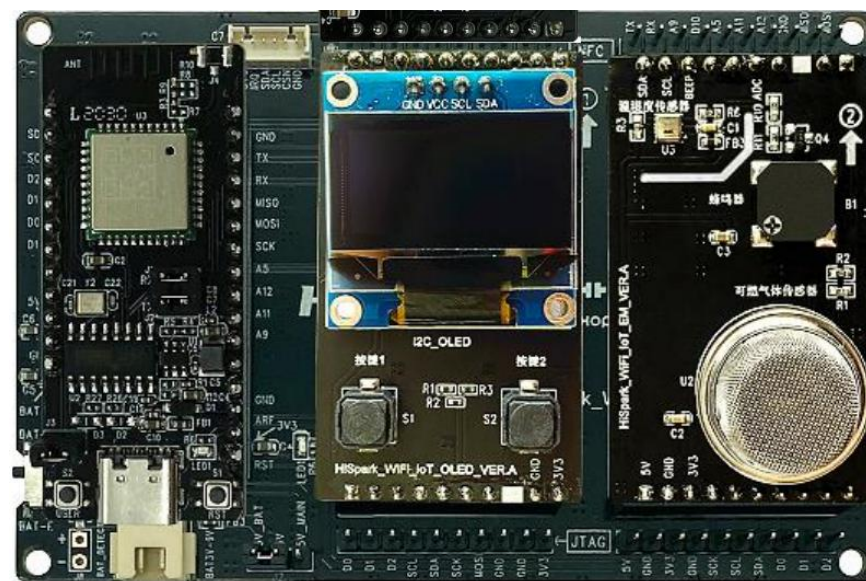


Harmony OS 智能硬件 入门系列课程 <快速上手>

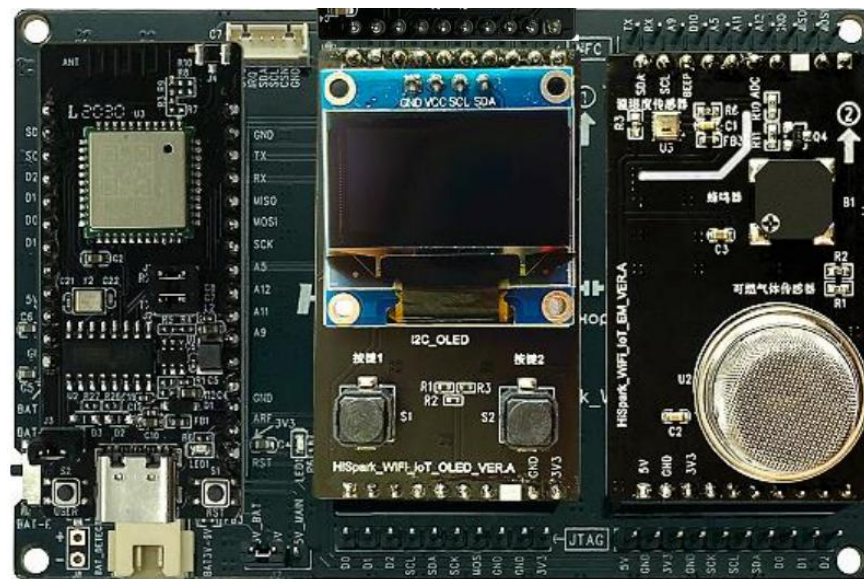
快速掌握Hi3861开发板基础开发技巧

第 1 讲：Hi3861开发环境搭建



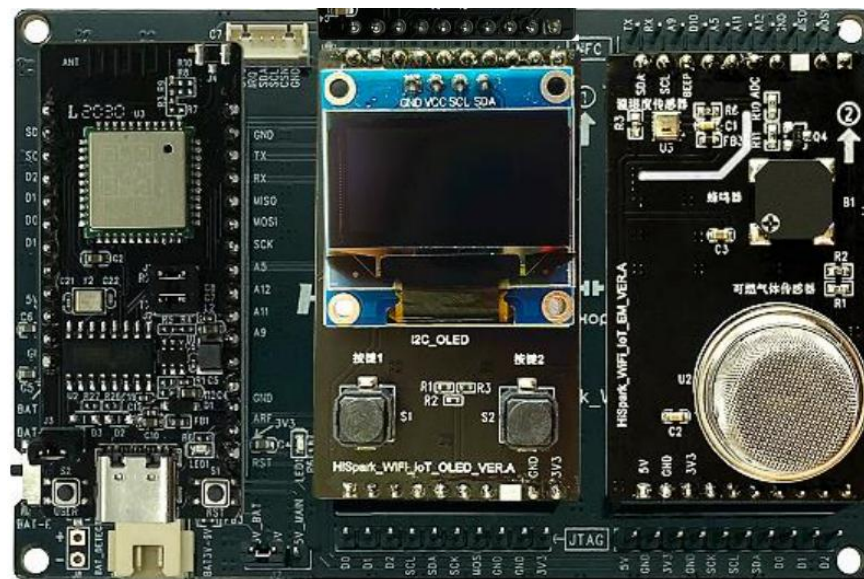
本讲内容

- 第1节：课程介绍
- 第2节：环境搭建
- 第3节：HarmonyOS 源码下载及编译
- 第4节：烧写固件及测试



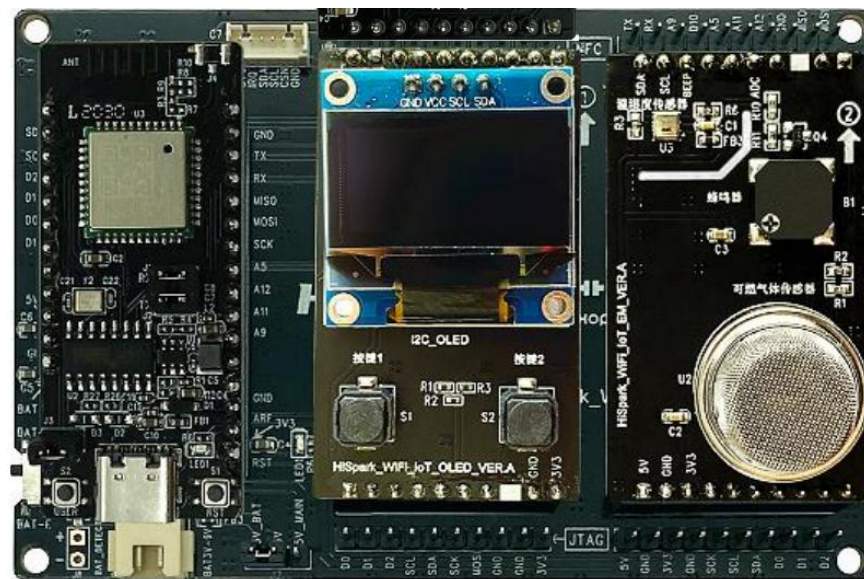
本讲目标

- 1、了解课程相关知识
- 2、能搭建Linux服务器环境
- 3、能搭建Window下开发环境
- 4、能下载及编译HarmonyOS 源码
- 5、掌握HiBurn工具烧录固件到开发板方法
- 6、会在windows下使用串口终端与开发板交互



