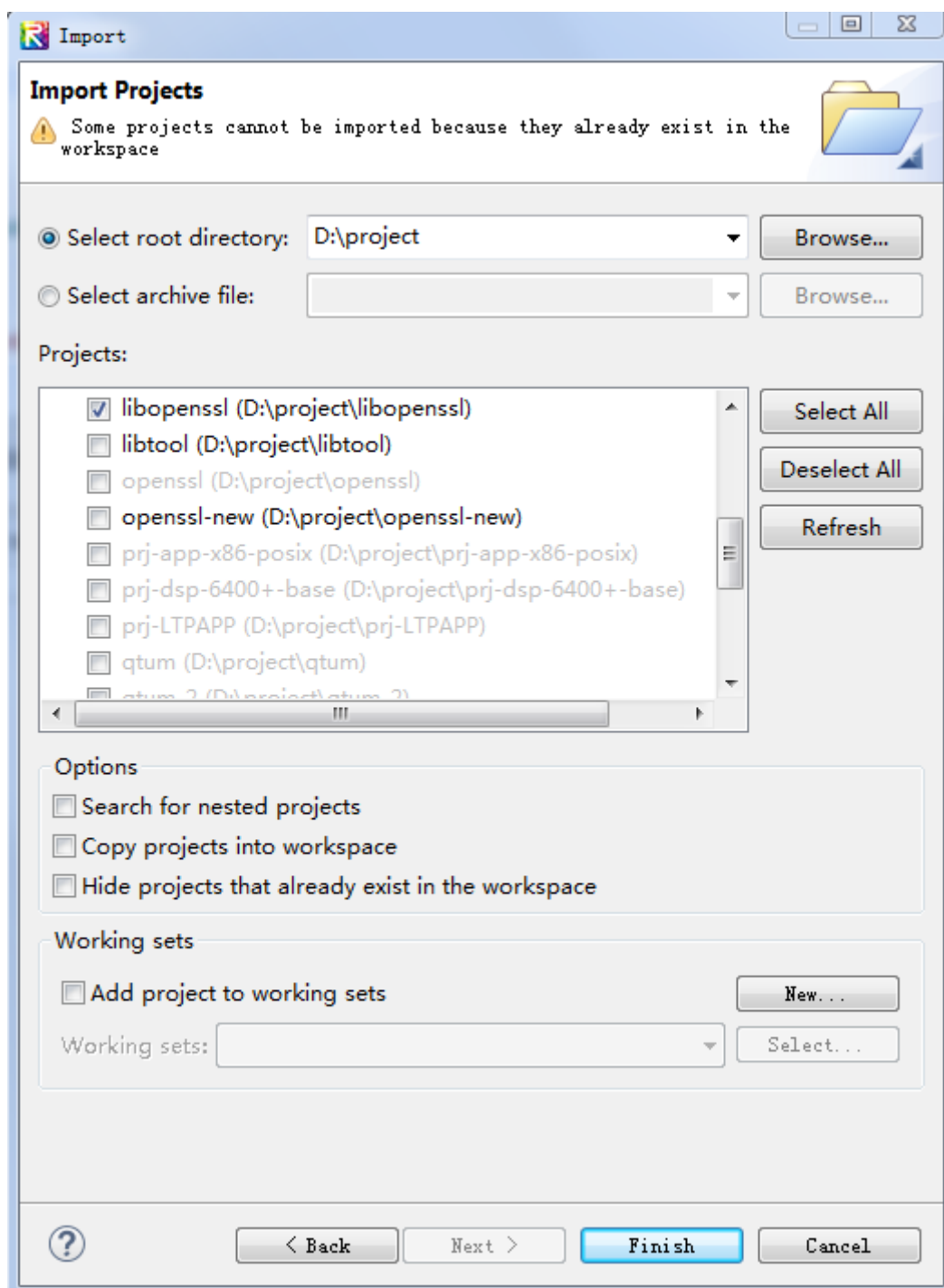


图 2.7 选择导入工程

点击“Browse”按钮，在弹出文件夹选择框中选中工程所在目录，在 Projects 列表中会列出该目录下可被导入的工程名称，选中需要导入的工程，点击“Finish”按钮完成工程导入。

