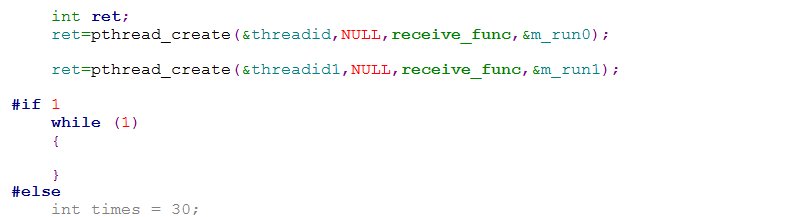
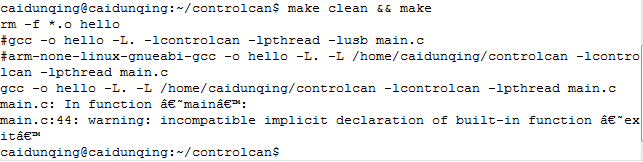
USBCAN操作流程V1.2

1. 将对应系统的controlcan文件夹，拷贝到linux桌面。
2. libcontrolcan.so、controlcan.h为二次开发库文件，直接引用即可。内部集成了USB驱动，不用单独安装USB驱动。
3. main.cpp为二次开发样例，一个简单的测试样例，可以完成数据的收发。
4. Makefile文件，编译文件，编译前需要修改相应的路径：g++ -o hello\_cpp main.cpp /home/ttc/Desktop/controlcan/libcontrolcan.so -lpthread（按实际路径修改）。
5. hello\_cpp为编译后生成的可执行文件。
6. Ctrl+Alt+T：打开命令窗口。
7. 按照ubuntu命令.TXT文档中的命令运行相应的样例。
8. 样例测试步骤一：ubuntu系统下设备用于接收

a. 在main.c文件中把图示中设置为#if 1

b. 保存后 执行 make clean && make



c. 硬件上把两个USBCAN设备相连（如果没有两个USBCAN设备，可以使用其它的CAN设备代替，或是修改代码使两个通道间相互通信），另一个设备连接到window系统下，且打到cantest工具

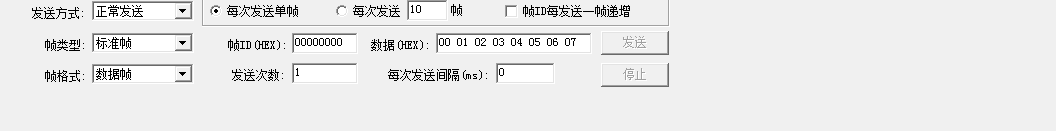


示例例程中使用的是125K的波特率，因此这里也要使用125K的波特率

d. 用root权限执行hello软件



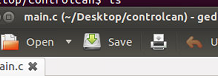
e. 用cantest工具软件发送数据给ubuntu下的设备。

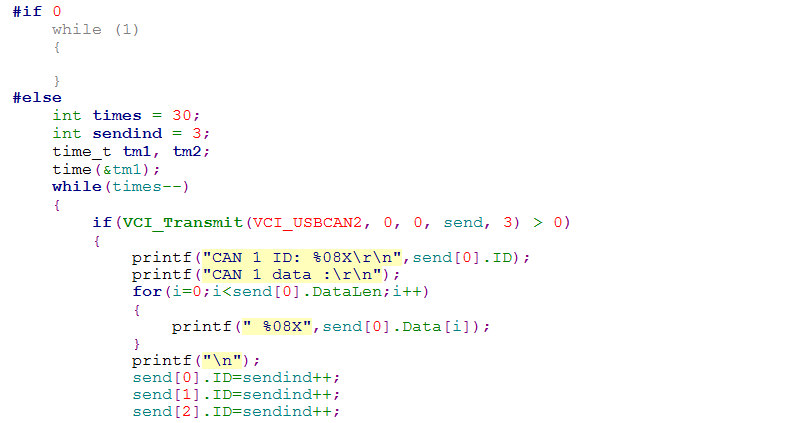


f. 此时就可以看到ubuntu系统下的终端上有一些数据被接收到。

四、ubuntu系统下设备用于发送

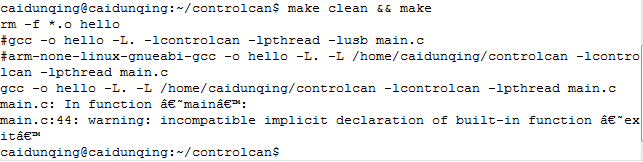
如果不会用vi编辑器，用gedit main.c直接更改也是一样的，更改后，保存,点击Save。

c



a. 把main.c代码中的#if 1改为#if 0，即运行while(1)下面的代码

b. 保存后 执行 make clean && make



c. 硬件上把两个usbcan设备相连，另一个设备连接到window系统下，且打到cantest工具



示例例程中使用的是125K的波特率，因此这里也要使用125K的波特率

d. 用root权限执行hello软件



e.在cantest工具软件上就可以看到一些数据被接受到。