第1节：课程介绍

- 知识点1：开发板Hi3861介绍
- 知识点2：开发模式
- 知识点3：课程内容



知识点1 【开发板介绍】



知识点1 【开发板介绍】

显示板



环境监
测板



智能红
绿灯板



智能(炫
彩)灯板



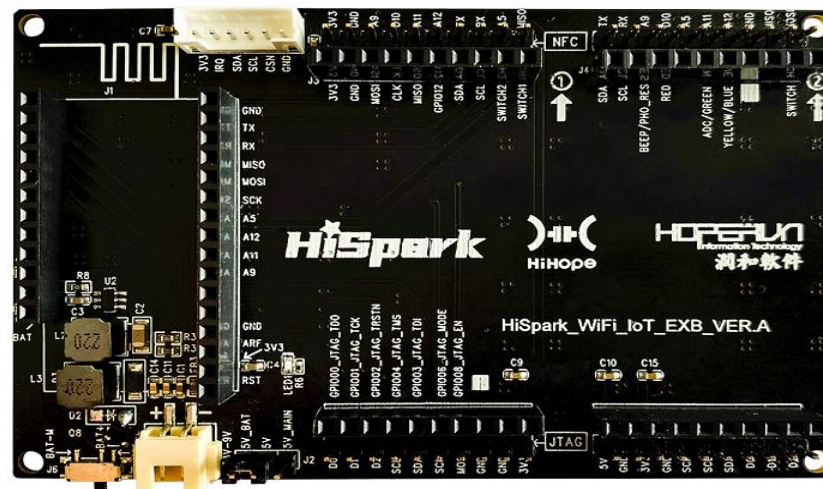
NFC板



3861_CH340G
主板



物联网底板



知识点2 【开发模式】

■ Linux compile server

- 编译open Harmony;
- 提供文件共享服务, 共享源码

■ Windows workstation

- 编辑源码、烧写程序,
- 通过USB/RS232连接

■ Hi3861 WLAN module

- 接收程序或指令, 运行程序或指令



知识点3 【课程内容介绍】

鸿蒙入门系列——Hi3861开发板基础课程 共分8讲

在线直播《鸿蒙入门系列——Hi3861开发板基础课程》大纲		
序号	名称	目标内容介绍
1	Hi3861开发环境搭建	1、课程介绍（开发语言C、开发板Hi3861介绍、开发方式、课程内容） 2、Linux服务器环境搭建 3、Window下开发环境搭建 4、HarmonyOS 源码下载及编译 5、使用HiBurn烧录固件到开发板 6、windows下串口终端测试效果
2	Hi3861上开发第一个Hello world程序	1、VS code中导入鸿蒙源码 2、编写第一个Hello world程序 3、完成HelloWorld程序编写、编译、烧写、运行 4、启动流程分析
3	Hi3861外设——智能红绿灯板开发	1、核心板上LED灯开发 2、交通灯功能实现 3、控制蜂鸣器发出声音 4、蜂鸣器实现播放音乐
4	Hi3861外设——环境监测板开发	1、AHT20温湿度传感器数据读取 2、MQ-2燃气传感器数据读取 3、OLED显示数据 4、综合案例讲解

知识点3 【课程内容介绍】

鸿蒙入门系列——Hi3861开发板基础课程 共分8讲

5	Hi3861上交互命令AT分析及实现	1、AT指令分类 2、通用AT指令 3、STA相关AT指令 4、TCP/IP相关AT指令 5、使用场景示例 6、系统AT指令源码分析 7、添加自定义AT命令
6	Hi3861上WiFi开发	1、AT指令开启STA功能 2、代码开启STA功能 3、AT指令开启AP功能 4、代码开启AP功能
7	Hi3861上网络编程	1、lwIP概述 2、lwIP功能特性 3、lwIP开发指引 4、lwIP示例代码 1) TCP服务端示例代码; 2) TCP客户端示例代码 3) UDP服务端示例代码 4) UDP客户端示例代码
8	Hi3861上物联网应用开发	1、MQTT协议 2、paho mqtt移植 3、搭建MQTT测试环境 4、测试MQTT

本节小结

本讲所学知识点有：

- 知识点1：开发板Hi3861介绍
- 知识点2：开发模式
- 知识点3：课程内容

第2节：环境搭建

- 知识点1：Linux服务器环境搭建
- 知识点2：Window下开发环境搭建



知识点1 【Linux服务器环境搭建】

- 步骤1：安装虚拟机软件（建议安装 VMWare）
- 步骤2：安装ubuntu20.04（python默认安装Python 3.8.5）
- 步骤3：root初始化密码：sudo passwd
- 步骤4：软件更新：sudo apt-get update
- 步骤5：安装vim：sudo apt install vim
- 步骤6：安装net-tools：sudo apt install net-tools
- 步骤7：安装ssh服务：sudo apt-get install openssh-server

启动服务：sudo service ssh start

- 接下来可参考官网进行操作：

https://device.harmonyos.com/cn/docs/start/introduce/oem_quickstart_3861_build-0000001054781998

知识点1 【Linux服务器环境搭建】

目录

Linux compile server 环境搭建

1. 安装 VirtualBox-6.1.16-140961-Win
2. 安装 Linux compile server 系统为 ubuntu-20.04 (默认安装即可)
3. 更新软件
4. 打开终端:showApplicatio -> terminal
5. root 初始化密码 (sudo passwd), 密码设置为 123456
6. 软件更新: sudo apt-get update
7. 安装 ssh 服务
8. 设置 Virtualbox 主机和虚拟机之间文件夹共享及双向拷贝, 安装增加功能, 设置启用卡 2, 然后 ...
9. 安装 net-tools
10. 下载 PuTTY 工具
11. PuTTY 连接
12. 将 Linux shell 改为 bash
13. 安装 Python 环境 (默认安装 3.8.2, 所以不需要安装)
14. 安装更新
15. 安装环境依赖(gcc, g++, make, zlib, libffi)请运行如下命令
16. 查看 python 版本,
17. 如果没有, 把在/usr/bin 中创建 python3.8 的软连接
18. 安装并升级 Python 包管理工具 (pip3)
19. 运行如下命令, 安装 python 模块 setuptools.
20. 安装 GUI menuconfig 工具 (Kconfiglib), 建议安装 Kconfiglib 13.2.0+版本
21. 安装 pycryptodome, 任选如下一种方式。
22. 安装 six
23. 安装 ecdsa
24. 安装 Scons
25. 安装 samba 服务
26. 安装 vim
27. 安装 gn (在 ubuntu 上下载保存, 下载地址:
28. 安装 ninja方法同上, 下载地址:
29. 安装 gcc_riscv32 (WLAN 模组类编译工具链 下载地址:
30. 获取全量代码 (下载地址:
31. 进行编译
32. 烧写
33. 测试

29、安装 gcc_riscv32 (WLAN 模组类编译工具链 下载地址:
https://repo.huaweicloud.com/harmonyos/compiler/gcc_riscv32/7.3.0/linux/gcc_riscv32-linux-7.3.0.tar.gz)

```
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export PATH=~/.gn:$PATH
export PATH=~/.ninja:$PATH
export PATH=~/.gcc_riscv32/bin:$PATH
gyz@gyz-VirtualBox: ~/Downloads$ a
```

