

# 二次开发示例程序 操作说明书

说明书版本：V2.04

说明书更新日期：2019.12.17

## 目录

第一章 序言 .....	2
1.1 概述 .....	2
1.2 示例代码 .....	2
第二章 示例程序操作说明 .....	3
1. C#_Example .....	3
2. C++ Builder6_Example .....	4
3. delphi7_Example .....	5
4. LabVIEW Example(8.6) .....	7
5. LabWindows CVI Example .....	8
6. USB CAN Tool 源代码(LabVIEW 2011 环境) .....	9
7. VB.NET Example .....	10
8. VB_Example .....	11
9. VC_Example .....	13
10. QT .....	15
11. python .....	16

# 第一章 序言

## 1.1 概述

首先感谢您对我公司USB\_CAN系列产品的支持与厚爱。

我公司的USB\_CAN系列产品支持二次开发，并提供了绝大多数主流开发平台的开发样例源代码，本说明书针对每种平台上的二次开发样例程序的操作流程和注意事项做简要说明，希望能够在您对产品进行二次开发时，能够更快速更好的完成相关项目。

详情请参见以下各章节。

## 1.2 示例代码

我公司提供市场上最为齐全的二次开发示例代码，并在不断完善更新更多的开发平台。

目前版本，包括以下示例：

代码名称	开发环境	备注
C#_Example (VS2005)	VS2005	
C#_Example (VS2008)	VS2008	
C++ Builder6_Example	C++ Builder6	
Delphi7_Example	Delphi7	
LabVIEW Example (8.6)	LabVIEW 8.6	
LabWindows2013_Example	LabWindows CVI 2013 SP1	
USB_CAN Tool 源代码 (LabVIEW 2011环境)	LabVIEW 2011	本源码是随机配送的专业调试工具 USB_CAN Tool的源代码，功能强大，不是一般样例代码可比的！
VB.NET Example	VS2005	
VB_Example	VB6.0	
VC_Example_VC6.0	VC6.0	
VC_Example_VS2008	VS2008	
Example For Send File.txt	自带USB_CAN Tool文件发送 帧序列定义示例	

