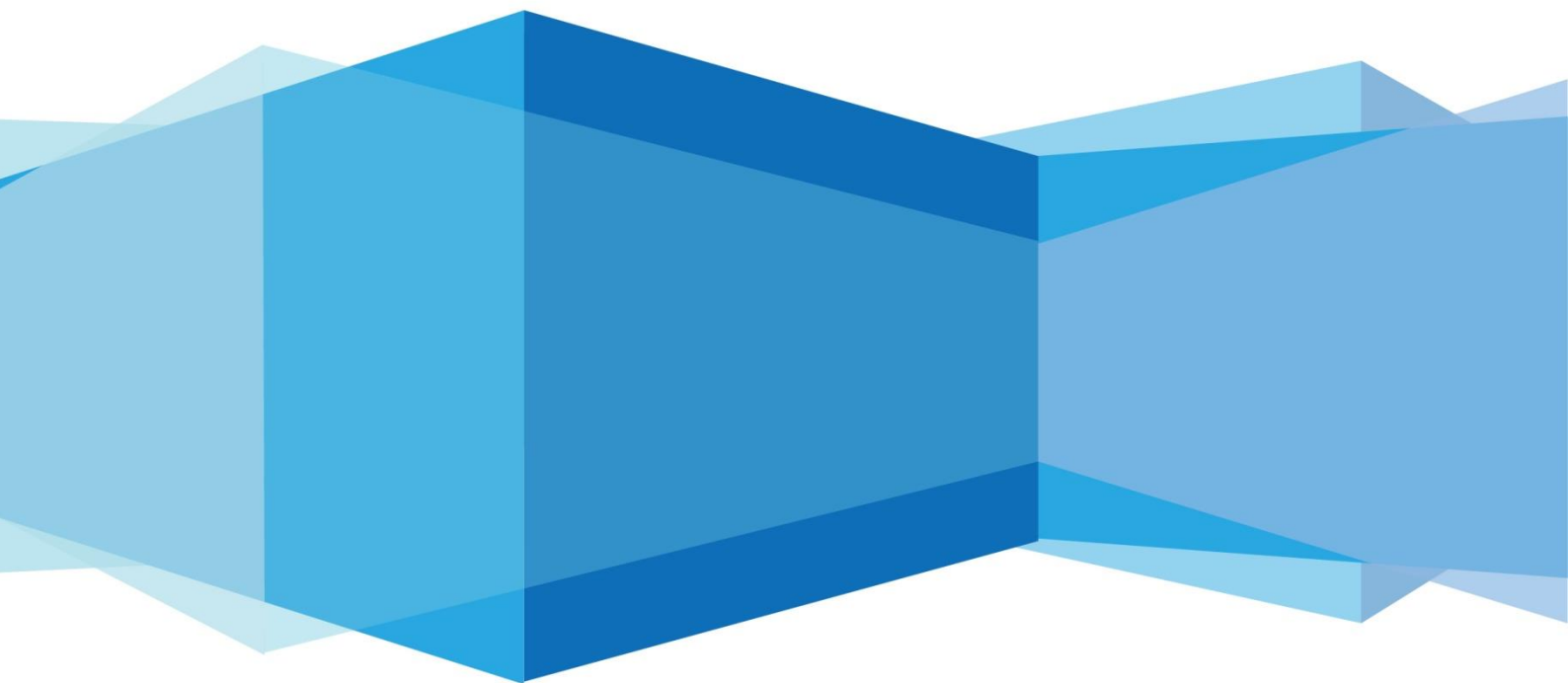


OpenCPU 开发环境搭建说明

版本：Rev1.0

日期：2019-06-14



法律声明

若接收浙江利尔达物联网技术有限公司（以下称为“利尔达”）的此份文档，即表示您已经同意以下条款。若不同意以下条款，请停止使用本文档。

本文档版权所有浙江利尔达物联网技术有限公司，保留任何未在本文档中明示授予的权利。文档中涉及利尔达的专有信息。未经利尔达事先书面许可，任何单位和个人不得复制、传递、分发、使用和泄漏该文档以及该文档包含的任何图片、表格、数据及其他信息。