1) 安装方法同上 gcc_riscv32 安装包至~/gcc_riscv32 路径下

```
tar -xvf gcc_riscv32-linux-7.3.0.tar.gz -C ~/
vim ~/.bashrc
export PATH=~/.gcc_riscv32/bin:$PATH
source ~/.bashrc
```

知识点1 【Linux服务器环境搭建】

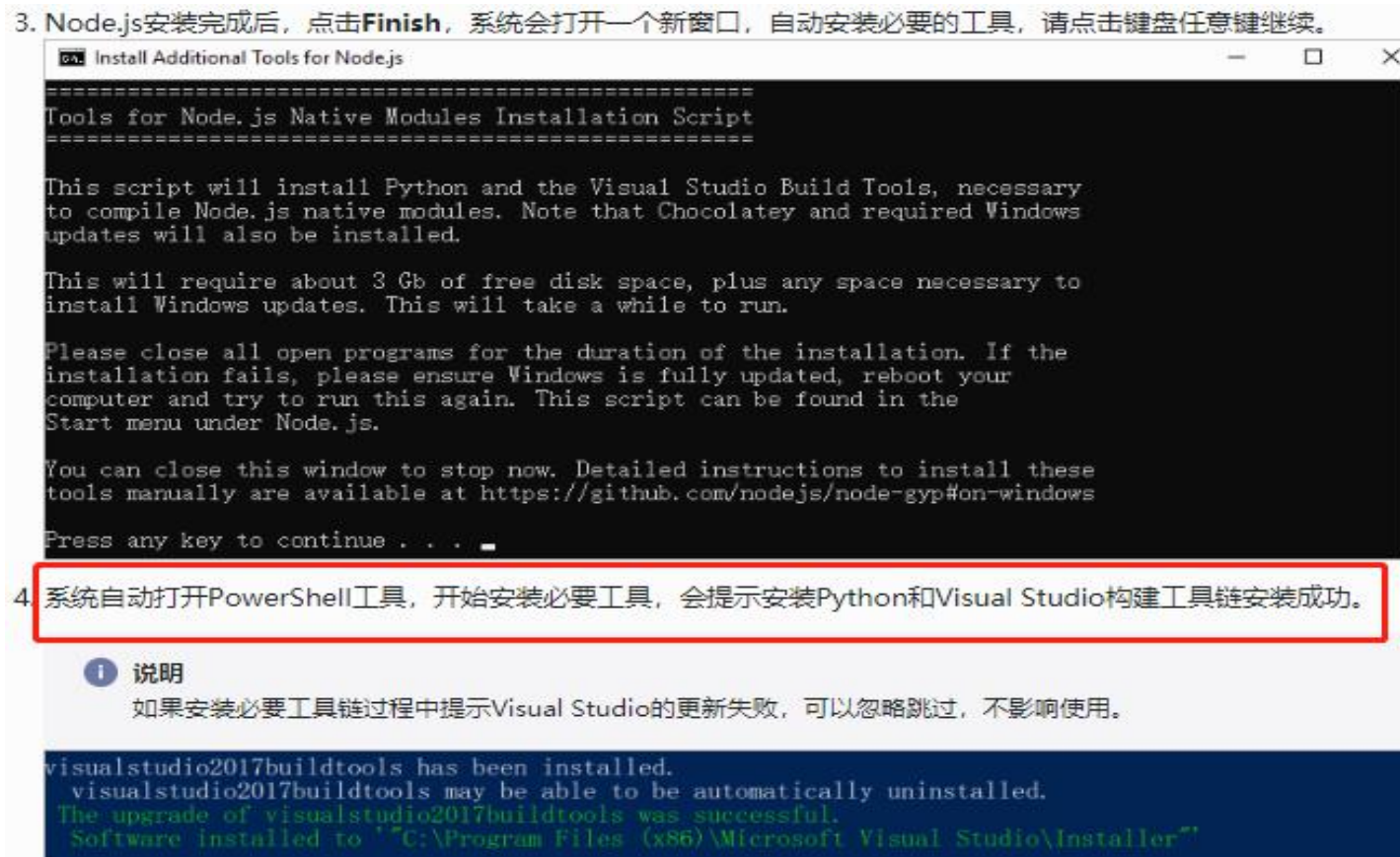
操作演示



知识点2 【Window下开发环境搭建】

请参考官网：https://device.harmonyos.com/cn/docs/ide/user-guides/tool_install-0000001050164976，安装分为如下几步（系统必须为Win10，原因安装过程中使用power shell）：

- 安装Visual Studio Code
- 安装Node.js
- 安装JDK
- 安装hpm
- 安装DevEco Device Tool插件
- 安装C/C++插件