每个示例源代码都存放在独立的文件夹下面，并都已经调试通过，已编译为exe可执行文件，一般情况exe文件都存在于源码的一级目录下面。这些exe文件可直接运行。

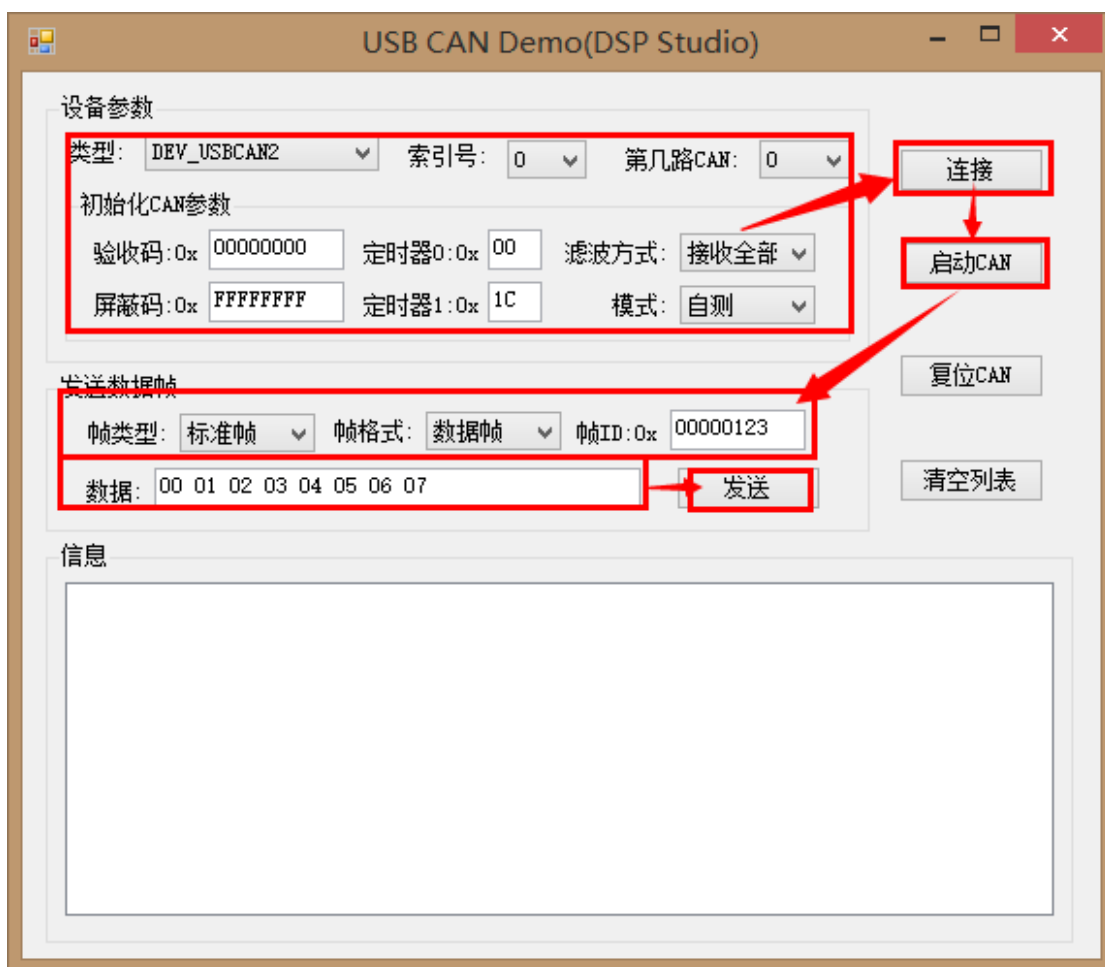
## 第二章 示例程序操作说明

### 1. C#\_Example

C#的例程包含两种环境：VS2005和VS2008，其操作界面是相同的。其编译生成的exe文件分别位于：C#\_Example\C#\_Example(VS2005)\WindowsApplication1\bin\Release和C#\_Example\C#\_Example(VS2008)\WindowsApplication1\bin\x86\Release下面，exe文件名称为USBCAN\_Demo.exe：

名称	修改日期	类型	大小
controlcan.dll	2014-04-12 2:42	应用程序扩展	29 KB
USBCAN_Demo.exe	2014-02-09 17:24	应用程序	22 KB

其界面截图如下：



其操作流程，首先在设备参数区选择设备类型，设备索引号，及设备通道号，并且配置好初始化CAN参数，然后点击【连接】以连接USB\_CAN设备，然后点击【启动】，设备就可做

数据收发操作了。设置好要发送的数据帧的格式等信息，点击【发送】按钮即可发送数据，数据的接收在点击了【启动】按钮后将自动进行，基本的操作信息及发送和接收数据等都会在【信息】列表中显示。








按下【复位CAN】按钮后，需要重新点击【启动】才能执行CAN的发送和接收操作。

更多内容请参见源代码及注释。

## 2. C++ Builder6\_Example

C++ Builder6下的示例，其编译生成的exe文件分别位于：C++

Builder6\_Example\Project1.exe:

 ControlCAN.dll	2014-04-12 2:42	应用程序扩展	29 KB
 ControlCAN.lib	2014-04-12 2:42	LIB 文件	9 KB
 Project1.bpr	2012-07-12 20:24	BPR 文件	5 KB
 Project1.cpp	2012-07-12 20:24	CPP 文件	2 KB
 Project1.exe	2012-09-13 20:27	应用程序	530 KB
 Project1.obj	2012-08-23 22:52	OBJ 文件	18 KB
 Project1.res	2012-07-12 20:24	RES 文件	1 KB

其界面截图如下：

The screenshot shows a Delphi application window titled 'Form1'. It contains the following controls:

- 设备参数 (Device Parameters):** Includes dropdowns for '设备类型' (Device Type) set to 'USB\_CAN2' and 'CAN通道' (CAN Channel) set to '通道1'. A '连接' (Connect) button is to the right.
- 初始化CAN参数 (Initialize CAN Parameters):** Includes text boxes for '验收码' (Checksum) '0x 00000000', '屏蔽码' (Mask) '0x FFFFFFFF', '定时器0' (Timer 0) '0x 00', and '定时器1' (Timer 1) '0x 14'. It also has a '滤波方式' (Filtering Method) dropdown set to '接收所有' (Receive All) and a '模式' (Mode) dropdown set to '正常模式' (Normal Mode).
- 发送数据帧 (Send Data Frame):** Includes dropdowns for '帧类型' (Frame Type) set to '标准帧' (Standard Frame) and '帧格式' (Frame Format) set to '数据帧' (Data Frame). It has a '帧ID' (Frame ID) text box set to '00000080' and a '数据' (Data) text box containing '01 02 03 04 05 06 07 08'. A '发送' (Send) button is to the right.
- Buttons:** '启动CAN' (Start CAN), '复位CAN' (Reset CAN), and '清空列表' (Clear List) buttons are located to the right of the '发送' button.
- 信息 (Information):** A large text area at the bottom for displaying logs or messages.

Red annotations highlight the '连接' button, the '启动CAN' button, and the '发送' button, with arrows indicating a sequence of operations.

本示例中默认使用设备索引号为0。

其操作流程，首先在设备参数区选择设备类型，设备通道号，并且配置好初始化CAN参数，然后点击【连接】以连接USB\_CAN设备，然后点击【启动CAN】，设备就可做数据收发操作了。设置好要发送的数据帧的格式等信息，点击【发送】按钮即可发送数据，数据的接收在点击了【启动CAN】按钮后将自动进行，基本的操作信息及发送和接收数据等都会在【信息】列表中显示。

