参考：http://blog.sina.com.cn/s/blog\_86d540e401019yie.html  
  
做这一步之前，首先确定你的本机上有交叉编译环境

目标板（即开发板）和主机可以ping通。

### **1》下载需要的源码** 下载zlib: zlib-1.2.3.tar.gz       http://www.zlib.net 下载ssl : openssl-0.9.8d.tar.gz    http://www.openssl.org/source/ 下载ssh : openssh-4.6p1.tar.gz     http://mirror.aarnet.edu.au/pub/OpenBSD/OpenSSH/portable/ **2》编译** /home/arm下新建目录sshwork，并且将源码复制到该目录下 mkdir /home/arm/sshwork cp zlib-1.2.3.tar.gz openssl-0.9.8d.tar.gz openssh-4.6p1.tar.gz /home/arm/sshwork /home/arm/sshwork下新建目录lib，用来保存生成的库文件。 mkdir /home/arm/sshwork/lib **1、编译zlib** tar zxvf zlib-1.2.3.tar.gz -C . cd zlib-1.2.3/ ./configure -prefix=/home/arm/sshwork/lib/zlib-1.2.3 修改Makefile : CC=gcc 改为: CC=arm-linux-gcc LDSHARED= gcc 改为:LDSHARED=arm-linux-gcc CPP= gcc - E  改为:CPP=arm-linux-gcc - E AR= ar rc     改为:AR=arm-linux-ar rc 开始编译: make make install **2、编译openssl** tar zxvf openssl-0.9.8d.tar.gz ./Configure --prefix=/home/arm/sshwork/lib/openssl-0.9.8d os/compiler:arm-linux-gcc make make install **3、编译openssh** tar zxvf openssh-4.6p1.tar.gz cd openssh-4.6p1/ ./configure -host=arm-linux -with-libs -with-zlib=/home/arm/sshwork/lib/zlib-1.2.3 -with-ssl-dir=/home/arm/sshwork/lib/openssl-0.9.8d -disable-etc-default-login CC=arm-linux-gcc AR=arm-linux-ar make ##不要make install **3》安装** 确保目标板上有以下目录，如果没有，则新建（目录层次很重要，不要改变） /usr/sbin /usr/local/bin /usr/local/libexec /usr/local/etc/ **1、在 openssh-4.6p1 目录下创建 usr bin 目录，文件临时拷贝到这里，然后统一打包到开发板** cd openssh-4.6p1 mkdir -p bin mkdir -p usr/sbin mkdir -p usr/local/bin mkdir -p usr/local/libexec mkdir -p usr/local/etc #sshd 工作目录 mkdir -p var/run mkdir -p var/empty/sshd chmod sshd var/empty **2、将 openssh-4.6p1目录下的 sshd 拷贝到 目标板的 /usr/sbin 目录下** cp sshd ./usr/sbin **3、将 copy scp sftp ssh ssh-add ssh-agent ssh-keygen ssh-keyscan 到目标板/usr/local/bin目录下** cp scp  sftp  ssh  ssh-add  ssh-agent  ssh-keygen  ssh-keyscan ./usr/local/bin  **4、将 sshd\_config ssh\_config 拷贝到 /usr/local/etc/ 目录下** cp sshd\_config ssh\_config ./usr/local/etc  **5、将 sftp-server  ssh-keysign 到 /usr/local/libexec** cp sftp-server  ssh-keysign ./usr/local/libexec **6、在主机上生成密钥** ssh-keygen -t ecdsa -f ssh\_host\_ecdsa\_key -N "" ssh-keygen -t rsa -f ssh\_host\_rsa\_key -N "" ssh-keygen -t dsa -f ssh\_host\_dsa\_key -N "" 将生成的 ssh\_host\_\*\_key这3个文件copy到./usr/local/etc/目录下 cp ssh\_host\_\* ./usr/local/etc/ **7、在开发板 bin 目录下创建软连接** ln -s /usr/local/bin/scp  ln -s /usr/local/bin/sftp  ln -s /usr/local/bin/ssh  ln -s /usr/local/bin/ssh-add  ln -s /usr/local/bin/ssh-agent  ln -s /usr/local/bin/ssh-keygen  ln -s /usr/local/bin/ssh-keyscan **8、打包** mkdir ../final  cp -r bin sbin usr ../final  cd ../final  tar czvf openssh.tgz ./  将 openssh.tgz 拷贝到开发板根目录解压 **9、启动开发板，添加用户** touch /etc/passwd  touch /etc/group mkdir -p /home/sshd adduser sshd adduser root passwd root 为root用户设置密码 将开发板/usr/local/sshd\_config,将PermitRootLogin yes前的注释“#”号去掉 **10、测试** 开发板 /usr/sbin/sshd ps 可以看到 /usr/sbin/sshd 主机: $  ssh root@192.168.1.17 一开始提示密钥啥玩意，输入yes再输入密码就可以登录了，也可以直接用windows下的[Secure CRT](http://www.baidu.com/link?url=6w6J1ezzfvLQaFRdqt_vcB_uLA4Alepso6A8-eAKfaSgKpjwJWUVHksrCD9DYwFKvmvGMJ1s0aRoSqCLKbNTlot523hmFJF-GuSrty6ArnC)