知识点2 【Window下开发环境搭建】

操作演示



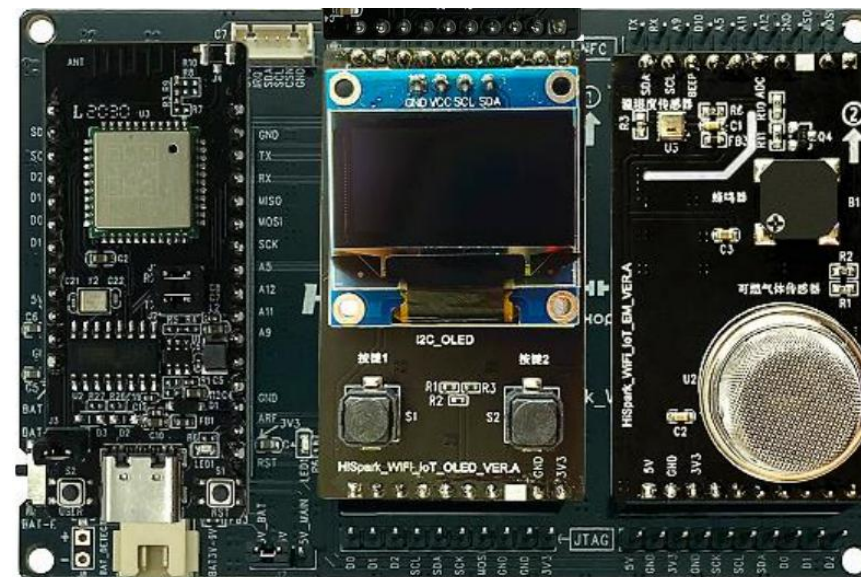
本节小结

本讲所学知识点有：

- 知识点1：Linux服务器环境搭建
- 知识点2：Window下开发环境搭建

第3节： HarmonyOS 源码下载及编译

- 知识点1： HarmonyOS 源码下载
- 知识点2： 编译



知识点1 【HarmonyOS 源码下载】

■ 方式1：全量代码获取

- <https://repo.huaweicloud.com/harmonyos/os/1.0/code-1.0.tar.gz>

■ 方式2：从代码仓库获取

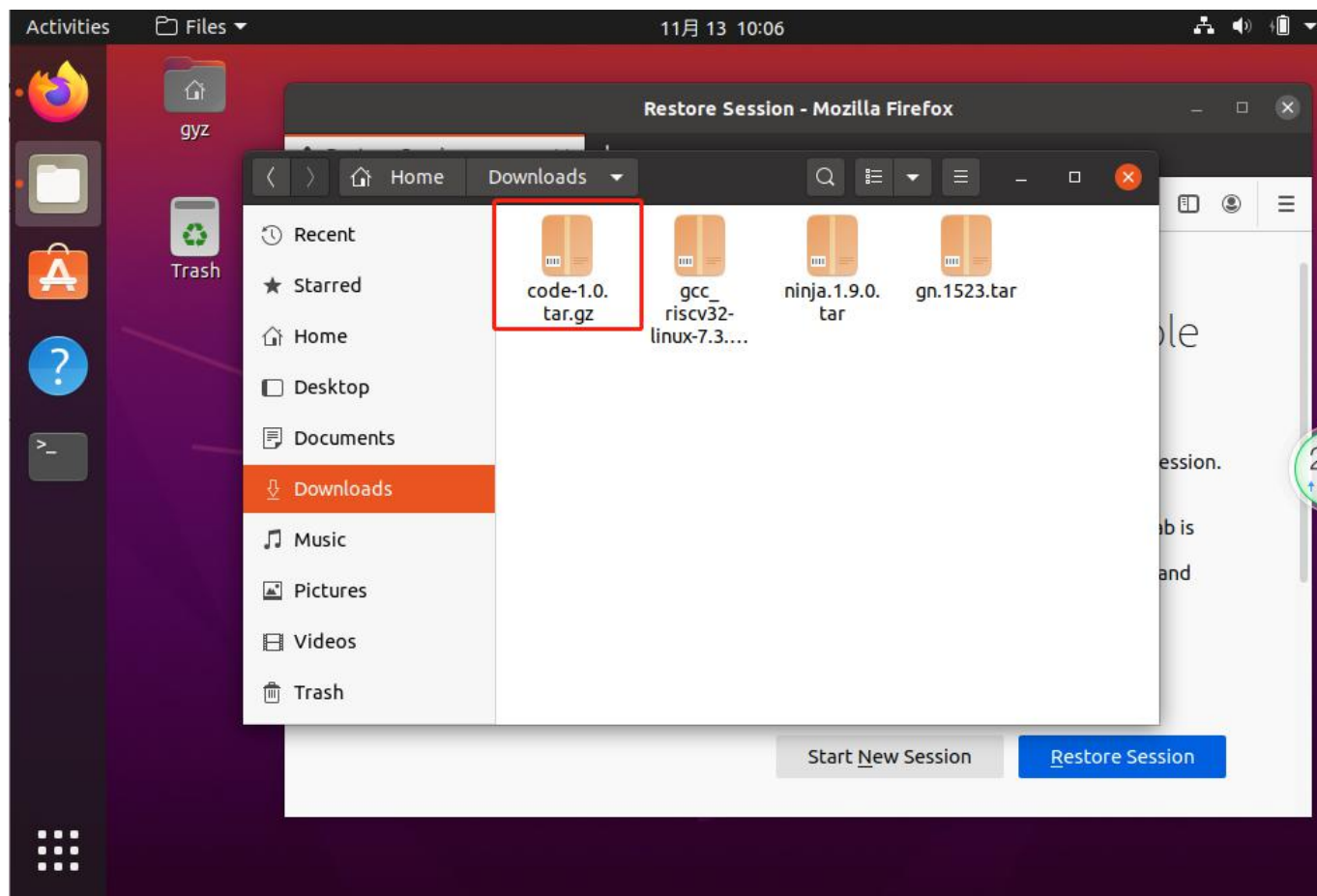
- 参考：Harmony OS 开发指南——源码下载和编译

https://gitee.com/hihopeorg/HarmonyOS-IoT-Application-Development/blob/master/01_envsetup/hos_source_code_download_and_compile.md

知识点1 【HarmonyOS 源码下载】

■ 方式1：全量代码获取

➤ <https://repo.huaweicloud.com/harmonyos/os/1.0/code-1.0.tar.gz>



知识点1 【HarmonyOS 源码下载】

- 方式2：从代码仓库获取，参考Harmony OS 开发指南——源码下载和编译（**华为官方文档无法实现会报错**），https://gitee.com/hihopeorg/HarmonyOS-IoT-Application-Development/blob/master/01_envsetup/hos_source_code_download_and_compile.md

➤ 配置 repo 工具

如果你的Linux系统上还没有配置repo命令，需要先下载并配置repo命令行工具：

```
mkdir ~/bin/
# sudo apt install curl # 如果没有curl命令需要先下载
curl https://gitee.com/oschina/repo/raw/fork_flow/repo-py3 > ~/bin/repo
chmod +x ~/bin/repo
echo 'export PATH=~/bin:$PATH' >> ~/.bashrc
source ~/.bashrc
```

➤ 下载Harmony OS源码

```
mkdir -p ~/harmonyos/openharmony && cd ~/harmonyos/openharmony
sudo apt install --no-install-recommends git python # repo工具本身是python脚本，它会调用git命令下载单个代码仓
# 开始前需要配置`user.name`和`user.email`，如果没有配置，使用如下命令进行配置：
# git config --global user.name "yourname"
# git config --global user.email "your-email-address"
repo init -u https://gitee.com/openharmony/manifest.git -b master --no-repo-verify
repo sync -c # 以后每天同步远程仓的修改，只需要执行这一条命令即可
```