本产品符合有关环境保护和人身安全方面的设计要求，产品的存放、使用和弃置应遵照产品手册、相关合同或者相关法律、法规的要求进行。

本公司保留在不预先通知的情况下，对此手册中描述的产品进行修改和改进的权利；同时保留随时修订或收回本手册的权利。

文件修订历史

版本	修订日期	修订日志
1.0	2019-05-06	新建文档

Lierda Science & Technology Group Co., Ltd

适用模块型号

序号	模块型号	模块简介
1	NB86-G	全频段版本，20×16×2.2（mm）
2	NB86-G 宽压型	全频段版本，20×16×2.2（mm）

安全须知

用户有责任遵循其他国家关于无线通信模块及设备的相关规定和具体的使用环境法规。通过遵循以下安全原则，可确保个人安全并有助于保护产品和工作环境免遭潜在损坏。我司不承担因客户未能遵循这些规定导致的相关损失。



道路行驶安全第一！当您开车时，请勿使用手持移动终端设备，除非其有免提功能。请停车，再打电话！



登机前请关闭移动终端设备。移动终端的无线功能在飞机上禁止开启用以防止对飞机通讯系统的干扰。忽略该提示项可能会导致飞行安全，甚至触犯法律。



当在医院或健康看护场所，注意是否有移动终端设备使用限制。RF 干扰会导致医疗设备运行失常，因此可能需要关闭移动终端设备。



移动终端设备并不保障任何情况下都能进行有效连接，例如在移动终端设备没有花费或 SIM 无效。当您在紧急情况下遇见以上情况，请记住使用紧急呼叫，同时保证您的设备开机并且处于信号强度足够的区域。



您的移动终端设备在开机时会接收和发射射频信号，当靠近电视，收音机电脑或者其它电子设备时都会产生射频干扰。



请将移动终端设备远离易燃气体。当您靠近加油站，油库，化工厂或爆炸作业场所，请关闭移动终端设备。在任何有潜在爆炸危险场所操作电子设备都有安全隐患。

目 录

法律声明.....	2
文件修订历史.....	3
适用模块型号.....	4
安全须知.....	5
目 录.....	6
1. 编译器 -gcc.....	7
1.1. B300SP2&B300SP5.....	7
1.2. B500SP1	8
2. Python 2.7.....	9
2.1. Python 2.7 安装	9
2.2. pywin32 安装.....	10
2.3. 安装 Python PyYAML 库	11
2.4. Scons 安装	12
3. IDE – Eclipse 安装.....	12
4. 参考资料.....	13
5. 相关文档及术语缩写.....	14

1. 编译器 -gcc

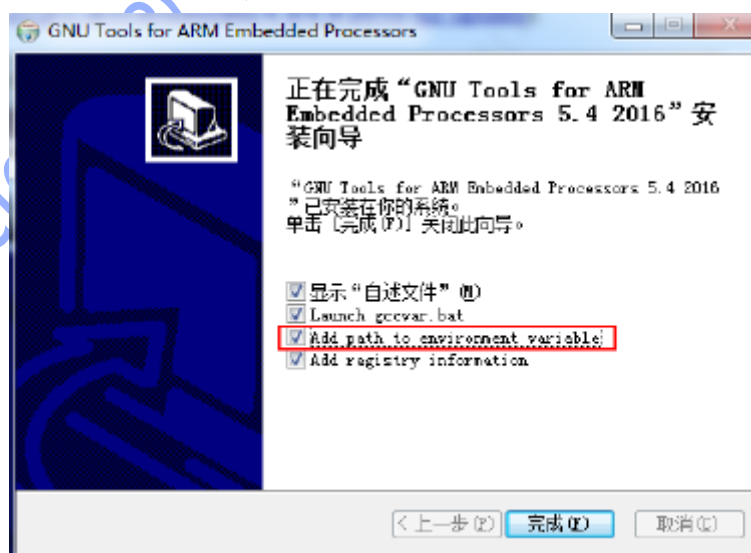
B500 编译调用的 GCC 编译器版本与 B300SP2&B300SP5 不一样，若要支持 B300SP2&B300SP5 和 B500SP1 则需要安装两个不同版本的 GCC 编译器。具体的安装方法如下：