导入工程成功后，可在 Workspace 工程列表中看到新导入的工程 libopenssl，如图 2.8。

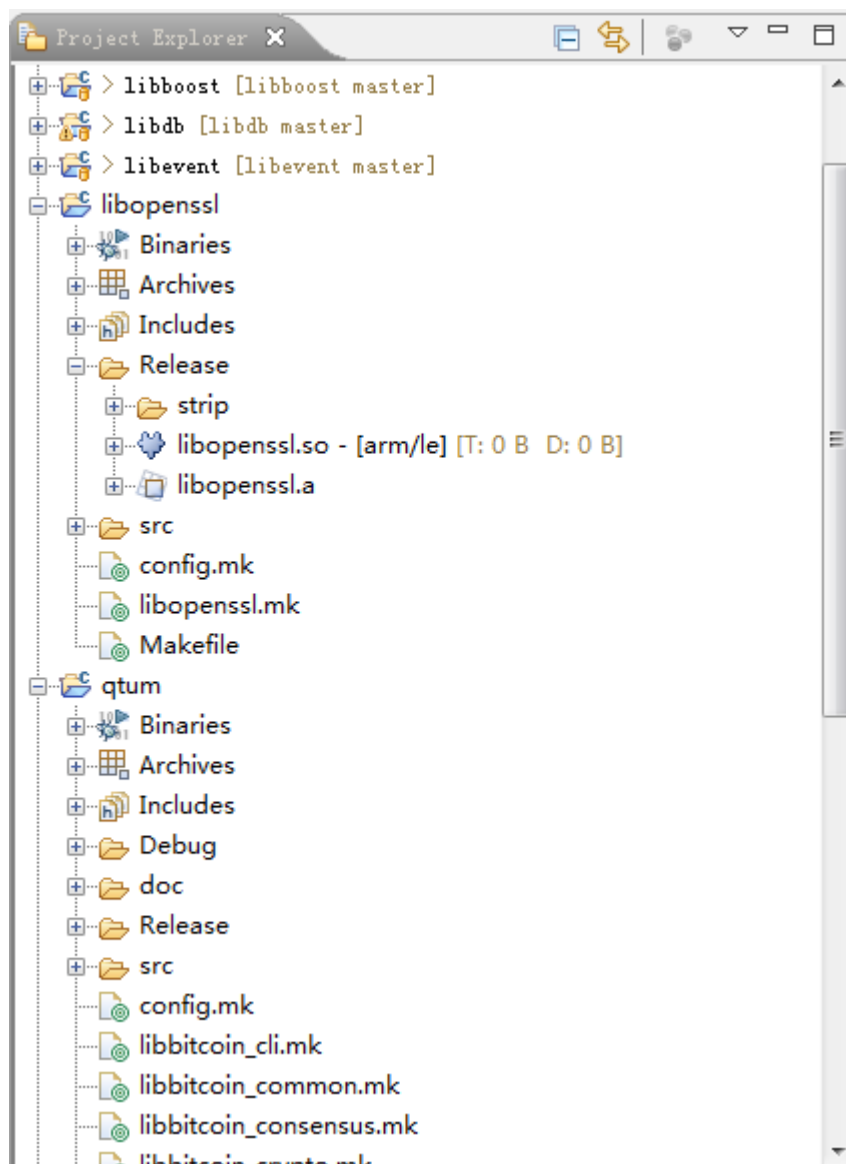


图 2.8 工程导入完成

2.4 工程部署

本项目需要分别部署下列库文件：

- Libopenssl
- Libboost
- Libdb
- Libevent

库文件部署完成后可以部署应用程序 qtum；

可使用 RealEvo-IDE 的一键部署功能实现程序部署（以 libopenssl 为例），右键 libopenssl 工程，选择菜单“Properties”打开工程属性页，选择“SylixOS Project→Device Setting”选项卡，可查看和更改部署配置。在创建 SylixOS App 工程时，RealEvo-IDE 默认会添加工程当前配置输出文件夹下与工程名同名的文件到文件列表。