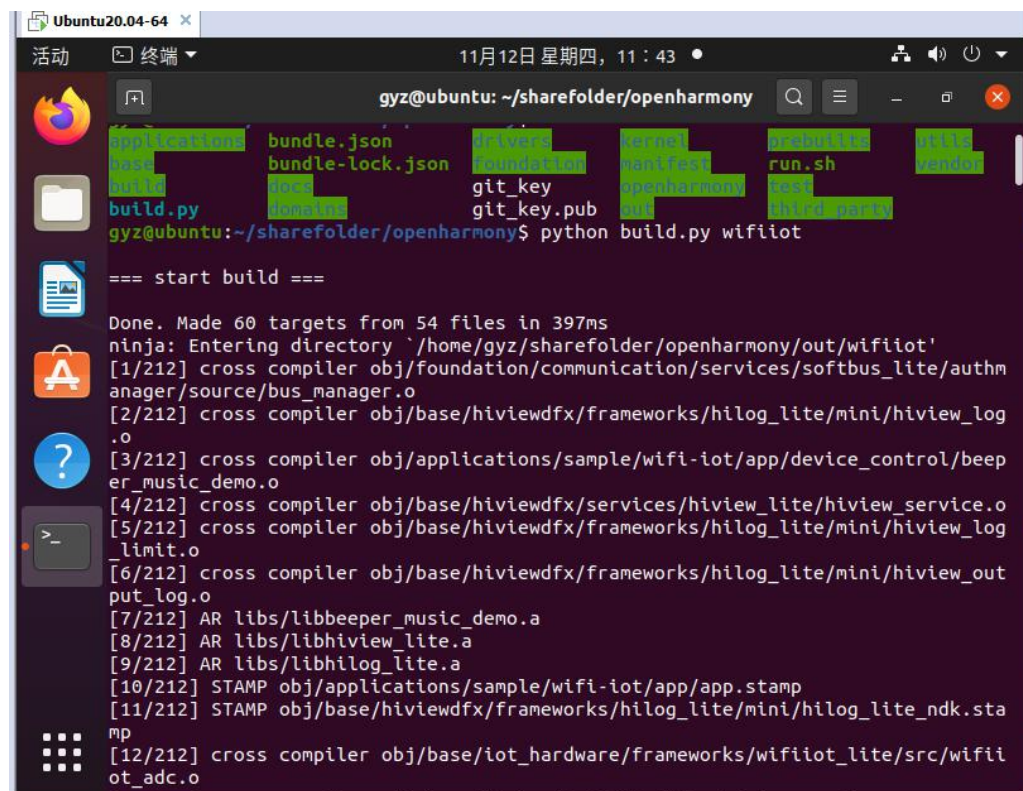
知识点1 【HarmonyOS 源码下载】

■ 源码结构

sharefolder (\\192.168.17.139) (Z:) > openharmony				
名称	修改日期	类型	大小	
applications	2020/10/31 11:10	文件夹		
base	2020/10/31 11:10	文件夹		
build	2020/10/31 11:10	文件夹		
developtools	2020/11/12 11:59	文件夹		
docs	2020/10/31 11:10	文件夹		
domains	2020/10/31 11:10	文件夹		
drivers	2020/10/31 11:10	文件夹		
foundation	2020/10/31 11:11	文件夹		
kernel	2020/10/31 11:11	文件夹		
manifest	2020/10/31 10:54	文件夹		
openharmony	2020/10/31 11:04	文件夹		
out	2020/11/12 11:42	文件夹		
prebuilts	2020/10/31 11:11	文件夹		
test	2020/10/31 11:11	文件夹		
third_party	2020/11/6 11:15	文件夹		
utils	2020/10/31 11:11	文件夹		
vendor	2020/10/31 11:11	文件夹		
build.py	2020/10/31 11:10	Python File	2 KB	
bundle.json	2020/11/10 9:36	JSON 文件	1 KB	
bundle-lock.json	2020/11/3 15:48	JSON 文件	1 KB	
git_key	2020/10/31 10:42	文件	3 KB	
git_key.pub	2020/10/31 10:42	PUB 文件	1 KB	

知识点2【编译】

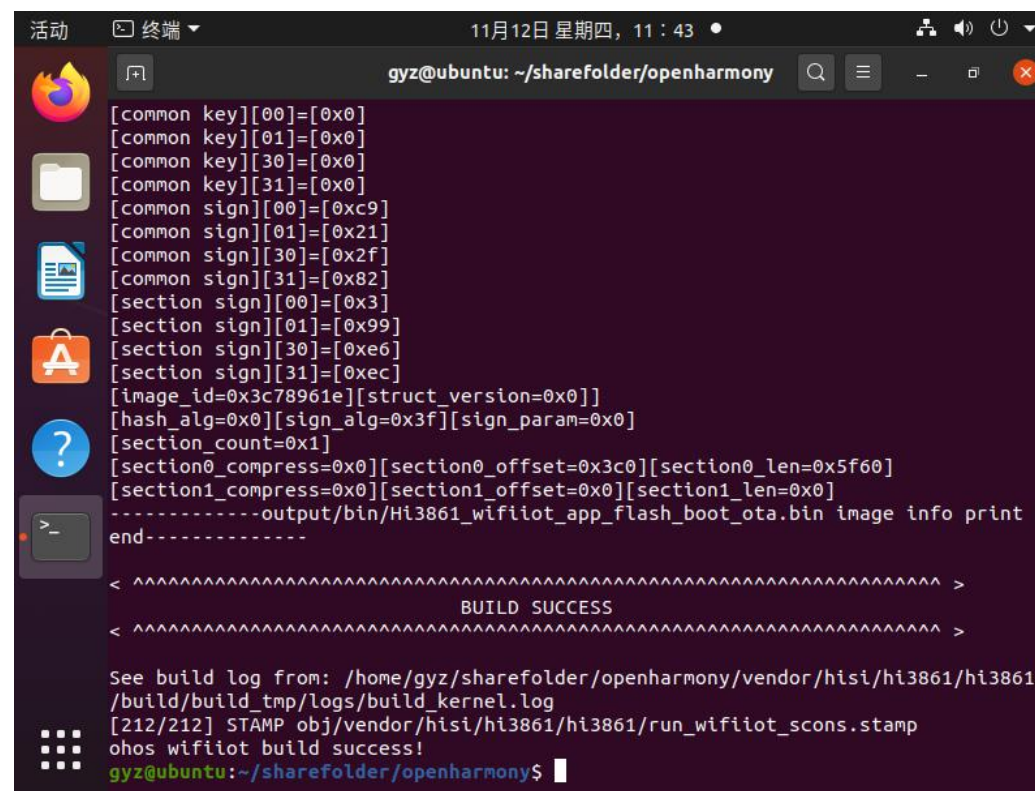
- 进入源码根目录执行命令：
 - python build.py wifiiot



```
gyz@ubuntu: ~/sharefolder/openharmony
python build.py wifiiot

=== start build ===

Done. Made 60 targets from 54 files in 397ms
ninja: Entering directory '/home/gyz/sharefolder/openharmony/out/wifiiot'
[1/212] cross compiler obj/foundation/communication/services/softbus_lite/authmanager/source/bus_manager.o
[2/212] cross compiler obj/base/hiviewdfx/frameworks/hilog_lite/mini/hiview_log.o
[3/212] cross compiler obj/applications/sample/wifi-iot/app/device_control/beeper_music_demo.o
[4/212] cross compiler obj/base/hiviewdfx/services/hiview_lite/hiview_service.o
[5/212] cross compiler obj/base/hiviewdfx/frameworks/hilog_lite/mini/hiview_log_limit.o
[6/212] cross compiler obj/base/hiviewdfx/frameworks/hilog_lite/mini/hiview_output_log.o
[7/212] AR libs/libbeeper_music_demo.a
[8/212] AR libs/libhiview_lite.a
[9/212] AR libs/libhilog_lite.a
[10/212] STAMP obj/applications/sample/wifi-iot/app/app.stamp
[11/212] STAMP obj/base/hiviewdfx/frameworks/hilog_lite/mini/hilog_lite_ndk.stamp
[12/212] cross compiler obj/base/iot_hardware/frameworks/wifiiot_lite/src/wifiiot_adc.o
```



```
gyz@ubuntu: ~/sharefolder/openharmony

[common key][00]=[0x0]
[common key][01]=[0x0]
[common key][30]=[0x0]
[common key][31]=[0x0]
[common sign][00]=[0xc9]
[common sign][01]=[0x21]
[common sign][30]=[0x2f]
[common sign][31]=[0x82]
[section sign][00]=[0x3]
[section sign][01]=[0x99]
[section sign][30]=[0xe6]
[section sign][31]=[0xec]
[image_id=0x3c78961e][struct_version=0x0]]
[hash_alg=0x0][sign_alg=0x3f][sign_param=0x0]
[section_count=0x1]
[section0_compress=0x0][section0_offset=0x3c0][section0_len=0x5f60]
[section1_compress=0x0][section1_offset=0x0][section1_len=0x0]
-----output/bin/Hi3861_wifiiot_app_flash_boot_ota.bin image info print
end-----

< ~~~~~
BUILD SUCCESS
~~~~~ >