1.1. B300SP2&B300SP5

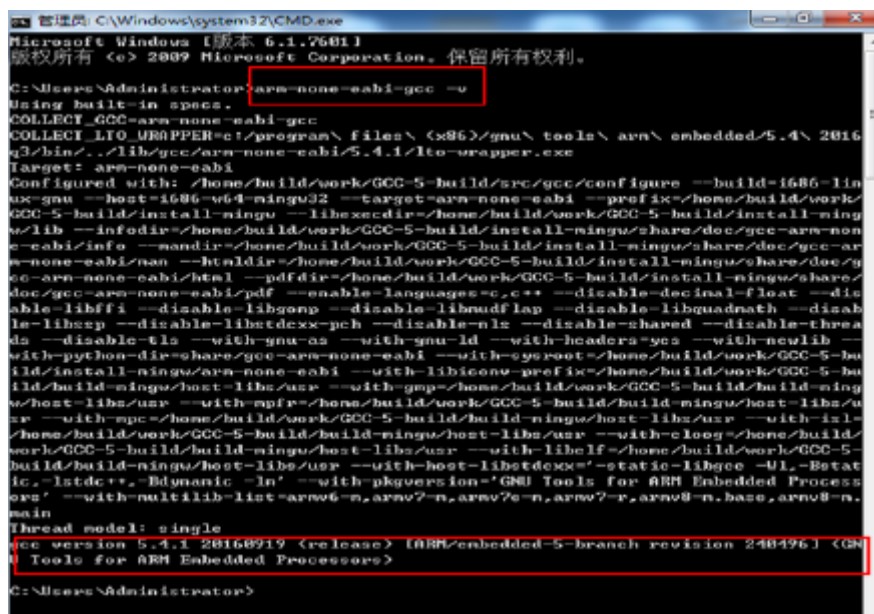
下载 [GCC 编译器安装文件](#) 进入软件下载界面下载所需的软件安装包如图



点击下载好的软件进行安装，需要注意的是安装完成后，确保选择了“添加环境变量的路径”选项如图



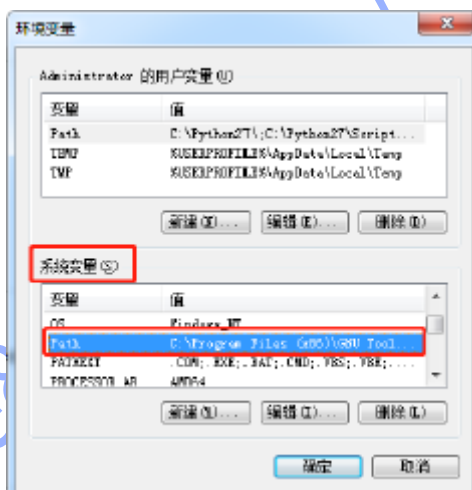
进入 windows 命令行输入 `arm-none-eabi-gcc -v` 检查 gcc 版本，判断 gccs 是否安装成功，如图



```
C:\Windows\system32\CMD.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation. 保留所有权利。

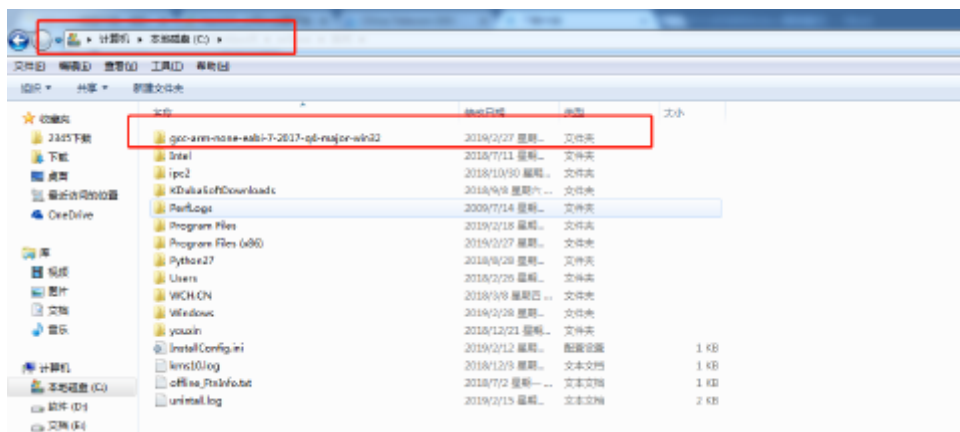
C:\Users\Administrator>arm-none-eabi-gcc -v
Using built-in specs.
COLLECT_GCC=arm-none-eabi-gcc
COLLECT_LTO_WRAPPER=c:/program files (x86)/gnu\ tools\ arm\ embedded\ 5.4\ 20160919\ bin\ ./lib/gcc/arm-none-eabi/5.4.1/lto-wrapper.exe
Target: arm-none-eabi
Configured with: /home/build/work/GCC-5-build/src/gcc/configure --build=i686-linux-gnu --host=i686-w64-mingw32 --target=arm-none-eabi --prefix=/home/build/work/GCC-5-build/install-mingw --libexecdir=/home/build/work/GCC-5-build/install-mingw/share/doc/gcc-arm-none-eabi/info --mandir=/home/build/work/GCC-5-build/install-mingw/share/doc/gcc-arm-none-eabi/html --pdfdir=/home/build/work/GCC-5-build/install-mingw/share/doc/gcc-arm-none-eabi/pdf --enable-languages=c,c++ --disable-decimal-float --disable-lhsffi --disable-libgomp --disable-libmudflap --disable-libquadmath --disable-libssp --disable-libstdc++-pch --disable-nls --disable-shared --disable-threads --disable-tls --with-gnu-as --with-gnu-ld --with-headers=yes --with-newlib --with-python-die=share/gcc-arm-none-eabi --with-sysroot=/home/build/work/GCC-5-build/install-mingw/arm-none-eabi --with-libiconv-prefix=/home/build/work/GCC-5-build/build-mingw/host-libc/usr --with-gmp=/home/build/work/GCC-5-build/build-mingw/host-libc/usr --with-mpfr=/home/build/work/GCC-5-build/build-mingw/host-libc/usr --with-mpc=/home/build/work/GCC-5-build/build-mingw/host-libc/usr --with-isl=/home/build/work/GCC-5-build/build-mingw/host-libc/usr --with-cloog=/home/build/work/GCC-5-build/build-mingw/host-libc/usr --with-libelf=/home/build/work/GCC-5-build/build-mingw/host-libc/usr --with-host-libstdc++=static-libgcc -U-L-Bstatic -lstdc++ -Bdynamic -ln' --with-pkgversion='GNU Tools for ARM Embedded Processors' --with-multilib-list=armv6-m,armv7-m,armv7e-m,armv7-r,armv8-m.base,armv8-m.main
Thread model: single
gcc version 5.4.1 20160919 (release) [ARM/embedded-5-branch revision 248496] (GNU Tools for ARM Embedded Processors)
```