如图 2.9 所示，可以选择 Workspace 中已添加的任一存在的设备部署文件。

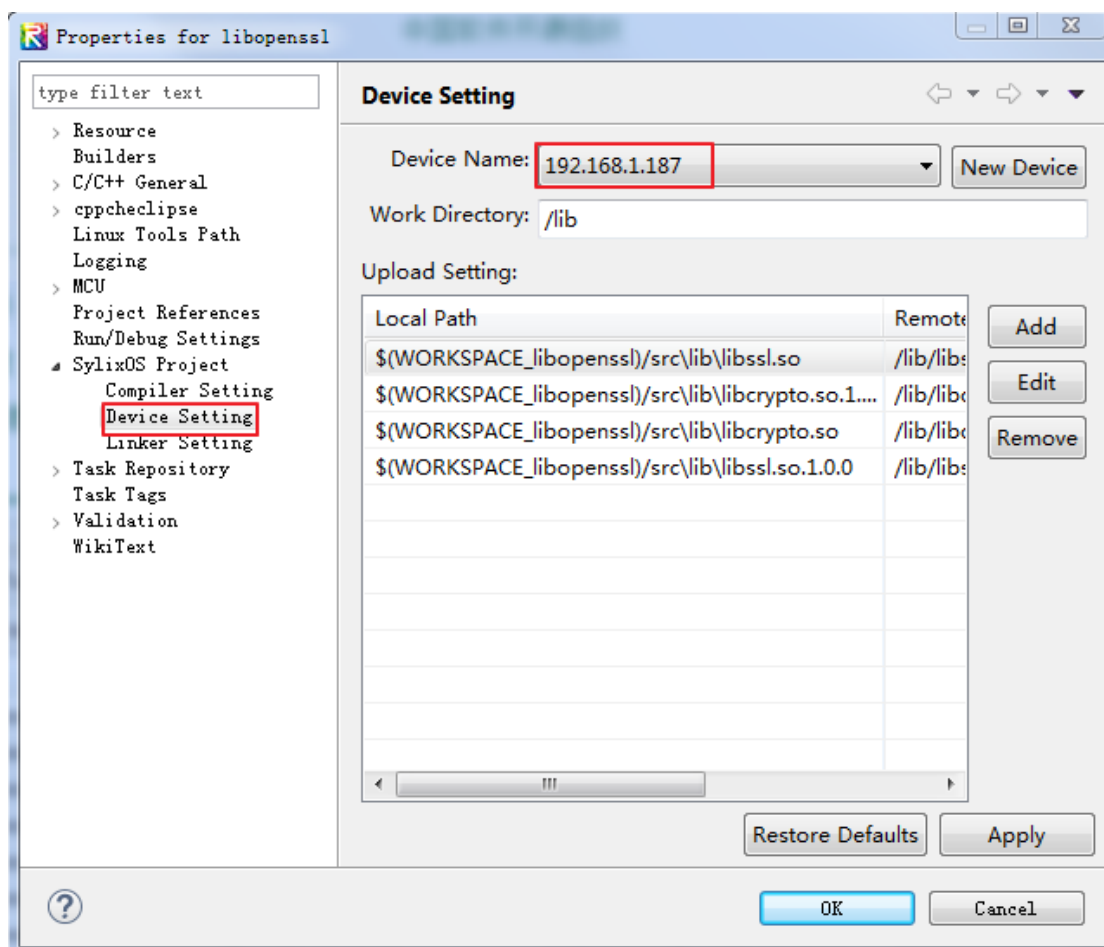


图 2.9 SylixOS libopenssl 工程部署设置

设置完成后，右键工程，选择菜单“SylixOS→Upload”部署工程。

注：如果部署不成功需要检查防火墙是否处于关闭状态。

