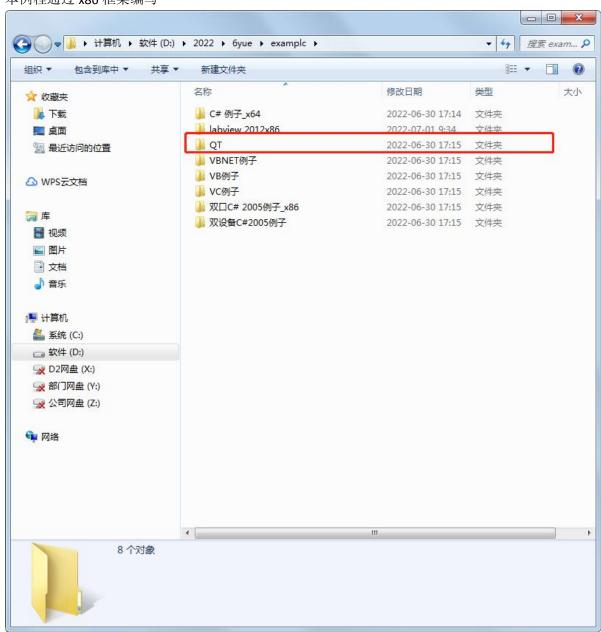
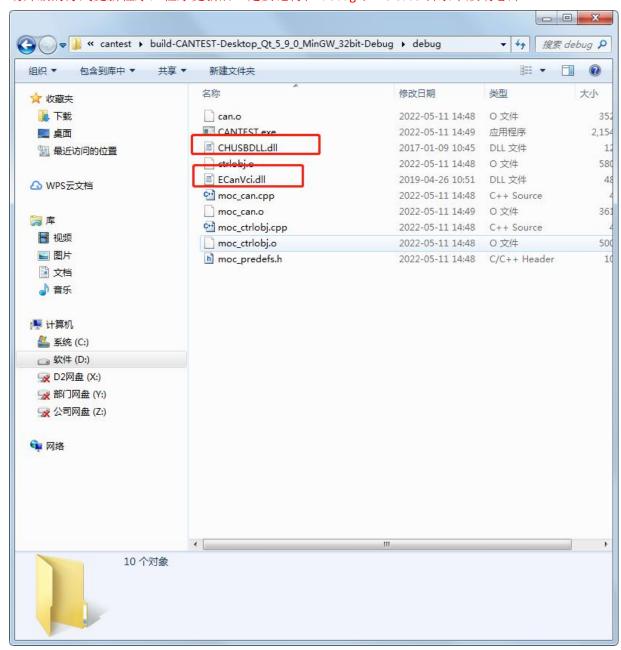
USBCAN QT 例程使用说明

本例程通过 x86 框架编写



注意: QT 语言的程序一定要放到英文路径下,所以要保证存放工程的路径下没有中文。 使用 qt 例程一定保证自己电脑 qt 环境安装完整,且可以正常运行 widgets 框架程序,可以自己先新建一个程序,随便加一个按钮试一下。

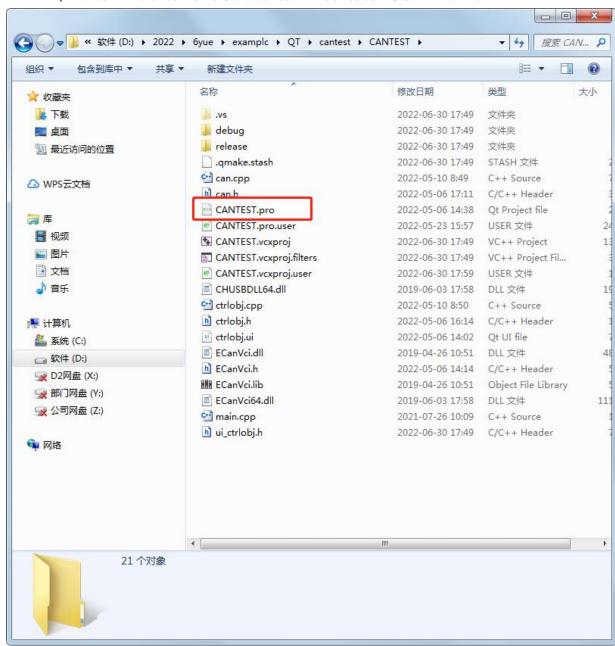
为了保证通用性所以我们提供的例程都是基于老版本环境编写,如版本不同可以编译器会自动升级的方式更新程序,程序更新后一定要记得在 debug 和 Release 目录下放动态库。



QT 例程使用方法(使用 Qt Creator 打开)