按下【复位CAN】按钮后，需要重新点击【启动CAN】才能执行CAN的发送和接收操作。

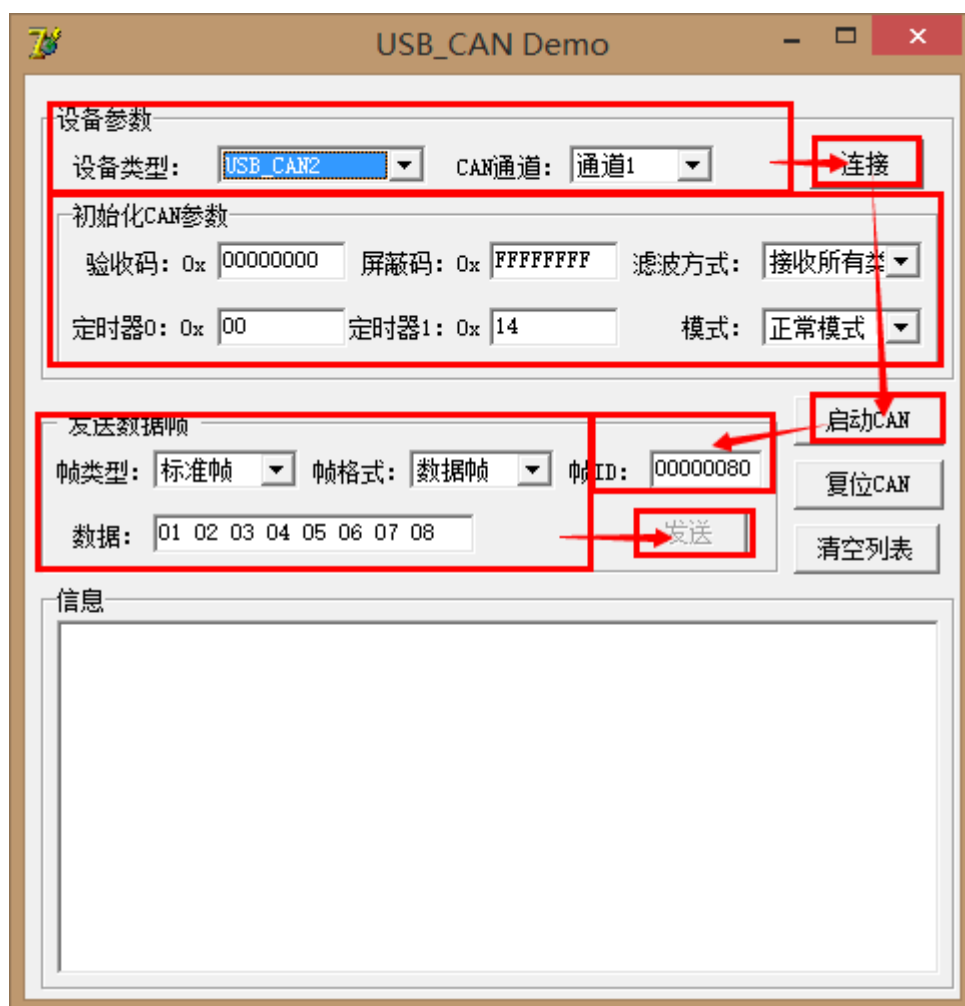
更多内容请参见源代码及注释。

### 3. delphi7\_Example

delphi7下的示例，其编译生成的exe文件分别位于：delphi7\_Example\Project1.exe：

名称	修改日期	类型	大小
ControlCAN.~pas	2012-09-04 19:24	~PAS 文件	8 KB
ControlCAN.dcu	2012-09-04 19:26	DCU 文件	5 KB
ControlCAN.dll	2014-04-12 2:42	应用程序扩展	29 KB
ControlCAN.pas	2012-09-04 19:25	PAS 文件	8 KB
Project1.cfg	2012-07-12 20:24	CFG 文件	1 KB
Project1.dof	2012-07-12 20:24	DOF 文件	2 KB
Project1.dpr	2012-07-12 20:24	DPR 文件	1 KB
<b>Project1.exe</b>	2012-09-13 20:29	应用程序	430 KB
Project1.res	2012-07-12 20:24	RES 文件	1 KB
Unit1.~ddp	2012-08-24 22:37	~DDP 文件	1 KB
Unit1.~dfm	2012-08-24 22:37	~DFM 文件	7 KB

其界面截图如下：



本示例中默认使用设备索引号为0。

其操作流程，首先在设备参数区选择设备类型，设备通道号，并且配置好初始化CAN参数，然后点击【连接】以连接USB\_CAN设备，然后点击【启动CAN】，设备就可做数据收发操作了。设置好要发送的数据帧的格式等信息，点击【发送】按钮即可发送数据，数据的接收

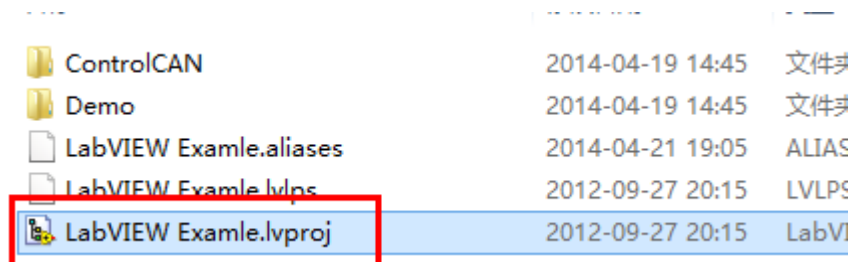
在点击了【启动CAN】按钮后将自动进行，基本的操作信息及发送和接收数据等都会在【信息】列表中显示。