登录到目标板系统，切换到“/apps/qtum”目录，运行 qtum 的命令，如图 2.10，具体命令执行见第三章。

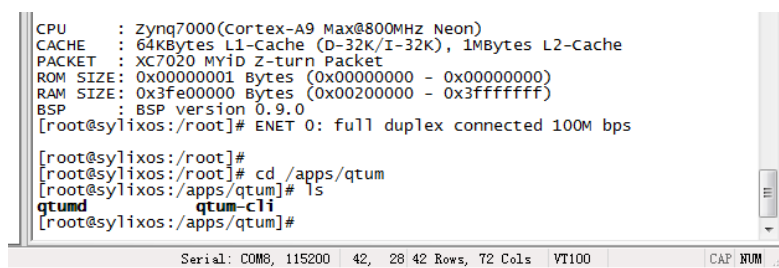


图 2.10 进入 qtum 环境，执行 qtum 命令

2.5 工程删除

右击待删除工程，选择菜单“Delete”可删除工程，如图 2.11。

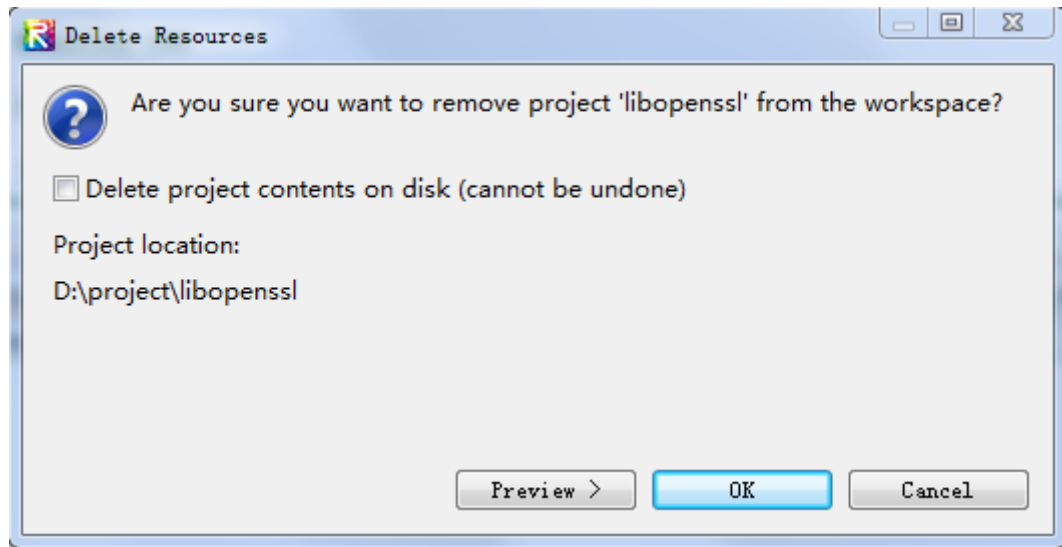


