

课程名称：Java应用技术 指导老师：翁恺 学生姓名：尹嘉权

实验名称：SerialTool 实验类型：编程实现 学生学号：3120000419

1. 实验目的和要求

目的：

利用Java写一个GUI program，主要实现串口之间的二进制数据的read和write，利用JFrame, JPanel, JLabel, JTextArea, JScrollPane 等框架来编写UI界面，然后通过导入java.comm的官方库，来进行串口通讯之间的代码实现。

要求：

Make a GUI program that reads and writes to serial ports in binary data. I'll explain the program at the 3rd lecture. So don't miss it. The attachment is a screenshot from a Mac OS app called UARTCom.

1. 实验内容和原理

内容：

编写GUI程序来对串口之间进行二进制数据的通讯（read 和 write）。

原理：

利用JFrame, JPanel, JLabel, JTextArea, JScrollPane 等框架来编写UI界面；

通过导入Java(tm) Communications API[[1]](#footnote-1)，来进行串口之间的代码实现。

1. 主要仪器设备

MacBook Pro, MacOSX 10.10, 2.6GHz Intel Core i5, 8G RAM

JSE: JavaSE-1.8

JDK: oracle-java8-jdk

1. 操作方法和实验步骤
2. 利用JFrame, JPanel, JLabel, JTextArea, JScrollPane 等框架来编写UI界面，由于之前几次实验都是需要绘制UI的，这里面不再赘述。需要注意的是，对于Status Panel的选择的时候，利用的是java.awt.Choice，把对应的选项加入到同一个JLabel里面，然后通过getSelectedItem()函数获得即可；对于Receive的JTextArea，必须设置为不可修改，即setEditable(**false**)。
3. 下载并导入java.comm 官方库，这是一个官方的java communication API，然后解压之后把comm.jar作为extension jar导入到项目中；
4. 学习如何编写串口通讯，具体可以参考官方压缩包/commapi里面的/examples/Simple/SimpleRead.java 和 /examples/Simple/SimpleWrite.java
5. 几个注意的地方，在link的时候必须重新获取status panel上面的参数，然后再重新link；在break的时候需要关闭串口之间的通讯；在send的时候通过setSerialPortParams来设置新的参数，然后过滤掉非16进制的字符，之后通过getOutputStream得到的outputstream来进行串口的write；重写serialEvent函数的时候，对inputstream得到的串口接收的信息须同样以16进制的形式来进行显示。通过线程的实现来实现一定的延迟，来保证串口之间的通讯数据没有被reflush掉。
6. 同时，由于受commapi的限制，程序必须得在32位linux系统上运行，同时得在本地重新编译，不然没法找到

/commapi/docs/javax.comm.properties 和

/commapi/lib/libLinuxSerialParallel.so[[2]](#footnote-2)

具体的设置过程可以查看原压缩包里面的

/commapi/docs/html/platform/linux/ PlatformSpecific.html

1. 编写对应的arduino程序，由于程序较短，所以在报告里面列出，读者也可以查阅ReadandWrite/ ReadandWrite.ino

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

while (Serial.available()){

char c = Serial.read();

Serial.println(c);