如果查不到版本信息，检查环境变量是否添加成功如图。如果环境变量里查不到 gcc 的安装路径，请根据自己 gcc 安装的路径添加环境变量。

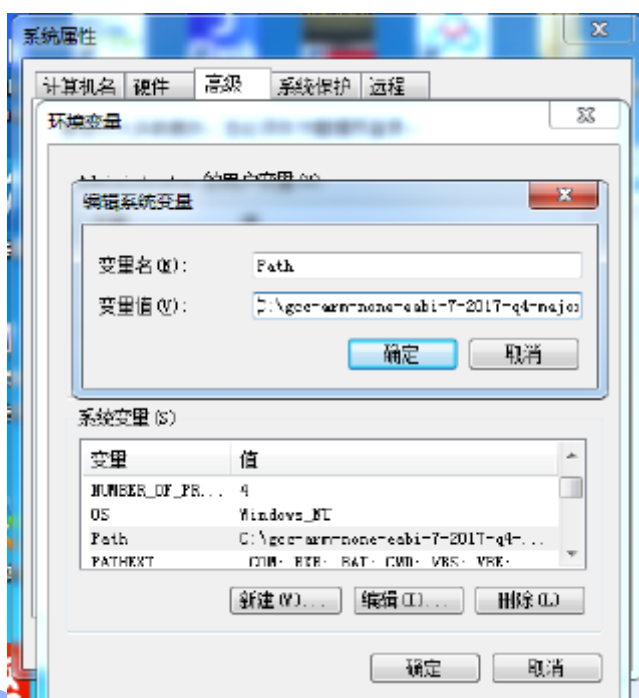


1.2. B500SP1

下载 [GCC 编译器安装文件](#) 并安装，只需将下载的 zip 文件解压放到 C 盘的根目录即可，如图



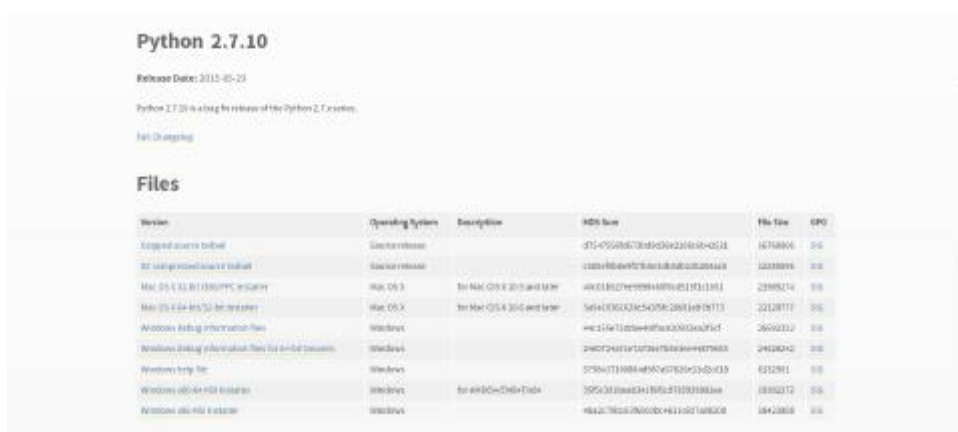
然后，添加环境变量 `C:\gcc-arm-none-eabi-7-2017-q4-major-win32\bin` 如图



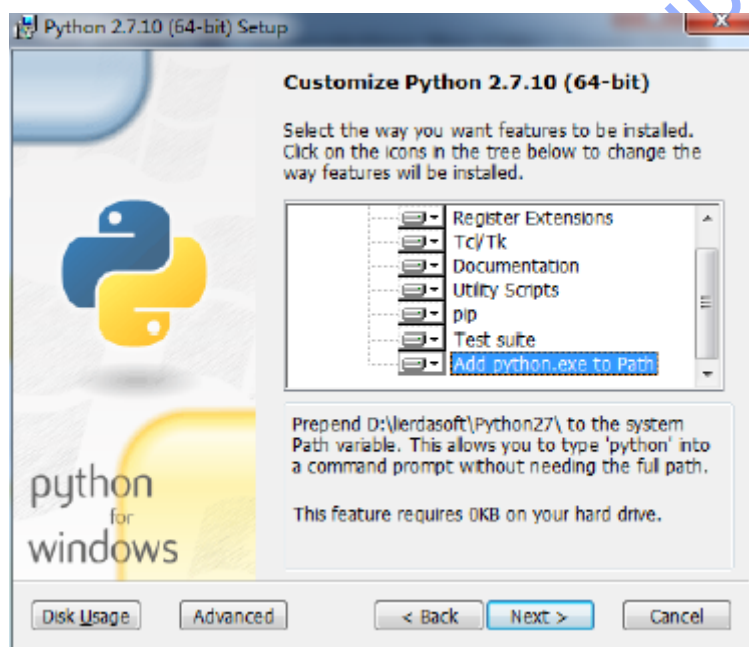
2. Python 2.7

2.1. Python 2.7 安装

Scons 是一个用于基于 Python2.7 的软件构建的工具。确保在安装 Scons 之前安装 Python 2.7 如图



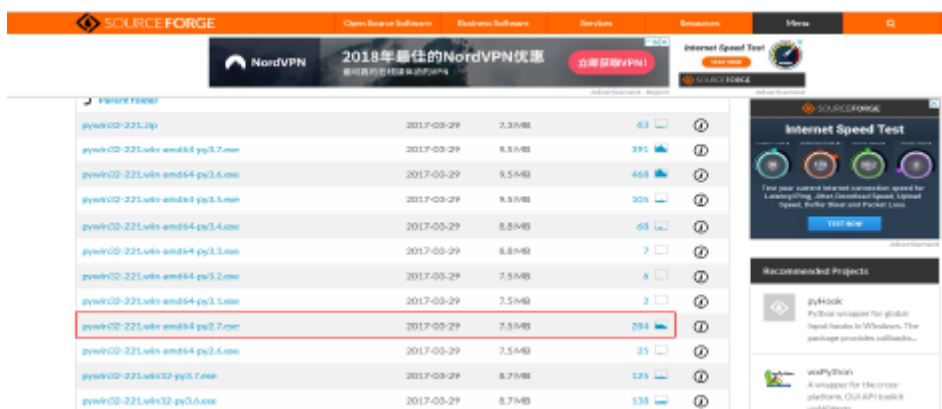
点击下载好的软件进行安装，需要注意的是在安装过程中，确保选择了“添加环境变量的路径”选项如图。点击下一步完成安装。