See build log from: /home/gyz/sharefolder/openharmony/vendor/hisi/hi3861/hi3861/build/build_tmp/logs/build_kernel.log
[212/212] STAMP obj/vendor/hisi/hi3861/hi3861/run_wifiiot_scons.stamp
ohos wifiiot build success!
gyz@ubuntu: ~/sharefolder/openharmony$
```

知识点2【编译】

操作演示



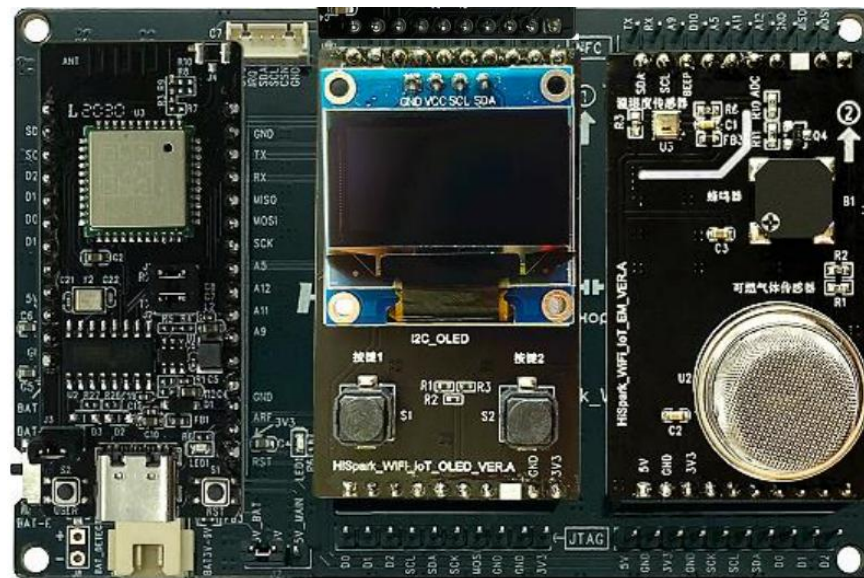
本节小结

本讲所学知识点有：

- 知识点1：HarmonyOS 源码下载
- 知识点2：编译

第4节：烧写固件及测试

- 知识点1：烧写固件
- 知识点2：测试

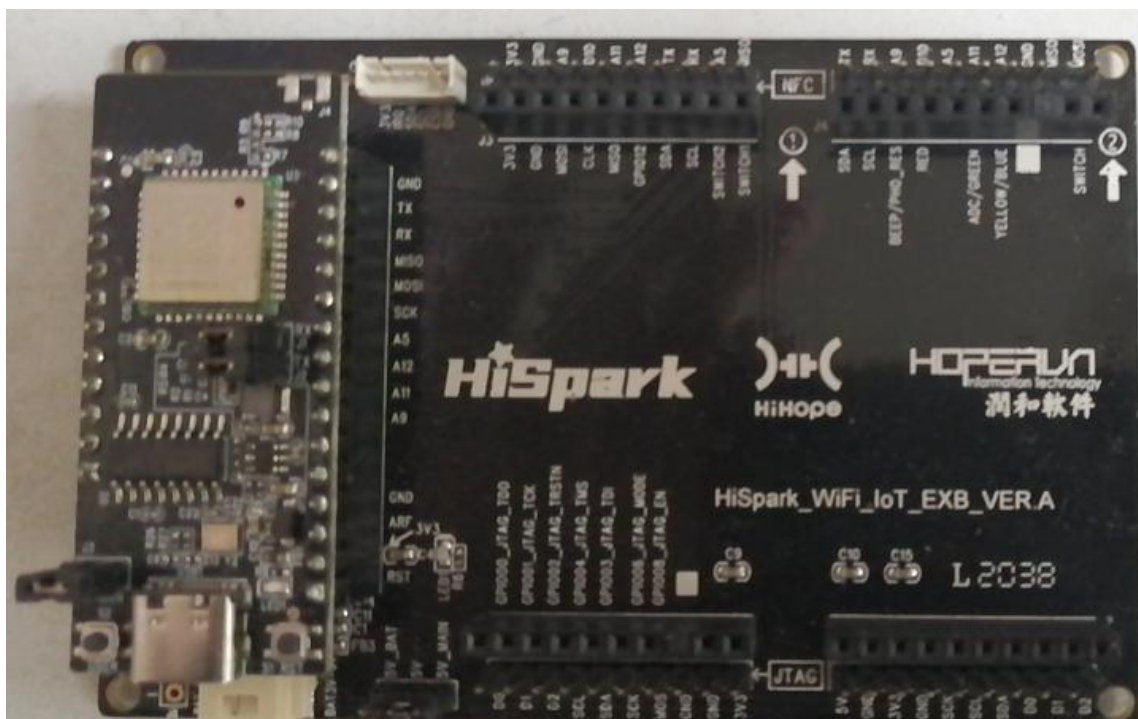


知识点1 【烧写固件】

- 步骤1：安装核心板到底板上
- 步骤2：下载HiBurn
- 步骤3：复制固件 Hi3861_wifiiot_app_allinone.bin到windows下