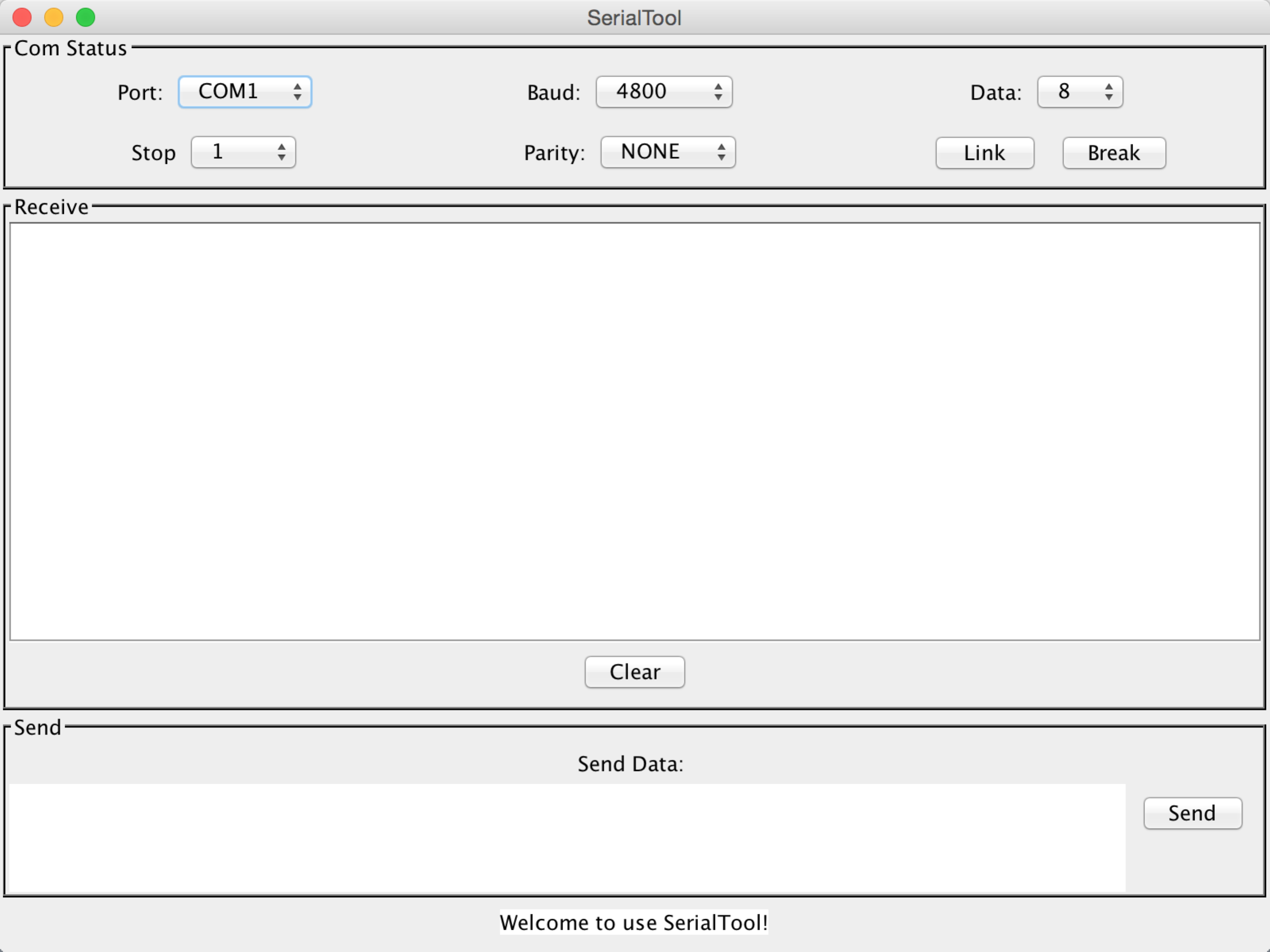
}

}

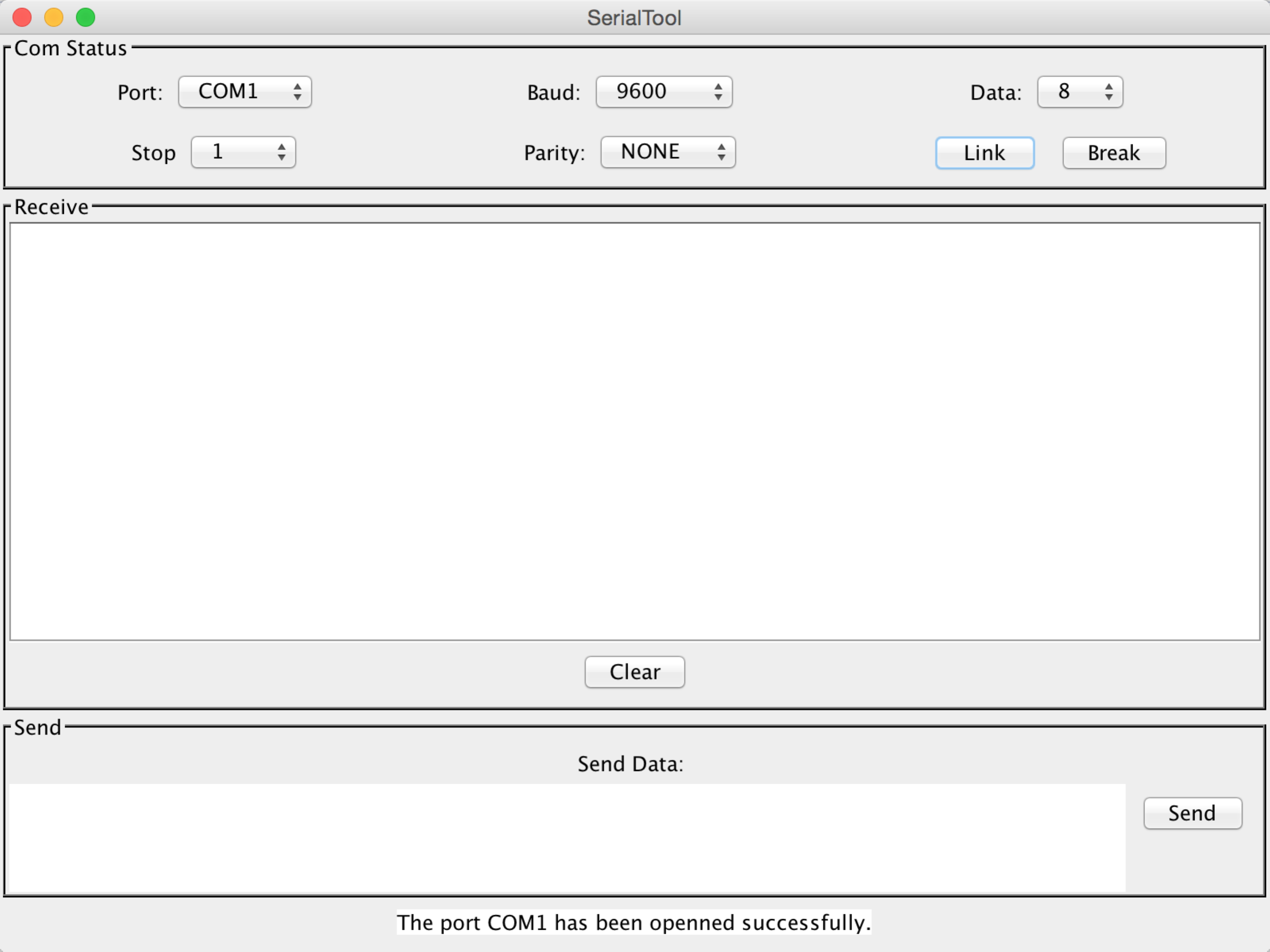
实质就是把arduino输入的东西按照原样输出，但是注意这里面是以一个字符的形式来作为串口的输入输出。

1. 在linux下安装好minicom来进行串口的一些基本配置，把arduino板设置为COM1，不然在link的时候会出错。（当然设置成别的串口也可以，但是要求要在Port口要对的上才行）
2. 实验数据记录和处理

程序的基本界面：



在Com Status栏可以通过相对应的下拉菜单来选择对应的参数，这里面已经用minicom来把端口号设置为COM1，然后在arduino程序里面设置的是Baud=9600，这时候把Baud设置为9600，然后点击Link：