2.2. pywin32 安装

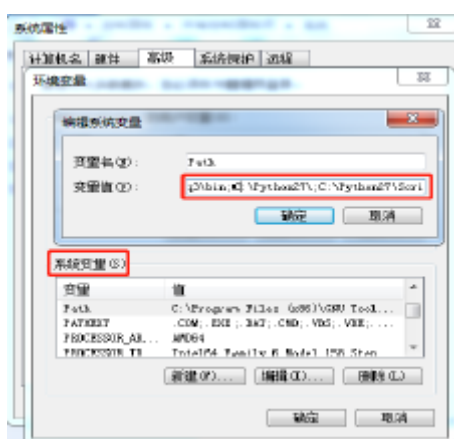
下载对应的 [pywin32](#)

注意：下载的版本和 Python 2.7 相对应，如图



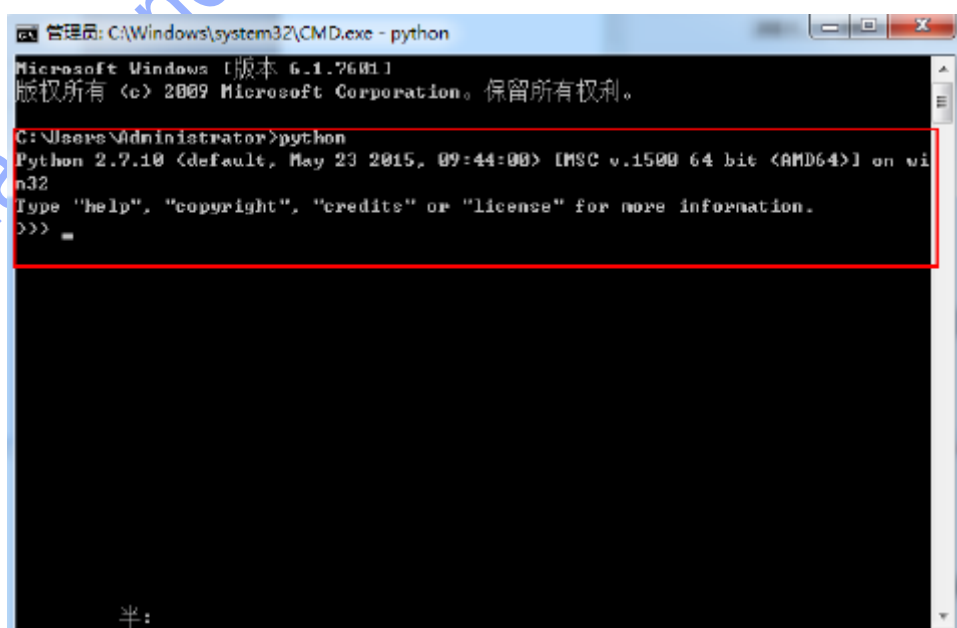
点击下载好的.exe 文件安装。

注意：检查系统环境变量中是否添加了 C:\Python27\;C:\Python27\Scripts，如果没有添加，一定要添加上去如图



2.3. 安装 Python PyYAML 库

进入 windows 命令行输入 python 检查 Python 是否安装成功，如图



注意：若果没有安装成功请检查环境变量是否添加正确

安装 PyYAML 库，在命令行输入 `pip install pyyaml`，如图

2.4. Scons 安装

下载 [Scons](#)