- 步骤4：使用HiBurn烧写固件

知识点1 【烧写固件】

■ 步骤1：安装核心板到底板上



知识点1 【烧写固件】

■ 步骤2：下载HiBurn，

➤ 解压DevEcoDeviceTool-1.0.1.vsix

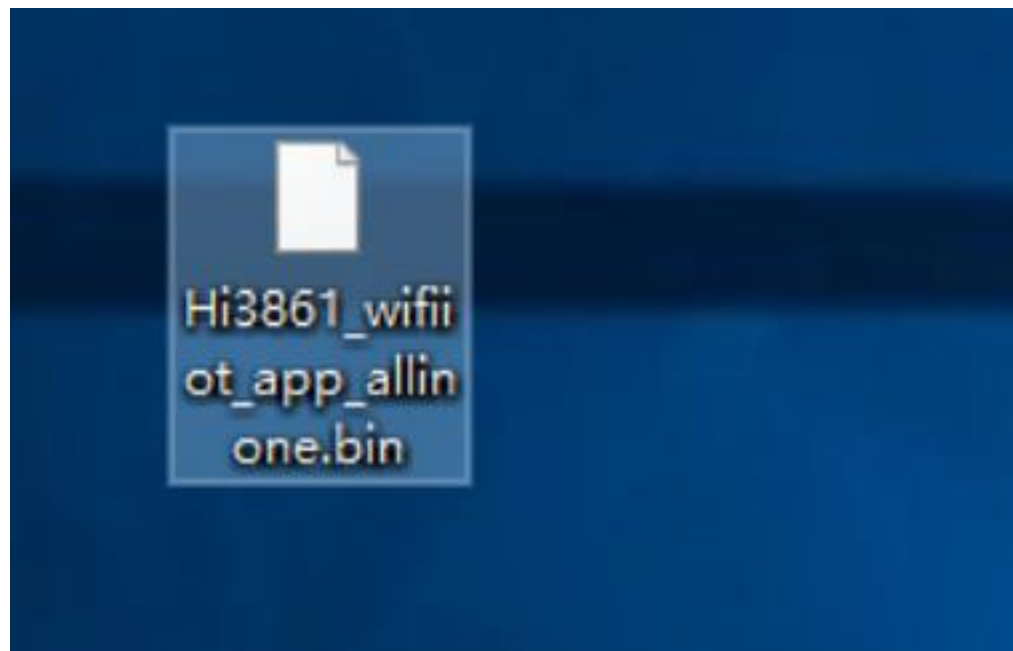
下载 > devicetool-device-1.0.0.1		
名称	修改日期	类型
DevEcoDeviceTool-1.0.1	2020/10/27 11:03	文件夹
DevEcoDeviceTool-1.0.1.vsix	2020/10/14 14:24	VSIX 文件

➤ 在extension\deveco\tools子目录中找到一个文件名为 HiBurn.exe 的文件

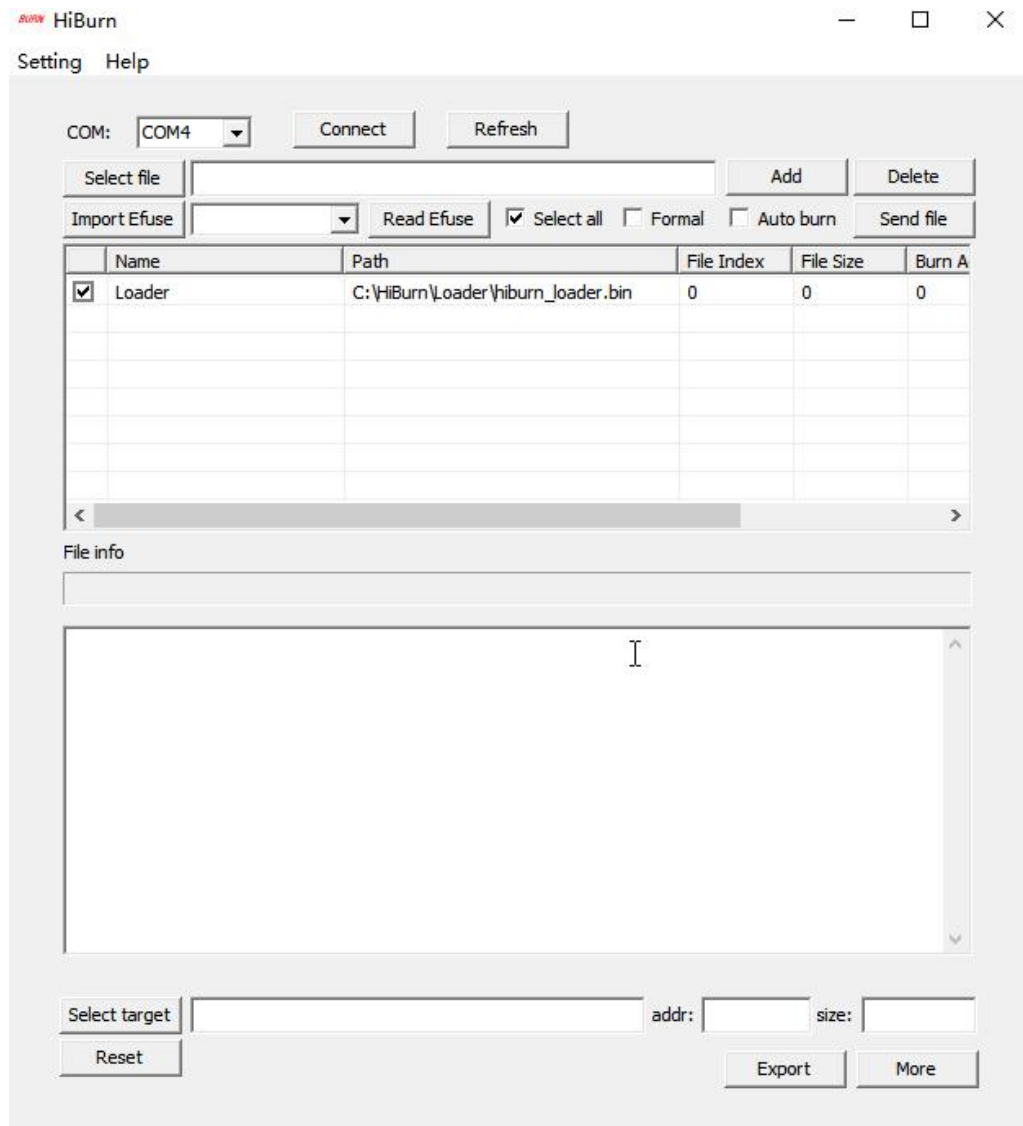
下载 > devicetool-device-1.0.0.1 > DevEcoDeviceTool-1.0.1 > extension > deveco > tools			
名称	修改日期	类型	大小
HiBurn.exe	2020/10/14 14:16	应用程序	3,541 KB

知识点1 【烧写固件】

