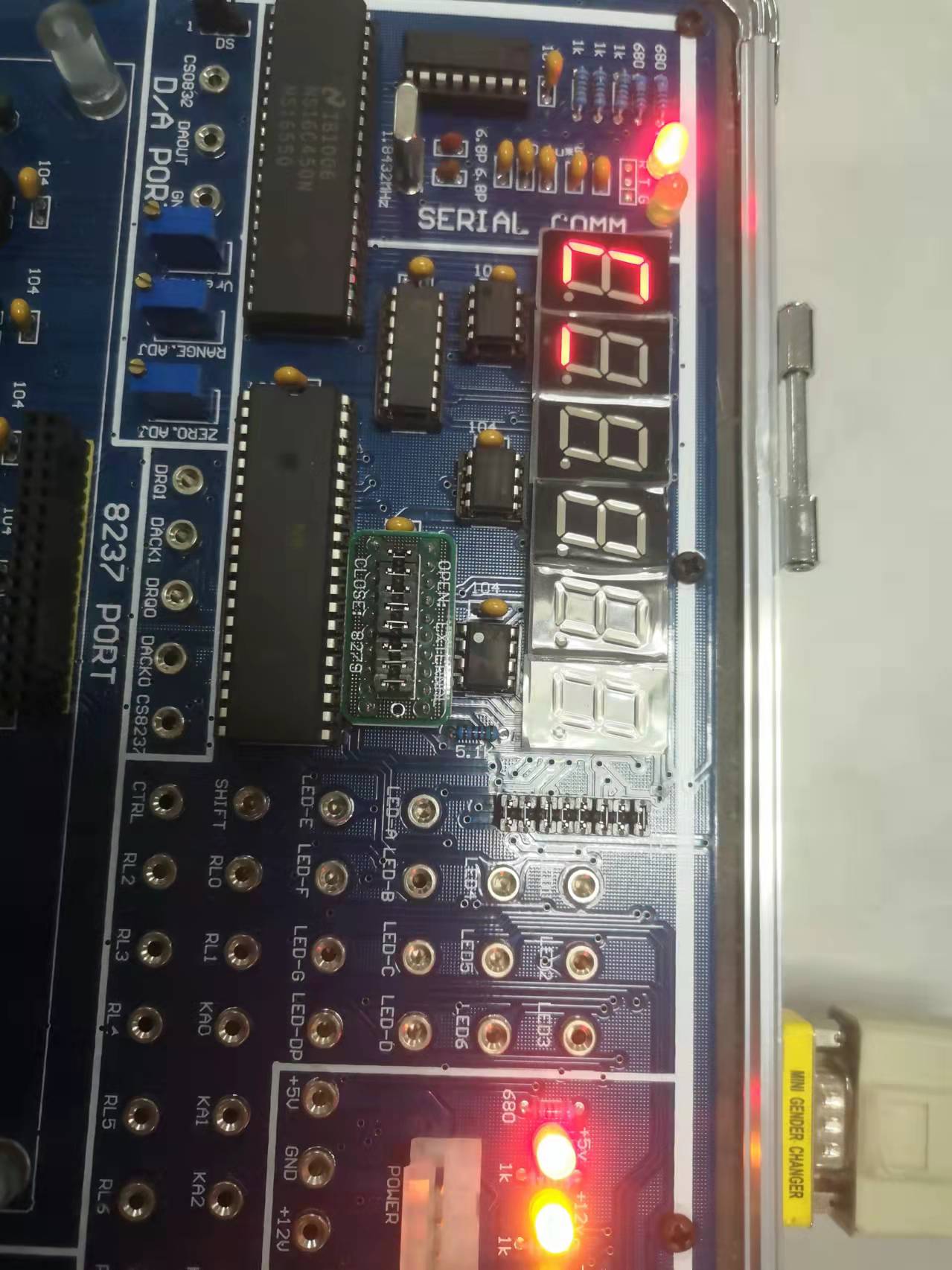
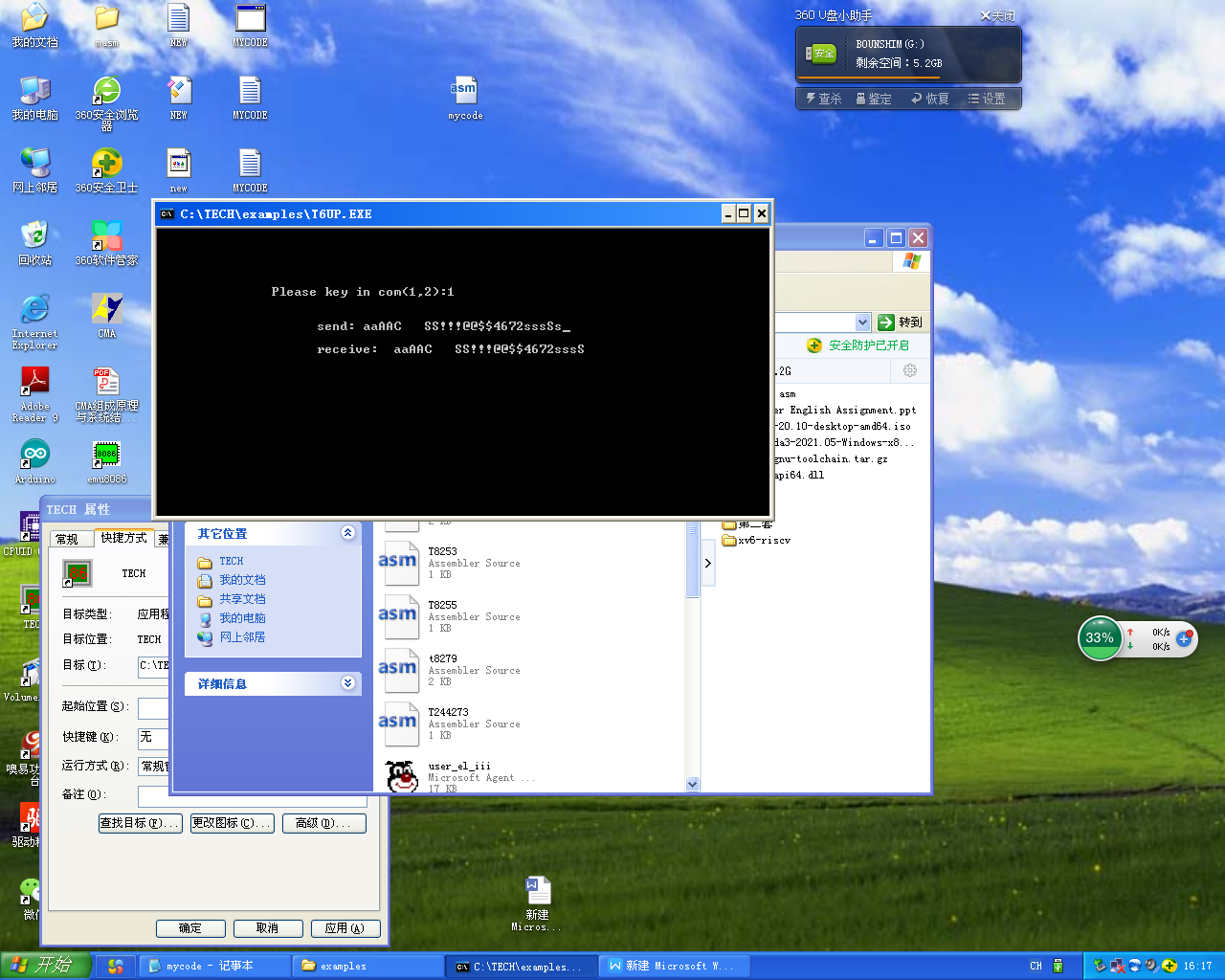
# 设计过程