1. 打开工程

CANTEST.pro 文件是工程文件,如果使用需要打开这个文件会直接打开整个工程。



2. 工程文件介绍



can.cpp 主要是动态库相关逻辑、ctrlobj.cpp 是与界面相关逻辑

如果直接使用例程需要需要把构建调后才能 DEBUG, 然后点击运行



4. 例程包含单通道的参数设置和收发数据功能

■ ctrlobj		
波特率:		
帧类型:	标准帧 ▼	
帧格式: 帧ID: 帧数据:	数据帧 ▼ 00000008	
"现金人"和。	00 01 02 03 04 05 06 07	

QT动态库调用方法

```
🤔 can.cpp @ CANTEST - Qt Creator
文件(F) 编辑(E) 构建(B) 调试(D) Analyze 工具(T) 控件(W) 帮助(H)
        项目
        △ 腸 CANTEST
                                                #include "can.h"
  #
             CANTEST.pro
                                                #include <QThread>
                                               bool IsOpenFlag=false;

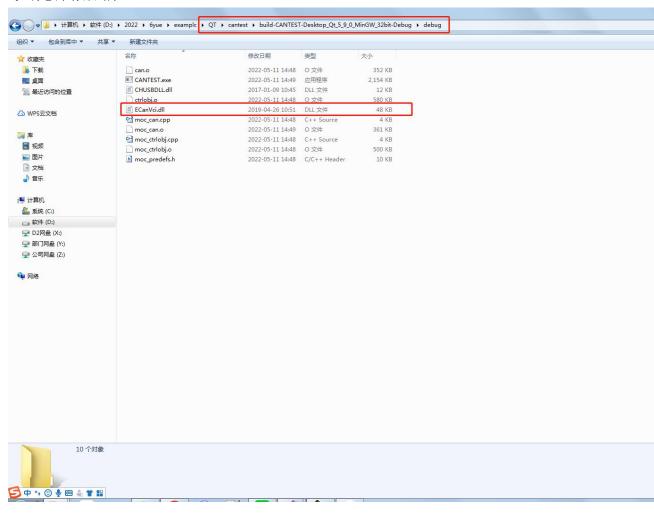
■ Headers

  long mrec=0:
               n can.h
                                             ▲ Thread::Thread()//构造函数//初始化变量
  编辑
               n ctrlobj.h
                                               {
               ECanVci.h
                                                    QLibrary lib("ECanVci.dll");

■ Bources

                                                    if(true==lib.load())
               can.cpp
                                                        qDebug()<<"ECanVci.dll load ok";
  Ú
               ctrlobj.cpp
               em main.cpp
                                                    pOpenDevice = (OpenDevice *)lib.resolve("OpenDevice");
  グ
項目
           ■ Forms
                                                    pCloseDevice = (CloseDevice *)lib.resolve("CloseDevice");
               / ctrlobj.ui
                                                    pInitCAN = (InitCAN *)lib.resolve("InitCAN");
                                          14
                                                    pStartCAN = (StartCAN *)lib.resolve("StartCAN");
  0
                                                    pTransmitCAN = (Transmit *)lib.resolve("Transmit");
                                                    pReceive = (Receive *)lib.resolve("Receive");
pGetReceiveNum = (GetReceiveNum *)lib.resolve("GetReceiveNum");
                                          18
                                                    pClearBuffer = (ClearBuffer *)lib.resolve("ClearBuffer");
                                          19
                                                    pReadErrInfoCAN = (ReadErrInfo*)lib.resolve("ReadErrInfo");
                                          20
                                                    pResetCAN=(ResetCAN *)lib.resolve("ResetCAN");
                                                    pNetReference=(SetReference *)lib.resolve("SetReference");
pReadBoardInfo=(ReadBoardInfo *)lib.resolve("ReadBoardInfo");
pReadCANStatus=(ReadCANStatus *)lib.resolve("ReadCANStatus");
                                          25 / void Thread::run()
                                          26
                                          27 4
                                                    if (musbcanstart==true)
                                          28
                                                        while(!stopped)//死循环//
                                          29 4
                                          30
                                                        {
                                                             ReceiveCANThread();//接收数据
                                                        stopped = false;
                                          34
                                          36
                                               1
                                          38 / void Thread::stop()
                                               {
                                          40
                                                    stopped = true;
                                               }
                                          41
 ,
                                          42
                                          43 🗸 void Thread::ReceiveCANThread()//接收数据线程
                                          44
                                               {
                                          45
                                                    bool ok;
                                          46
                                                    ERR_INFO vei;
                                          47
                                                    CAN_OBJ preceive[1000];
  1
                                          48
                                                    CAN_OBJ psend;
                                                      int baud=0x10000000;//参数有关数据缓冲区地址首指针
                                     1 问题 2 Search Results 3 应用程序输出 4 编译输出 5 Debugger Console 6 概要信息 8 Test Results 
 中。, 🙂 🎍 📟 🐁 👕 🔡 e (Ctrl+K)
```

QT 动态库存放路径



如运行例程后点击链接,出现如图所示情况说明动态库位置存放不对,清继续参考上一步(动态库使用错误)