- 步骤3: 复制固件 Hi3861_wifiiot_app_allinone.bin到windows下

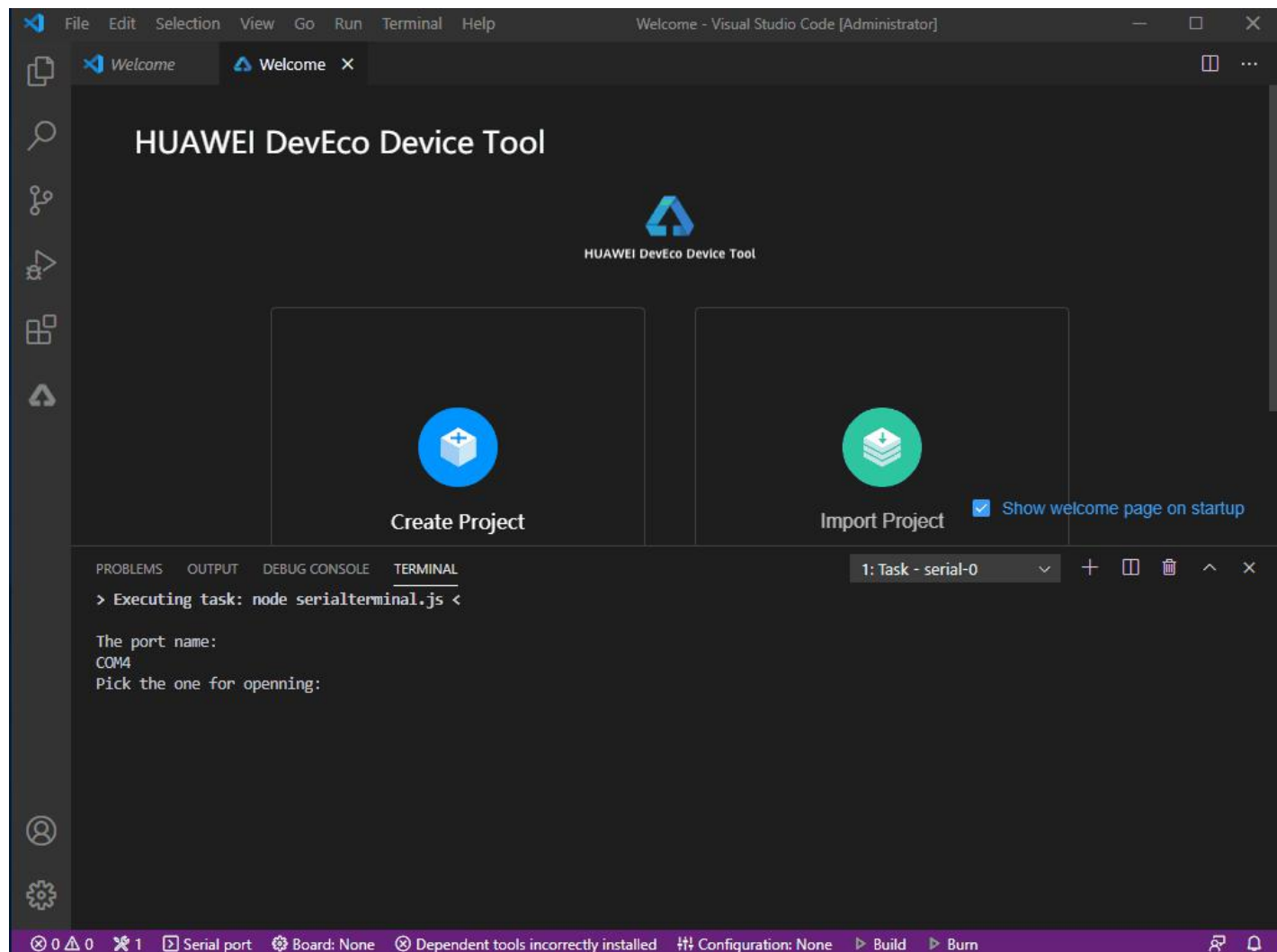


- 数据线连接Hi3861与电脑
- 点击Refresh
- 点击“ Select File ”选择固件
- 选择“ Auto burn ”
- 点击“ connect ”
- Hi3861按“ RST” 按钮进行复位
- 最后断开连接



知识点2 【测试】

- 步骤1：打开VS Code
- 步骤2：点击最下方 Serial port
- 步骤3：设备串口属性
- 步骤4：Hi3861复位
- 步骤5：输入AT命令测试



知识点2 【测试】

操作演示



本节小结

本讲所学知识点有：

- 知识点1：烧写固件
- 知识点2：测试

本章总结

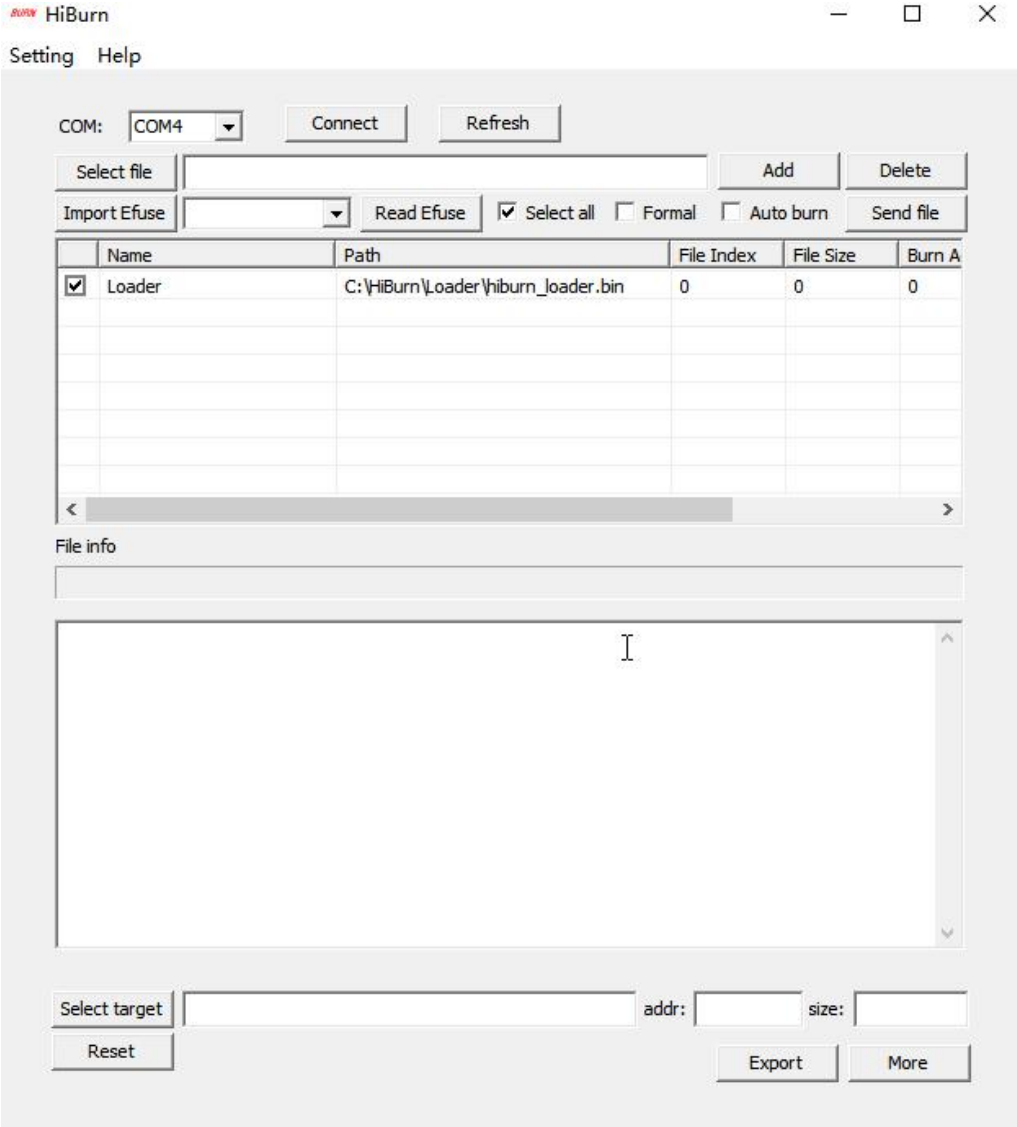
本章所学内容有：

- 第1节：课程介绍
- 第2节：环境搭建
- 第3节：HarmonyOS 源码下载及编译
- 第4节：烧写固件及测试

100



挑战任务：借助PPT和视频完成本章内容



THANKS

更多学习视频，关注宅客学院.....