按下【复位CAN】按钮后，需要重新点击【启动CAN】才能执行CAN的发送和接收操作。

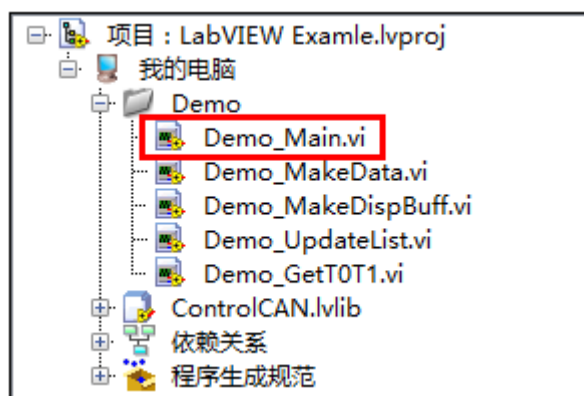
更多内容请参见源代码及注释。

## 4. LabVIEW Example(8.6)

LabVIEW Example(8.6)下的示例，是以LabVIEW工程的方式存放的，打开LabVIEW Example(8.6)\LabVIEW Examle.lvproj，

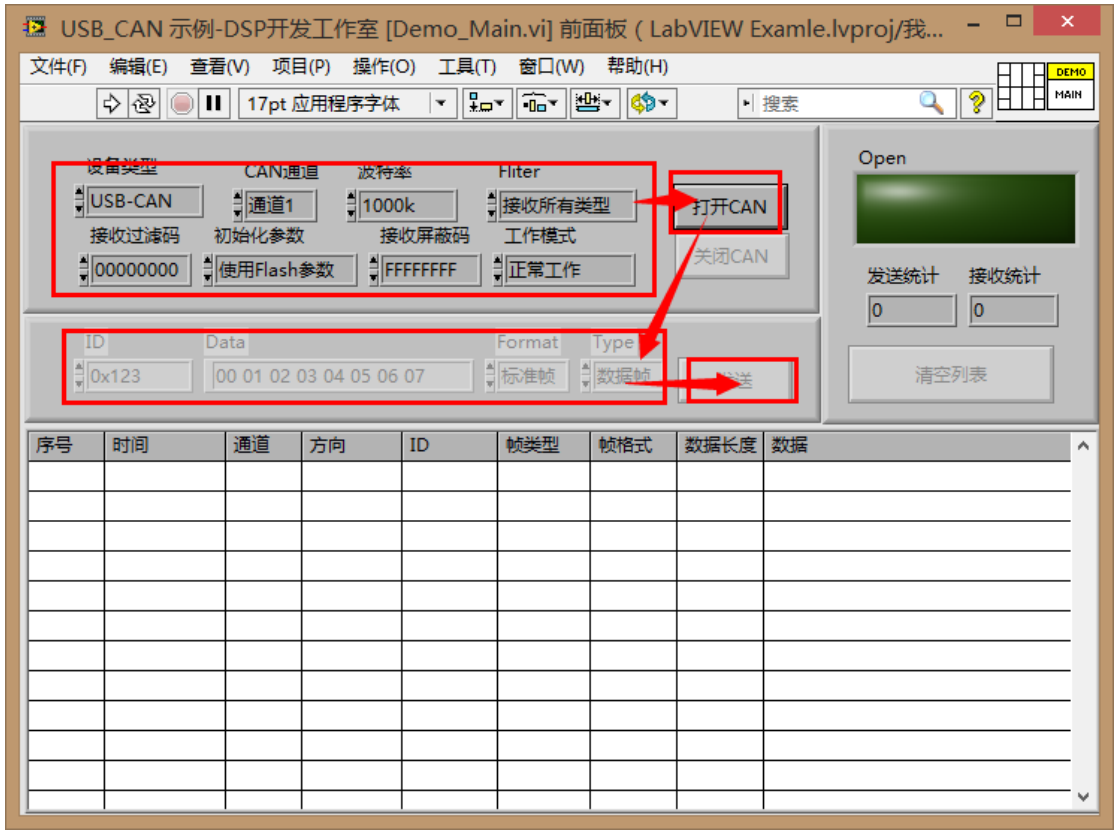


在 LabVIEW 工程中，打开 Demo\_main.vi



界面截图如下：





本示例中默认使用设备索引号为0。

其操作流程，首先在设备参数区选择设备类型，设备通道号，并且配置好初始化CAN参数，然后点击【打开CAN】以连接并启动USB\_CAN设备，。设置好要发送的数据帧的格式等信息，点击【发送】按钮即可发送数据，数据的接收在点击了【打开CAN】按钮后将自动进行，基本的操作信息及发送和接收数据等都会在列表中显示。