1. **实验箱**

实验箱内置电路已经接好，本实验不用进行额外的连线。首先将事先写好的双工通信汇编程序导入8086实验系统，选择编译并链接，然后选择系统复位，并按下实验台上的复位按钮，此时实验台数码管显示情况如下：

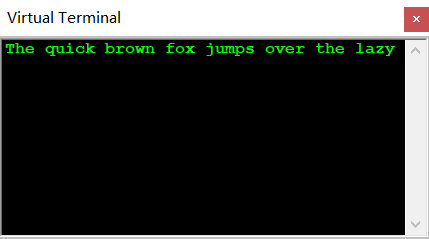


成功复位后全速运行程序，并退出8086实验系统。找到T6UP.EXE文件并打开执行，即可看到通信窗口。首先选择COM1串口（与8086实验系统中的选择要一致），然后在发送端输入任意字符，接收端便可接收到相应字符，任意发送一段字符串，结果如下：



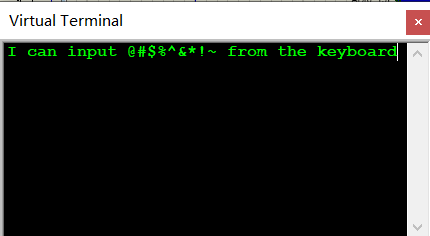
1. **必做实验1**

实验需要实现单工通信，所以应当在程序中编写好需要输出的字符串，并在打开虚拟终端时可以直接接收到。编写程序时，需要依次设置传输速率、数据包格式和中断允许。选择波特率为9600，因此对应的波特率因子为12，发送信息时，每读取一个字符后都将其入栈保护，直到发送保持寄存器为空，则出栈读取的字符并显示，重复以上过程直到读取完整个字符串。实验发送字符串“The quick brown fox jumps over the lazy”的结果如下：



1. **必做实验2**

实验要求实现双工通信，因此要在虚拟终端上实现发送和接收，也即从键盘输入一个字符，终端上立马能将该字符显示出来。与单工不同的是，双工通信发送的字符串是键盘输入的，因此要想及时接收到刚刚输入的字符，可以将收发端各写成一个子程序，且每次只收/发一个字符，在主程序中调用完发送端后立即调用接收端，如此循环，即可实现双工通信。任意输入字符串，结果如下：



# 实验总结

## 心得体会

本次实验让我对串行通信的单双工通信方式的原理和实现方法更加熟练，也对8250的工作原理有了更深刻的理解，能够利用8250编程解决一些基本的信息通信问题。

单工通信的仿真实验中，需要注意的是，在定义字符串后，默认以BAH为结尾，可以在单步调试程序的情况下查出对应值，如下图所示：



实验台和Proteus仿真对于双工通信的实现略有不同，实验台在发送一个字符后需要发送下一个字符才能接收上一个字符，而仿真软件上双工通信则是直接一发一收。

## 遇到的问题及解决方案

1. **必做实验1无法收到所发送的信息**：单工通信的电路图起初采用74154作为译码电路，但没修改其他地方的情况下改用74LS138作为译码电路，虚拟终端上则显示了发送出的信息。
2. **必做实验2无法输入字符**：这与8086程序加载段的设置有关。一开始Program Loading Segment设置为0x0800时，无法运行出正常结果，但将其改为0x0000又可以正常运行。综合分析了之前实验的程序发现，在不使用子程序，数据段中不设置变量的情况下，该问题不会发生。使用0000的情况下大多因为有类似code segment assume cs:code的定义语句，但仍要使程序加载段为0800可以尝试将数据段定义在代码段内，这种做法对某些实验有效，仍缺少普适性。