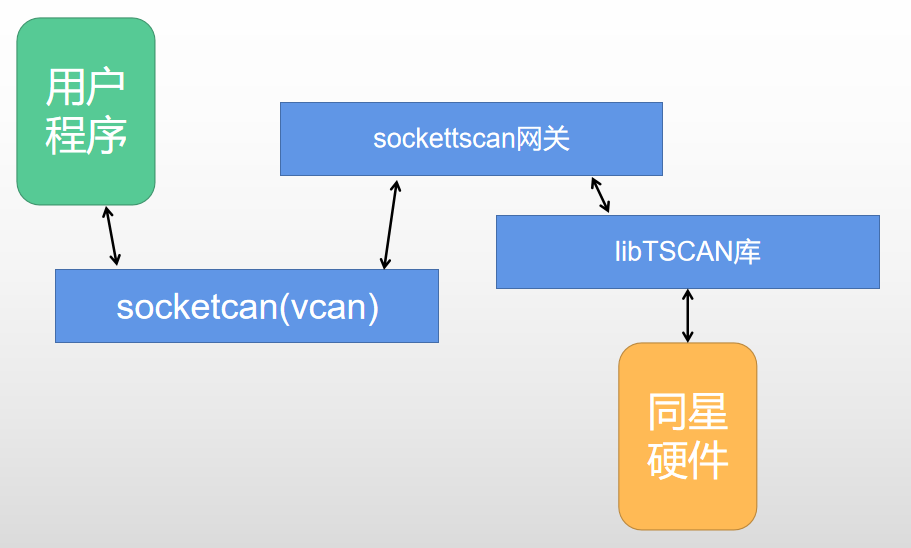
如果想要利用socketcan控制TSCAN的设备，有一个解决方法是利用linux自带的虚拟socketcan。我们利用同星提供的linux库libTSCANApiOnLinux.so，完成一个vcan在本机的server程序，用于把socketcan的报文和libTSCAN进行互转。这种方法虽然有很多弊端，比如无法查看到正确的统计数据，但是只进行收发是没有问题的。

这种方式配置同星硬件的波特率，以及查看信息都需要在sockettscan程序中。因为vcan本身不支持波特率配置，我们也无法把统计信息告知vcan。

逻辑图如下：



首先我们需要使能vcan，并创建vcan的通道。

> modprobe vcan

> ip link add dev vcan0 type vcan

> ip link set vcan0 up

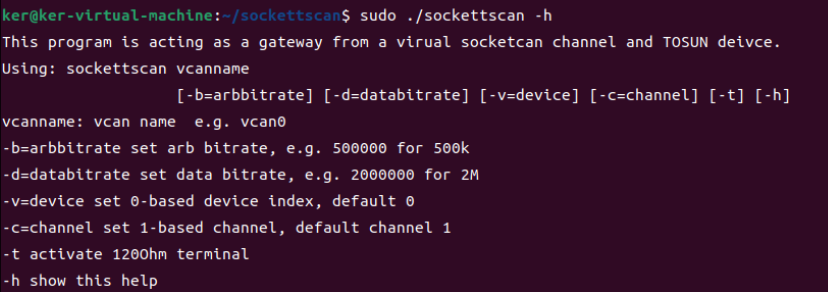
然后运行sockettscan程序（注意需要root权限）：

//绑定到设备0的1通道，波特率500k 2M，使能终端电阻

> sudo ./sockettscan vcan0 -b=500000 -d=2000000 -v=0 -c=1 -t

这样通过socketcan调用vcan0就可以把报文同步到TSCAN设备中了。

可以通过sudo ./sockettscan -h显示帮助信息：



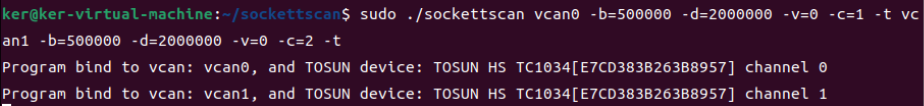
测试：

设备双通道互相通信。

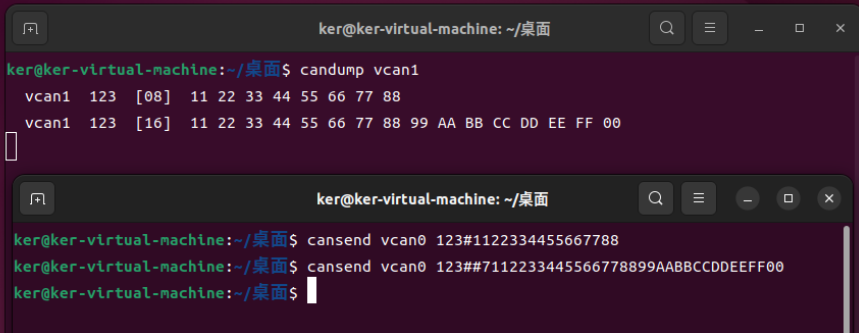
A 硬件：TC1034，需要连接CAN1和CAN2。

通过上述方法建立vcan0和vcan1两个通道。

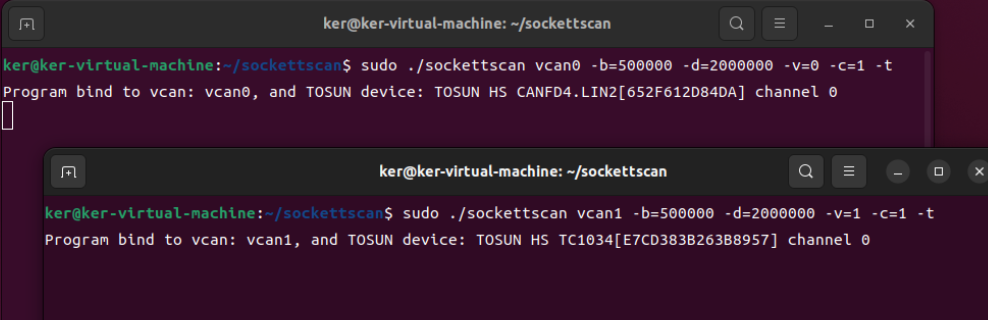
利用sockettscan绑定两个通道：关于多通道使用需要注意，每一个sockettscan用来连接单个TSCAN硬件。多个通道可以写多次（vcan0名称需要在最前面），多个硬件设备，可以在多个终端调用sockettscan实现。



通过cansend和candump测试通信，效果如下：



B 硬件TC1034和TC1016，使用两个设备的通道1



测试通信如下：