注意：Scon 应该安装在 Python 的脚本文件夹(.\\Python 27\\Script)中

3. IDE – Eclipse 安装

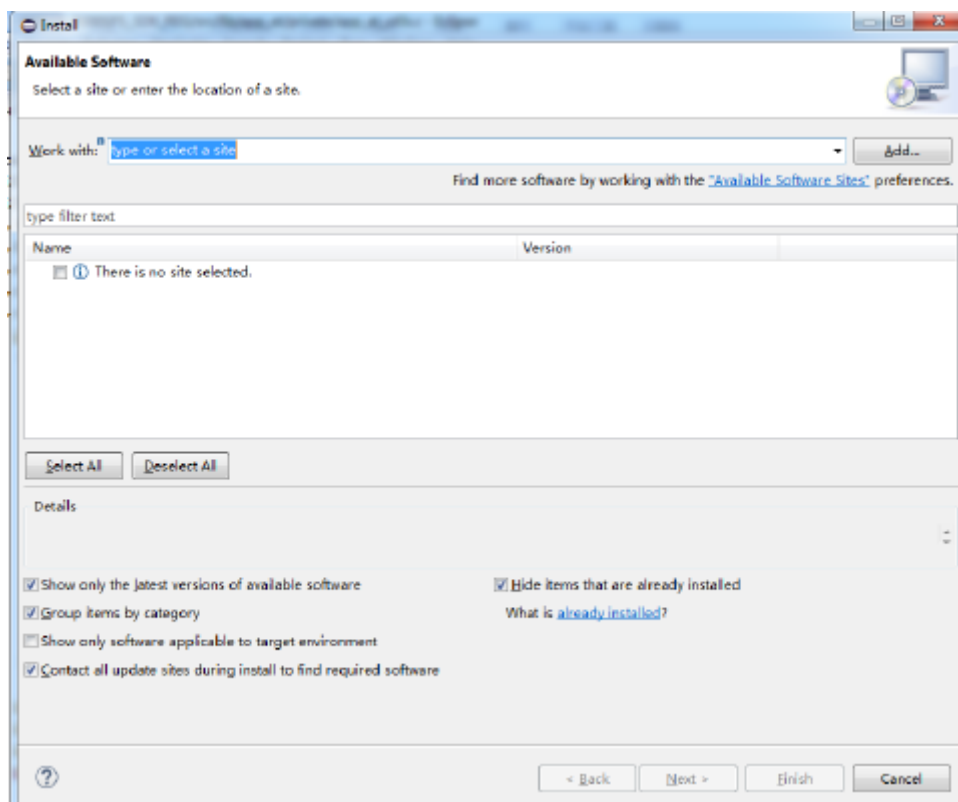
注意：检查 Java 是否安装，进入命令行，输入 `java -version`

若没安装请先安装 [Java](#)，如图

下载 [Eclipse](#)

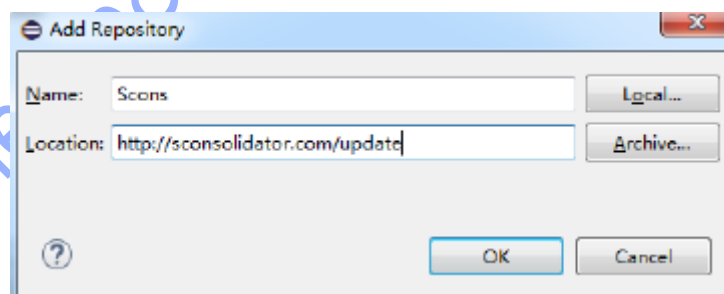
Eclipse 插件安装（Scons）

打开 Eclipse，按一下操作步骤进入 Scons 安装界面，如图



1. 点击选项栏：Help
2. 点击 help 下子选项栏：Install New Software...
3. 点击 Work with 后的 Add
4. 填入下列信息 Scons <http://sconsolidator.com/update>

如图



5. 从插件列表中为 scons 选择 Eclipse 插件
6. 单击 Next，并按照说明执行

至此 OpenCPU 开发环境搭建完成，工程导入及编译见下一小节。

4. 参考资料

技术论坛	OpenCPU 资料	NB86 EVK 资料
------	------------	-------------

[物联网开发者社区](#)[OpenCPU 基本资料集](#)[NB86 EVK 基本资料集](#)

5. 相关文档及术语缩写

以下相关文档提供了文档的名称，版本请以最新发布的为准。

表格 1 相关文档

序号	文档名称	注释
[1]	NB86-G硬件应用手册	
[2]	Lierda NB Module V150_AT CommandSet_B300SP5	
[3]	Lierda NB86-EVK测试终端固件烧写教程	
[4]	Lierda NB-IoT模组API使用文档	
[5]	Lierda NB-IoT模组DEMO说明文档	
[6]	Lierda NB-IoT模组V150 OpenCPU开发环境搭建指南	