可以看到，下面的提示符已经变成成功打开，如果是没有任何变化，查看terminal或者说Eclipse下的Console输出，看是否因为没有找到

/commapi/docs/javax.comm.properties 和

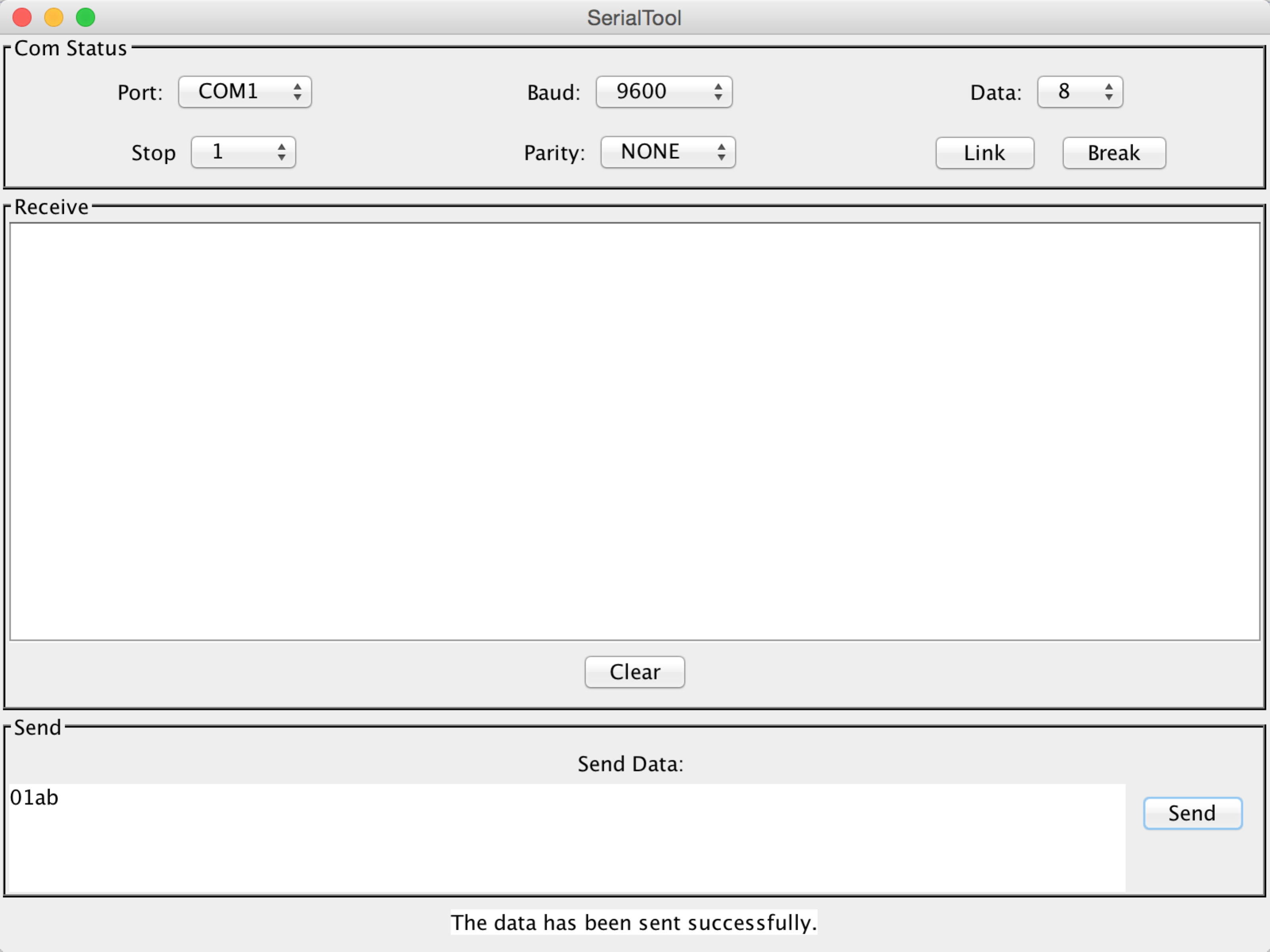
/commapi/lib/libLinuxSerialParallel.so

这两个文件导致

*portList* = CommPortIdentifier.*getPortIdentifiers*();