图 2.11 删除工程

第3章 区块链钱包基本使用

3.1 常用命令使用

Qtum 软件提供的命令很多，本文档以-testnet 模式下部分常用命令为例演示其使用方法和结果。

3.1.1 获得本节点信息

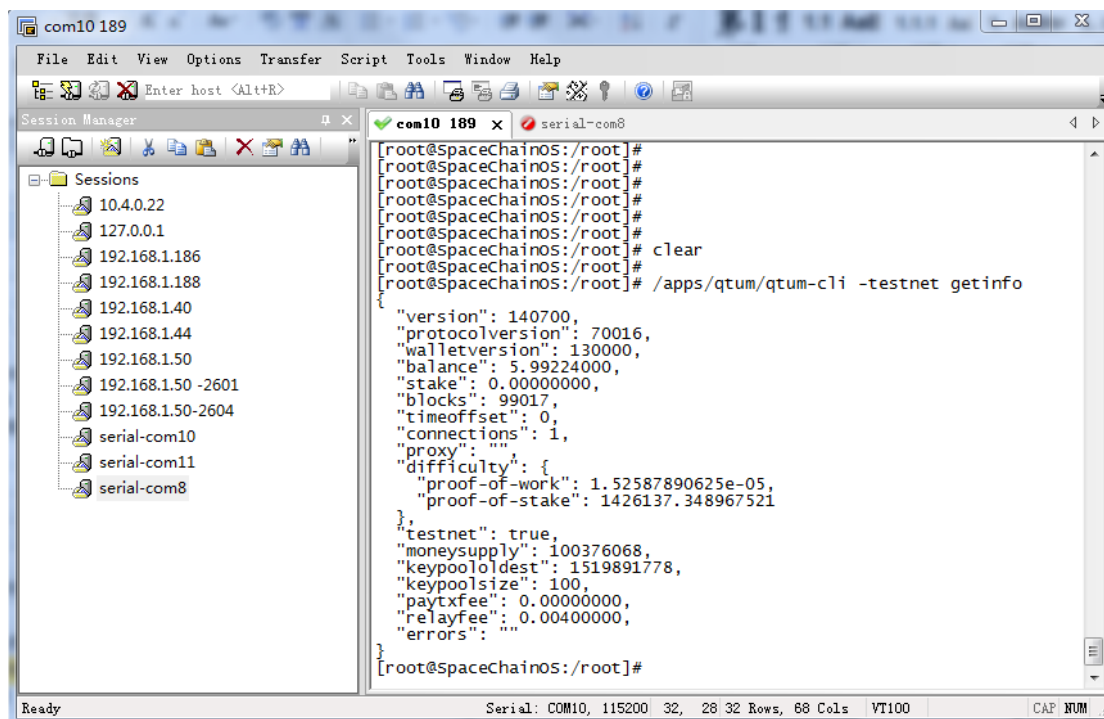


图 3.1 获得本节点信息命令演示

3.1.2 获得邻居节点信息