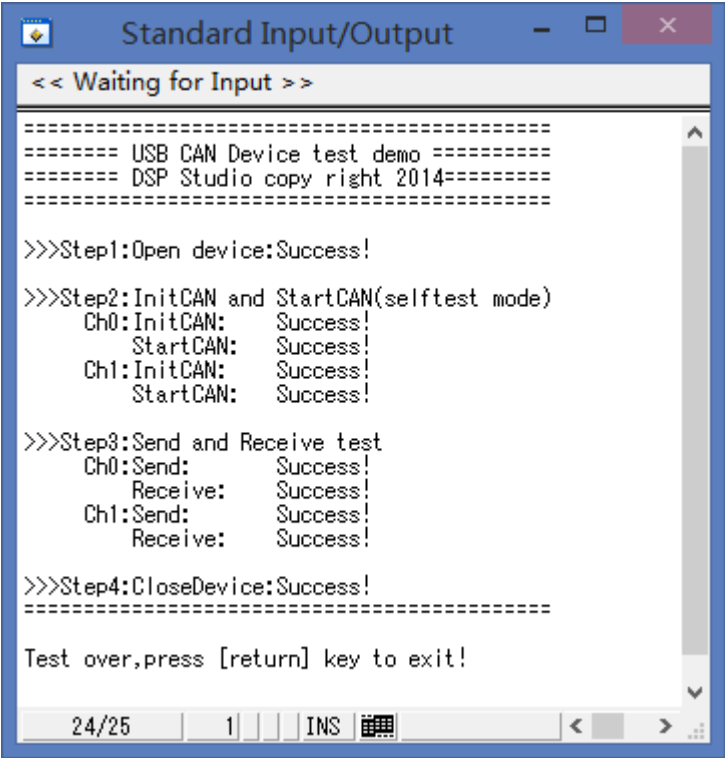
5. LabWindows CVI Example

LabWindows下的示例，打开LabWindows2013\_Example\TestCan下的test. prj，

	ControlCAN.dll	2014/5/3 17:58	应用程序扩展	30 KB
	ControlCAN.h	2014/5/3 17:58	C Header File	5 KB
	ControlCAN.lib	2014/5/3 17:58	LIB 文件	8 KB
	Test.c	2014/5/3 18:54	C Source File	4 KB
	test.cws	2014/5/3 18:56	LabWindows/CV...	5 KB
	test.exe	2014/5/3 18:55	应用程序	505 KB
	test.prj	2014/5/3 17:38	LabWindows/CV...	16 KB