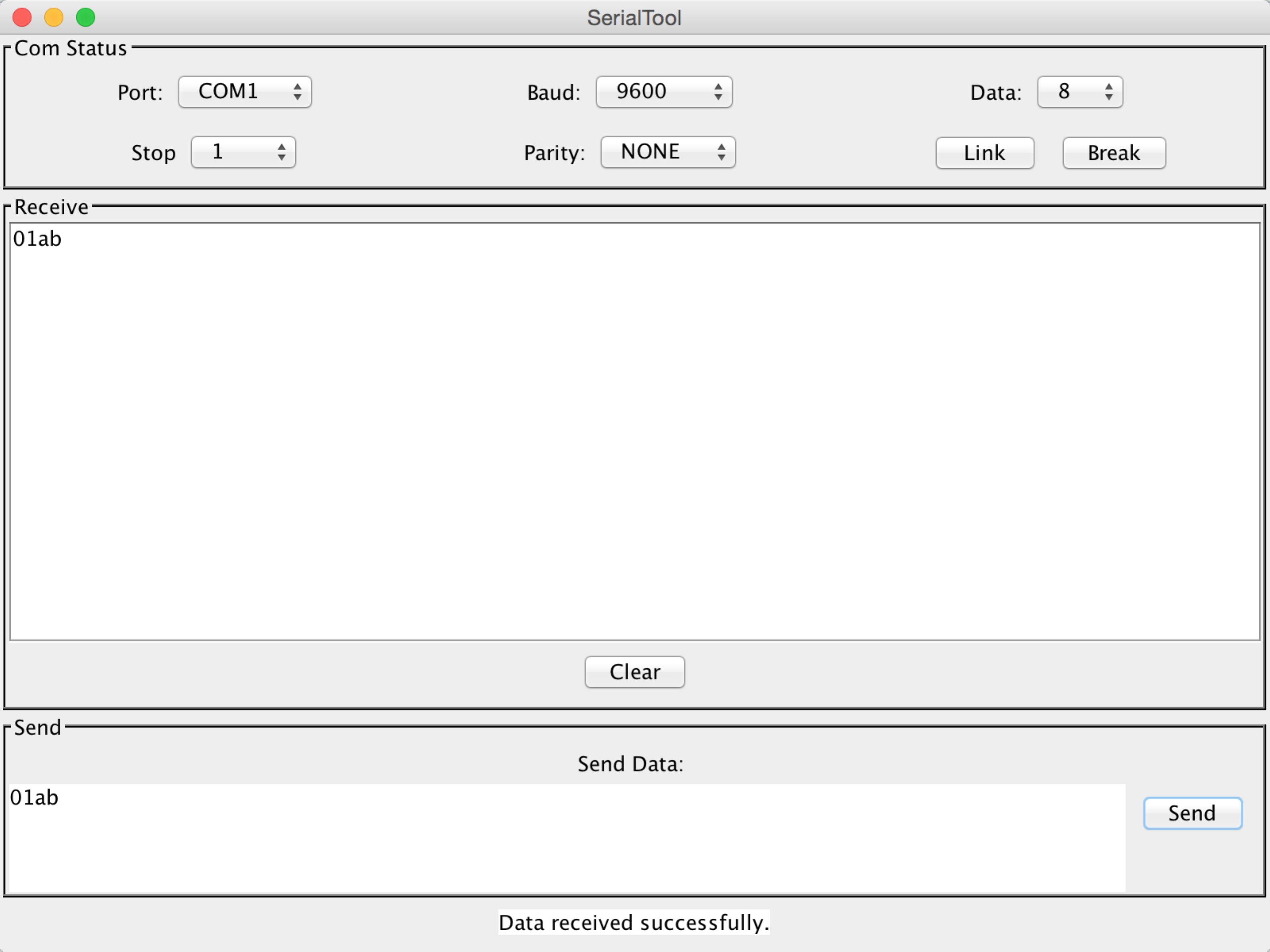
出错

然后输入一些内容，点击send



可以看到下面状态栏变成成功发送，同时可以看到arduino上面的RX灯亮了一下。

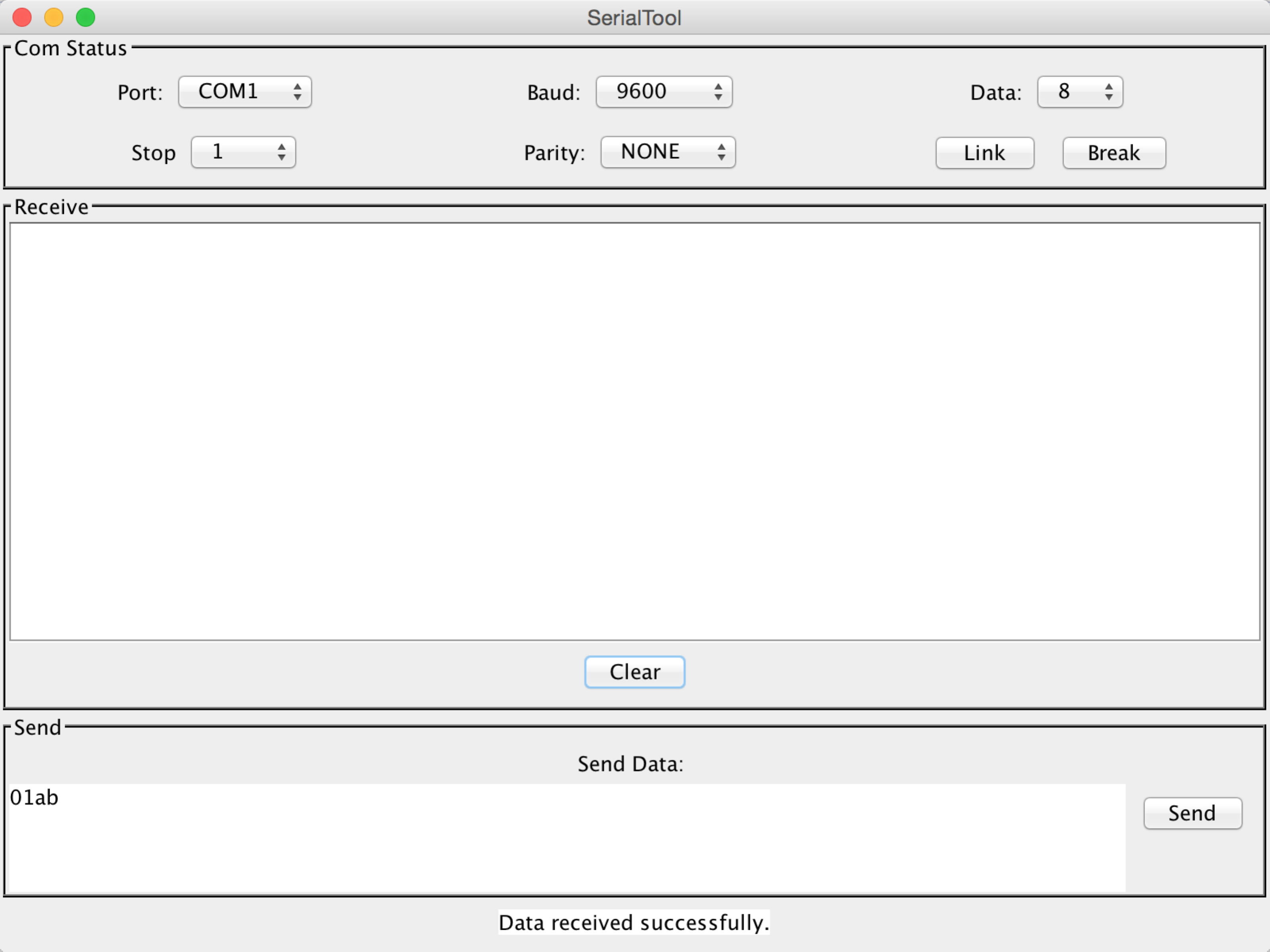
随后很快地，TX灯亮了一下，然后在UI界面上面接收到一样的数据内容：