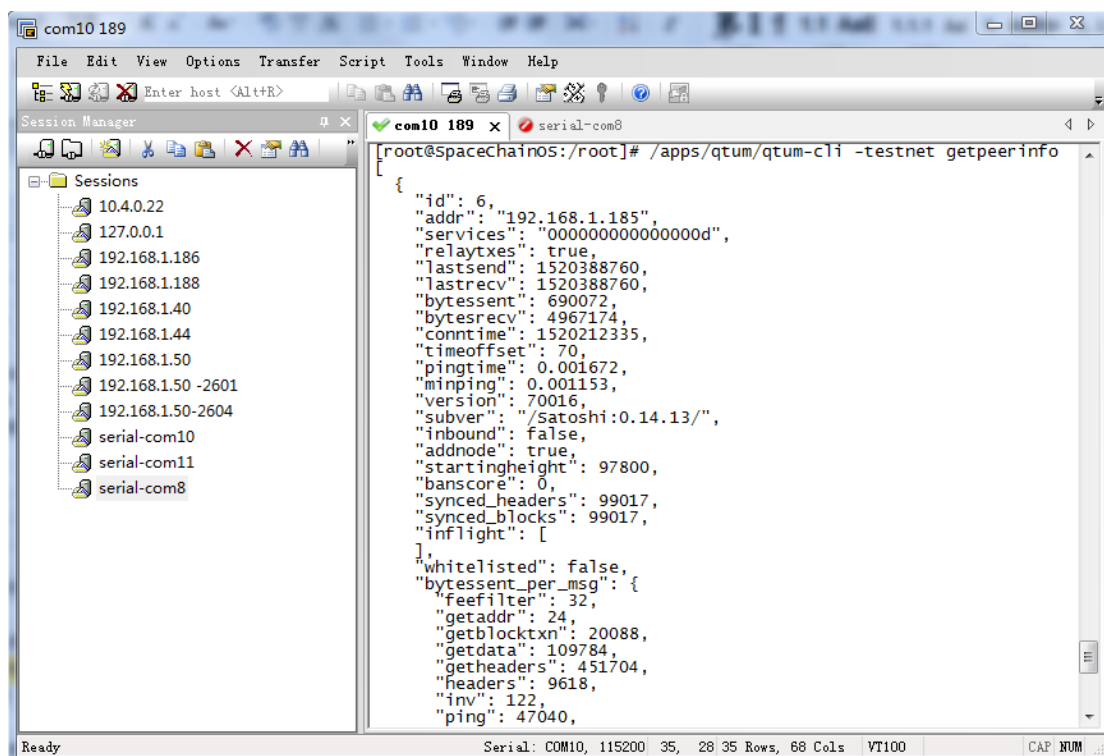


图 3.2 获得邻居节点信息命令演示

3.1.3 转账

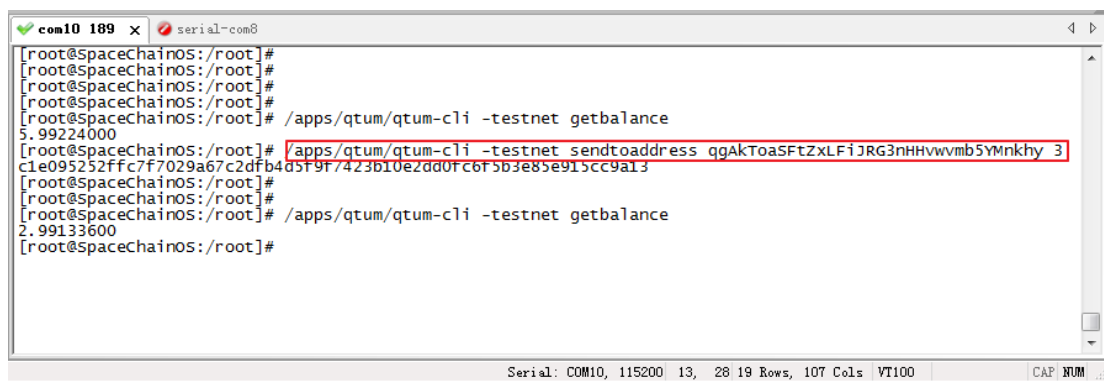
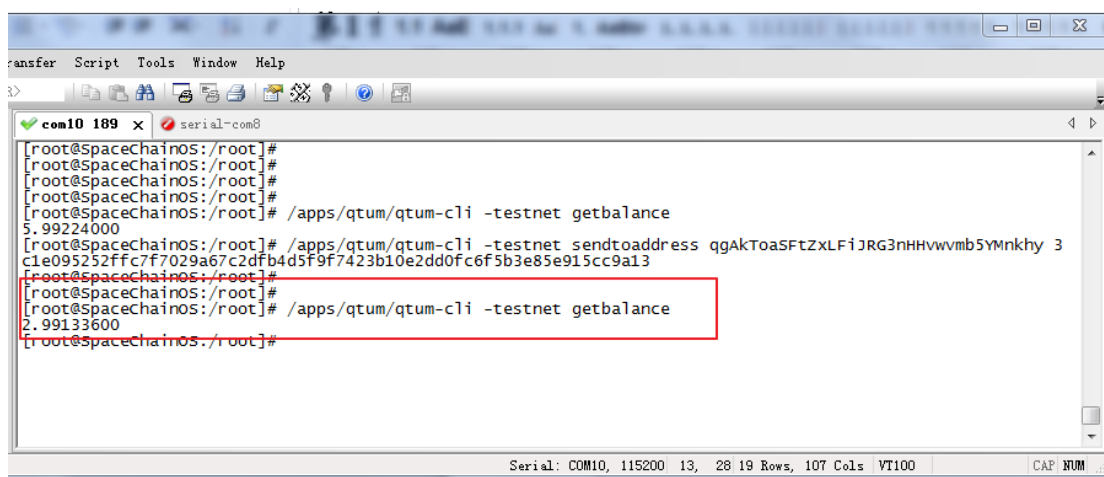


图 3.3 转账命令演示

3.1.4 转账后查看余额:

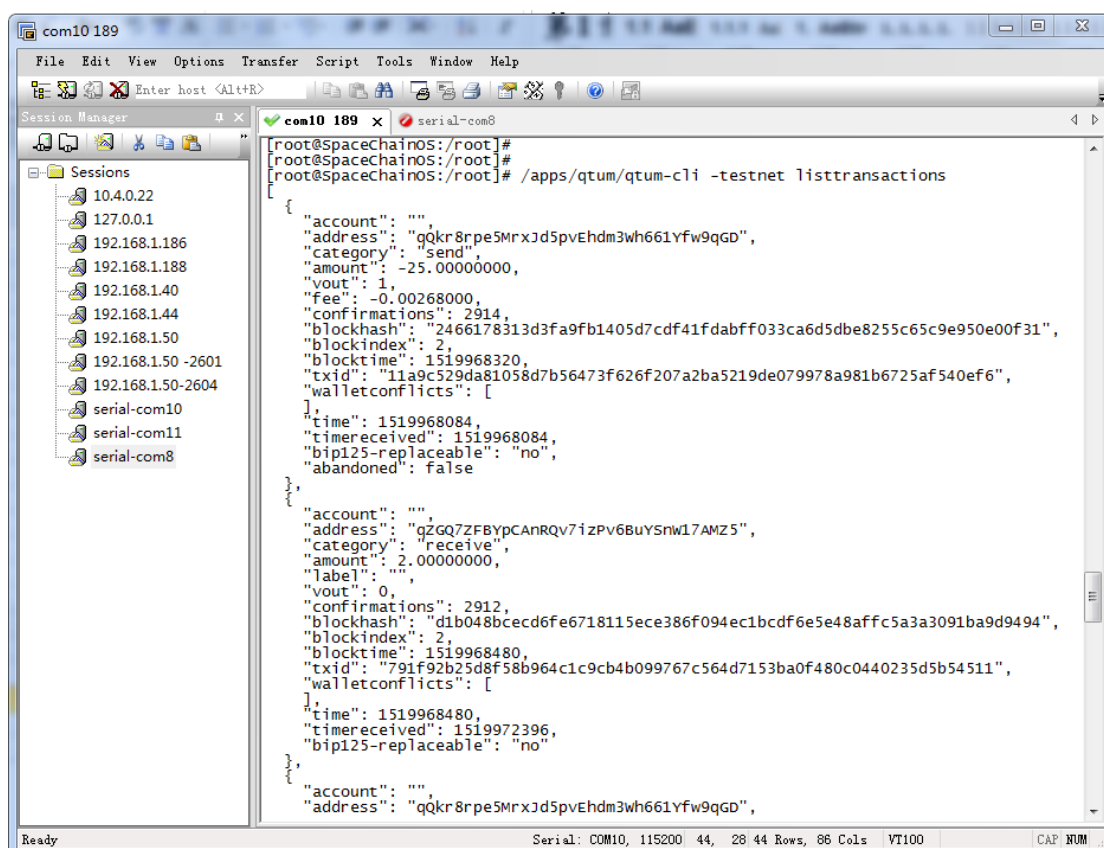


```

[com10 189 x] serial-com8
[root@spacechainos:/root]#
[root@spacechainos:/root]#
[root@spacechainos:/root]#
[root@spacechainos:/root]# /apps/qtum/qtum-cli -testnet getbalance
5.99224000
[root@spacechainos:/root]# /apps/qtum/qtum-cli -testnet sendtoaddress qgAkToaSFTzXLFjRG3nHHVwvmb5YMnky 3
cle095252ffc7f7029a67c2dfb4d5f9f7423b10e2dd0fc6f5b3e85e915cc9a13
[root@spacechainos:/root]#
[root@spacechainos:/root]# /apps/qtum/qtum-cli -testnet getbalance
2.99133600
[root@spacechainos:/root]#
  
