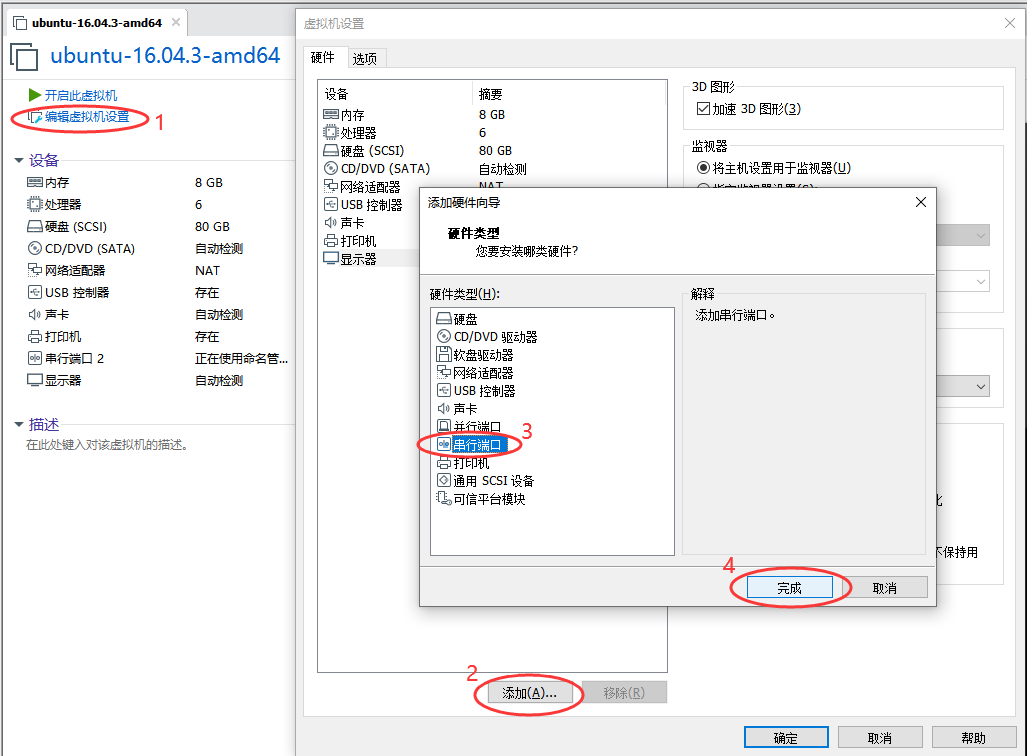
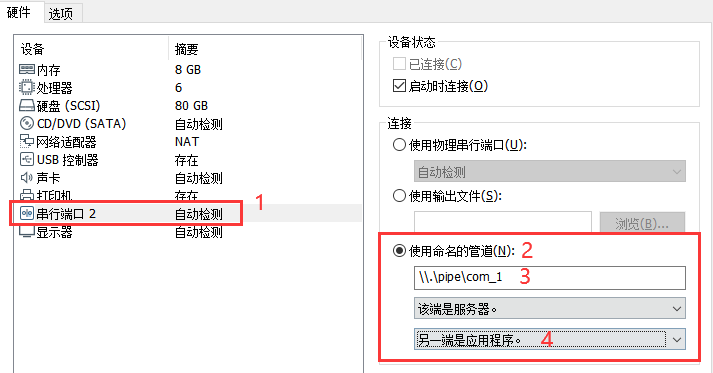
1. windows下使用
2. 使用命名管道
3. 关闭虚拟机（貌似不用也行）；
4. 编辑虚拟机->添加->串行端口->完成；



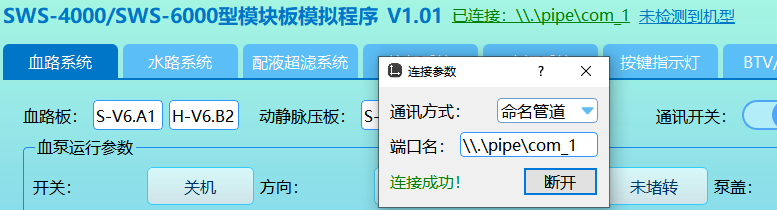
1. 选择添加好的串行端口->使用命名的管道->输入“\\.\pipe\com\_1”->选择“另一端是应用程序”；  
   注：如果你有两个虚拟机，那就命名“\\.\pipe\com\_2”，以此类推；



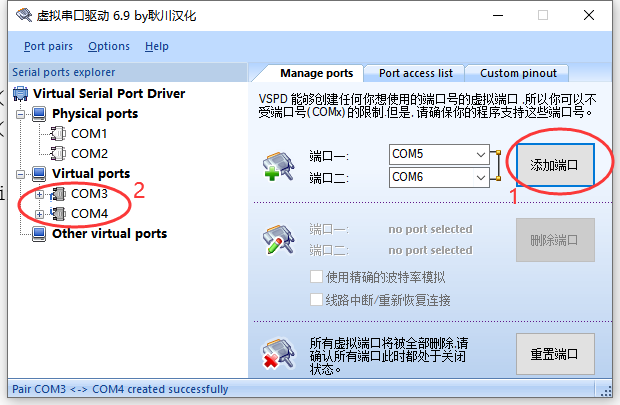
1. 这时就可以启动虚拟机啦，虚拟机内的Linux下就是“/dev/ttyS1”，刚好是6000型的485串口；若是4000型，使用的是“/dev/ttyS3”，怎么办呢？直接“rm /dev/ttyS3”，然后“mv /dev/ttyS1 /dev/ttyS3”（简单粗暴）；