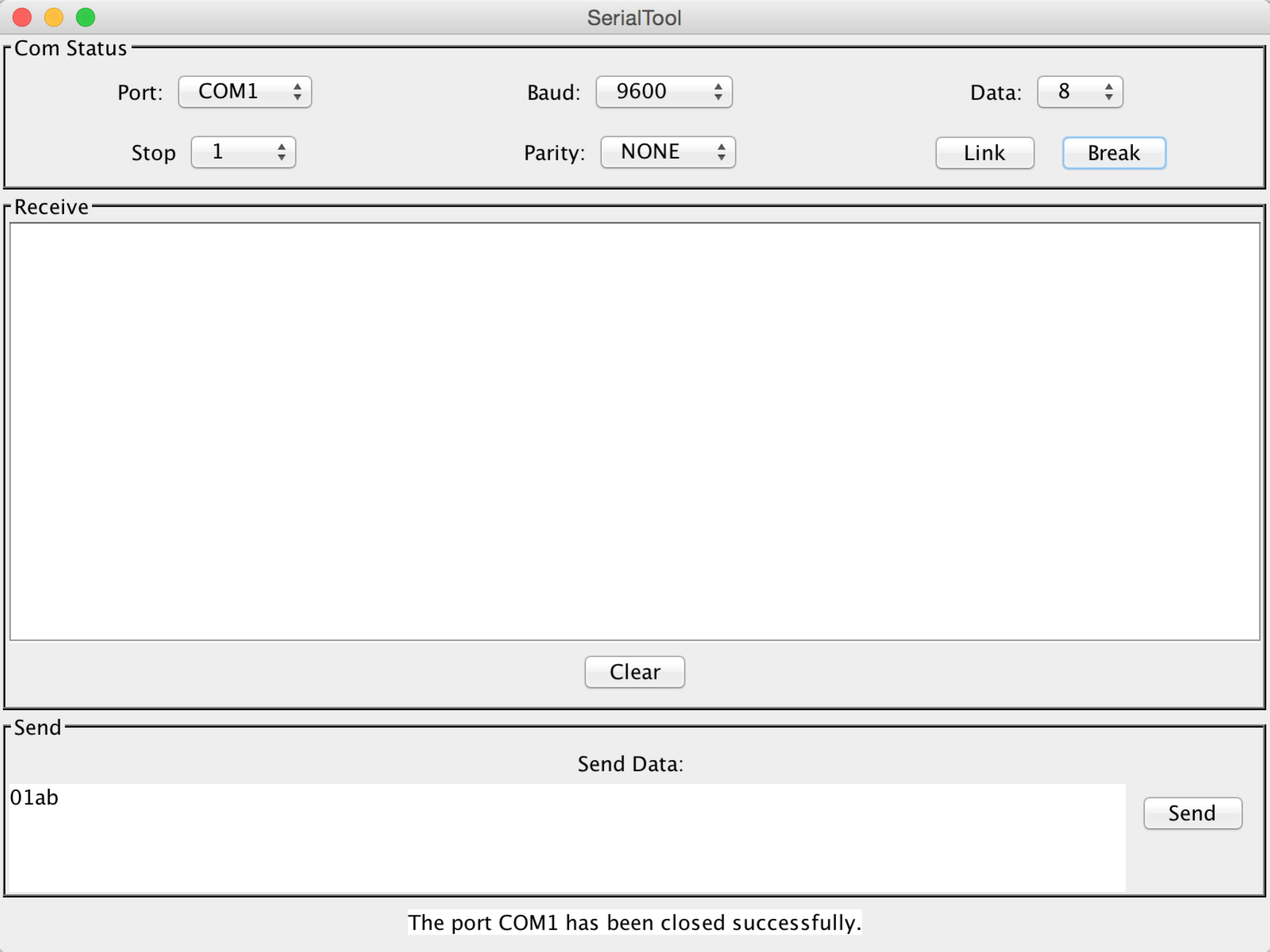
同时最下方的状态栏变成数据成功接收的欢迎语。

然后测试clear:

按了clear之后，Receive那一栏的内容没了，另外不修改最下方的欢迎语：



最后测试break：



最下方的欢迎语提示串口已经被成功关闭，结果正确！

1. 实验结果与分析

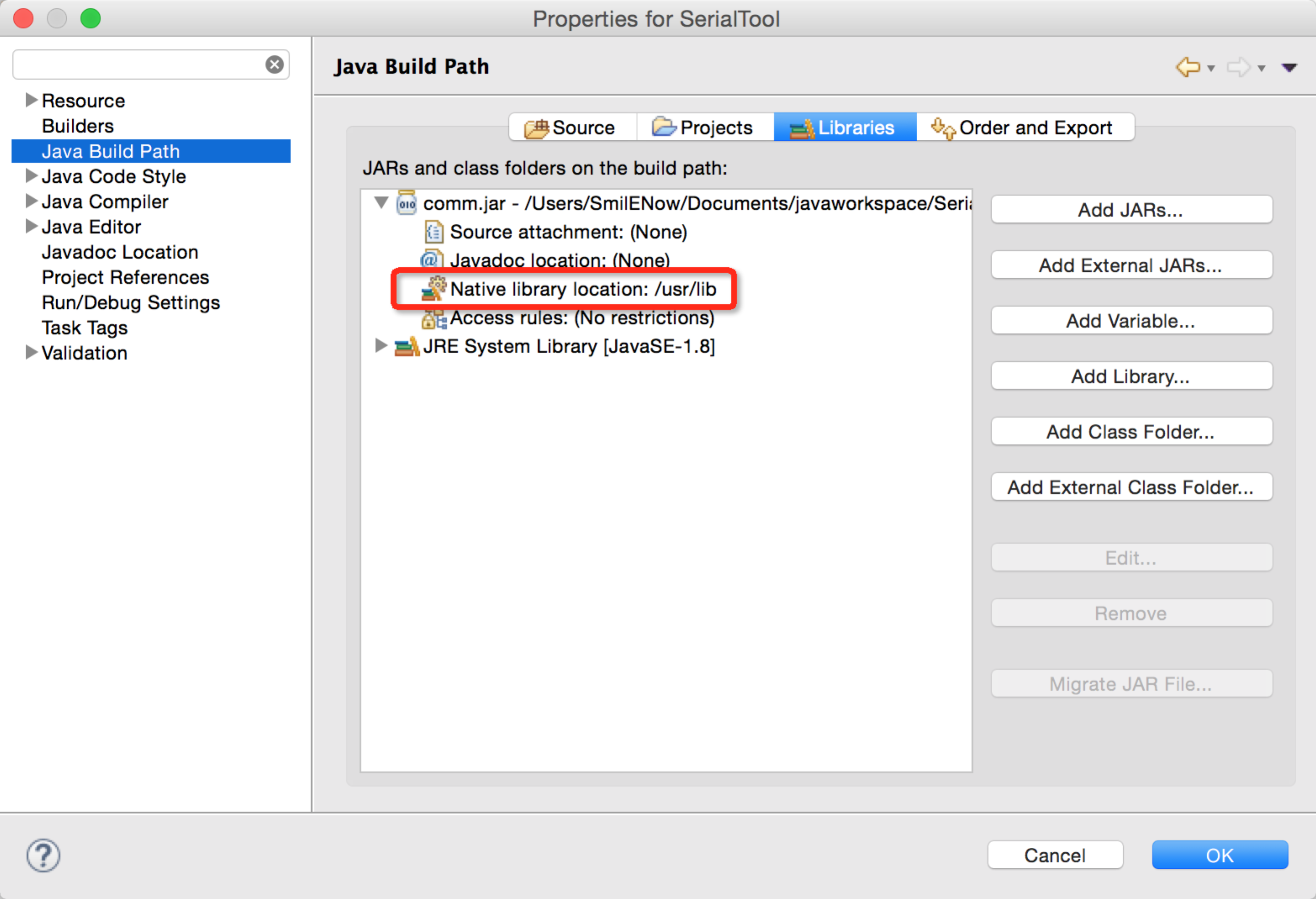
实验输出正确。

1. 讨论心得

通过这次实验，对于在java上面UI的绘制有了更深的理解，对于一些文本区域的属性设置，让程序变得更加多元化，同时也是必须的，比如说不能修改Receive框的内容。

另外，这次大程除了编写代码之外，在查找如何正确使用java.comm上面花了很大功夫，官方提供的jar下载之外，别的信息相对来说几乎没有。一开始在Mac OS X上面，Import了第三方包发现一只没法正确运行程序，后来看了压缩包里面关于linux下面的 PlatformSpecific.html才意识到要把/commapi/docs/javax.comm.properties 和

/commapi/lib/libLinuxSerialParallel.so 导入到对应位置，同时也需要对Eclipse上面的Native Library location做对应的设置才行



但是当设置好之后，还是提示

java.lang.UnsatisfiedLinkError: no LinuxSerialParallel in java.library.path

之后查阅相关资料，才认识到在mac下是无法识别.so的，而由于没有源码，因此不能通过本地手动编译成.dylib，所以放弃了在mac上使用，转战官方源包里面提示的Linux的设置步骤，一步一步设置好之后，如果能跑通Example基本上就说明设置没什么问题了，然后再对程序进行测试。

整个大程下来，对串口的东西有了基本的认识，也算是为下个学期嵌入式系统这门课稍作准备吧！

1. Java Communications API 3.0 : http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-misc-419423.html [↑](#footnote-ref-1)
2. 笔者没有找到可以把.properties导入到jar的方法，如果老师、助教知道，烦请email to [troysmilenow@gmail.com](mailto:troysmilenow@gmail.com) [↑](#footnote-ref-2)