```

图 3.4 转账后余额命令演示

3.1.5 查询交易记录



```

[com10 189 x] serial-com8
[root@spacechainos:/root]#
[root@spacechainos:/root]# /apps/qtum/qtum-cli -testnet listtransactions
[
  {
    "account": "",
    "address": "qQkr8rpe5MrxJd5pvEhdm3Wh661Yfw9qGD",
    "category": "send",
    "amount": -25.00000000,
    "vout": 1,
    "fee": -0.00268000,
    "confirmations": 2914,
    "blockhash": "2466178313d3fa9fb1405d7cdf41fdabff033ca6d5dbe8255c65c9e950e00f31",
    "blockindex": 2,
    "blocktime": 1519968320,
    "txid": "11a9c529da81058d7b56473f626f207a2ba5219de079978a981b6725af540ef6",
    "walletconflicts": [
    ],
    "time": 1519968084,
    "timereceived": 1519968084,
    "bip125-replaceable": "no",
    "abandoned": false
  },
  {
    "account": "",
    "address": "qZGQ7ZFBypCAnRQV7izPV6BuYSnW17AMZ5",
    "category": "receive",
    "amount": 2.00000000,
    "label": "",
    "vout": 0,
    "confirmations": 2912,
    "blockhash": "d1b048bcecd6fe6718115ece386f094ec1bcd6e5e48affc5a3a3091ba9d9494",
    "blockindex": 2,
    "blocktime": 1519968480,
    "txid": "791f92b25d8f58b964c1c9cb4b099767c564d7153ba0f480c0440235b54511",
    "walletconflicts": [
    ],
    "time": 1519968480,
    "timereceived": 1519972396,
    "bip125-replaceable": "no"
  },
  {
    "account": "",
    "address": "qQkr8rpe5MrxJd5pvEhdm3Wh661Yfw9qGD",
  }
]
  
```

图 3.5 交易查询命令演示

3.1.6 获得本节点收款地址

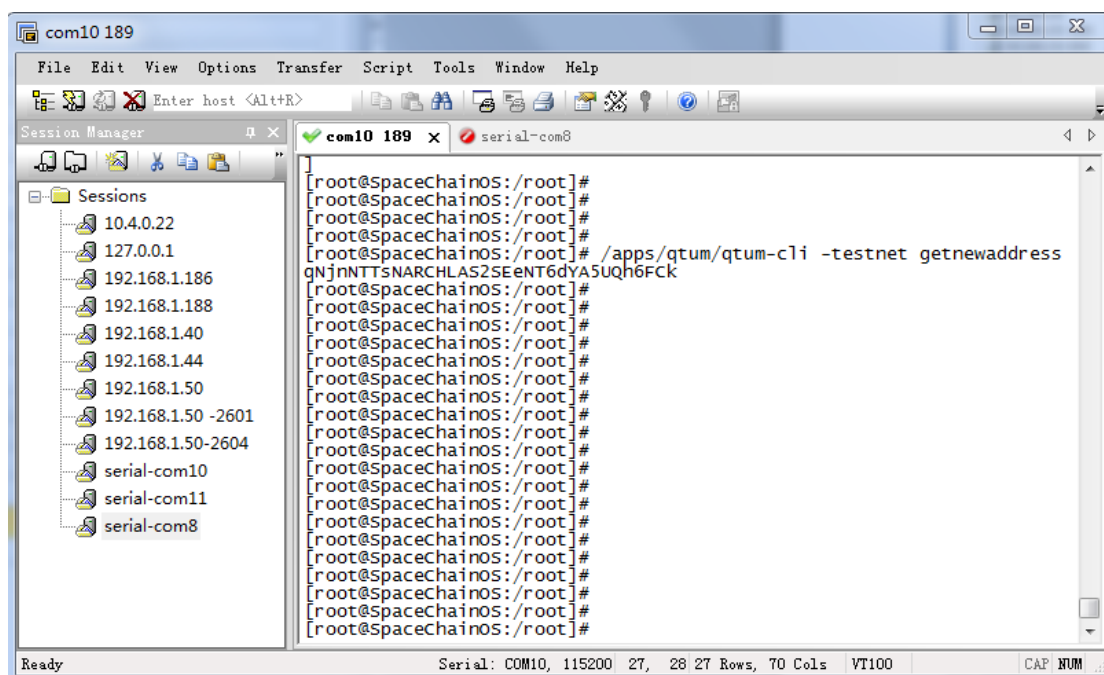


图 3.6 获得本节点收款地址命令演示