注：4000型暂时还没有完善，无法正常使用。

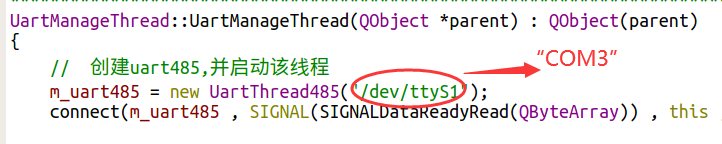
1. 打开模拟器，点击“未连接”，输入端口名，点击连接，若显示连接成功就OK啦；



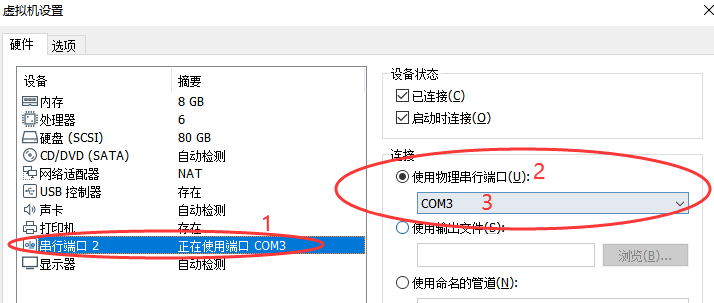
1. 在虚拟机内运行SWS-6000主控软件。
2. 使用串口
3. 下载并安装“Virtual Serial Port Driver”软件；
4. 添加一对虚拟串口（它俩是连在一起的），如我这里叫“COM3”和“COM4”；



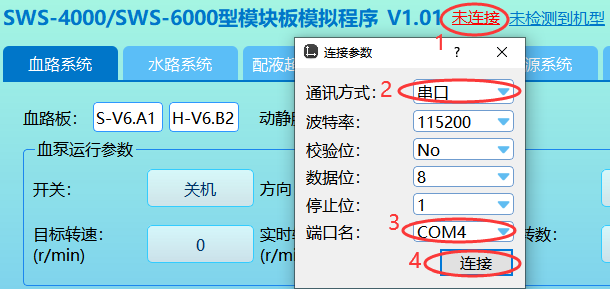
1. 如果是直接在windows上跑主控程序的话，按照下图修改代码，就OK了，否则进入下一步；



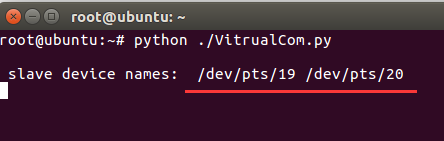
1. 如果在Linux上跑主控程序，编辑虚拟机->添加->串行端口->完成（同一(1)）；
2. 选择刚添加的串行端口，选择“使用物理串行端口”，选择端口号为刚才添加好的虚拟串口号（我这里是“COM3”），点击确定后启动虚拟机；



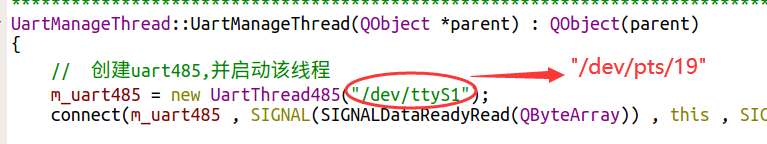
1. 相当于此时“COM3”与虚拟机内部“/dev/ttyS1”相连，我们打开模拟器，选择“未连接”，通讯方式选择“串口”，端口名选择“COM4”，点击连接即可；  
   PS：相当于此时“COM4”就和“/dev/ttyS1”相连了；



1. 在虚拟机内运行SWS-6000主控软件。  
   PS：不建议使用这种方式，通讯一段时间后，数据可能会乱。
2. Linux下使用
3. 使用python运行脚本“VitrualCom.py”（代码附予最后），记下打印出的两个虚拟串口号，我这里分别是“/dev/pts/19”和“/dev/pts/20”；



1. 按下图修改主控程序代码；



1. 打开模拟器，选择“未连接”，输入端口名（“/dev/pts/20”）后，连接即可；



1. 运行SWS-6000主控软件。

附，“VitrualCom.py”源码：

|  |
| --- |
| #! /usr/bin/env python  #coding=utf-8  import pty  import os  import select  def mkpty ():  master1 , slave = pty . openpty ()  slaveName1 = os . ttyname ( slave )  master2 , slave = pty . openpty ()  slaveName2 = os . ttyname ( slave )  print ' \n slave device names: ' , slaveName1 , slaveName2  return master1 , master2  if \_\_name\_\_ == "\_\_main\_\_" :  master1 , master2 = mkpty ()  while True :  rl , wl , el = select . select ([ master1 , master2 ], [], [], 1 )  for master in rl :  data = os . read ( master , 128 )  #print "read %d data." % len ( data )  if master == master1 :  os . write ( master2 , data )  else :  os . write ( master1 , data ) |