Test. prj是CVI下的工程文件，打开该文件可在LabWindows环境下查看或编辑示例源代码。该示例实现了在CVI下的USB CAN设备的自测试。

其目录下的test. exe是已生成的可执行文件，运行效果如下图所示：

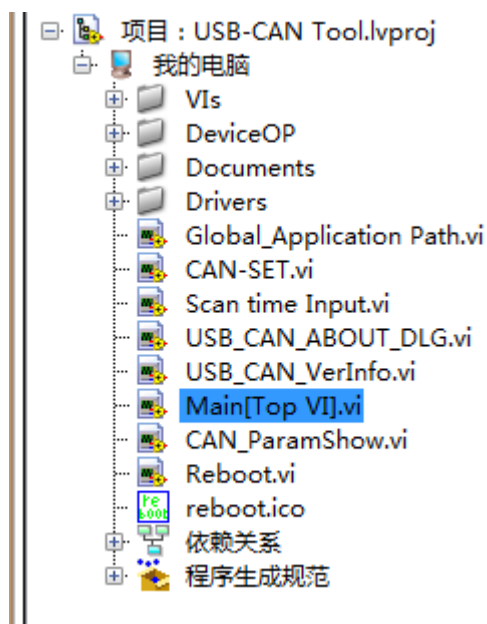


6. USB CAN Tool 源代码(LabVIEW 2011环境)

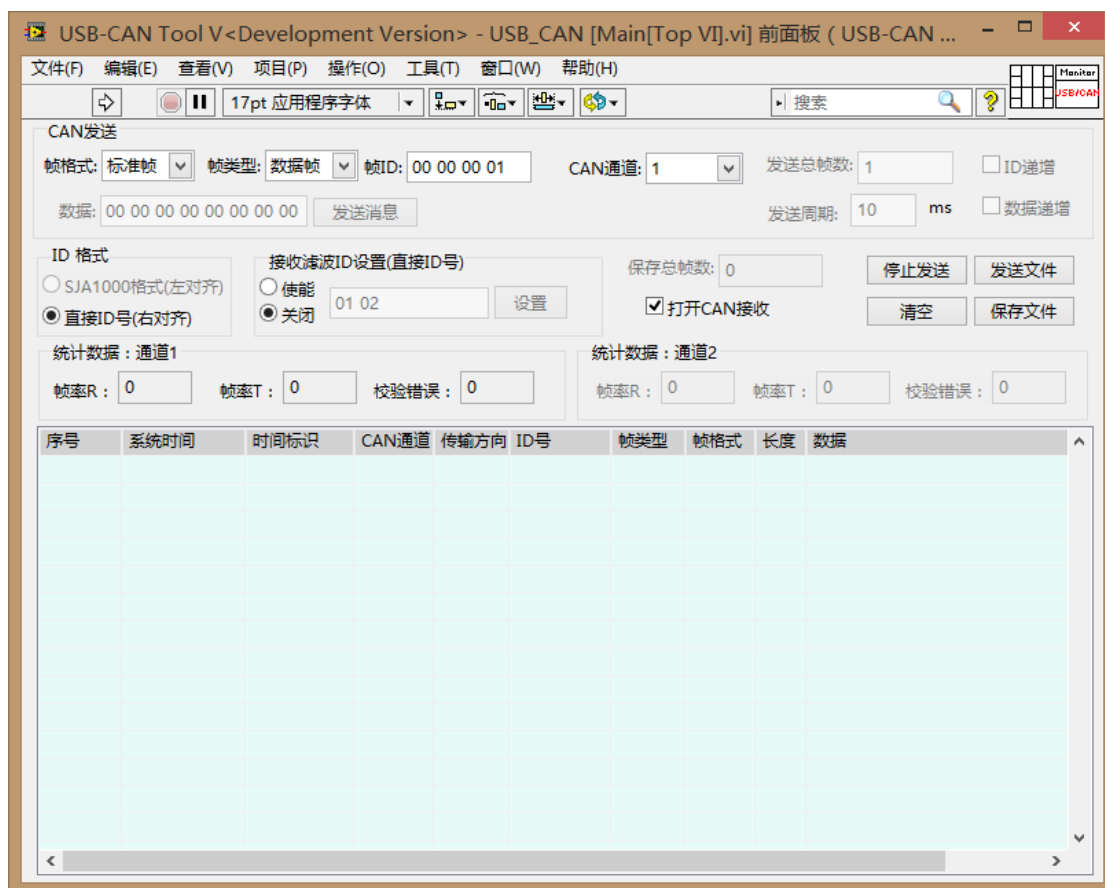
USB CAN Tool 源代码(LabVIEW 2011环境), 是随机自带的USB-CAN TOOL的源代码, 打开“USB CAN Tool 源代码(LabVIEW 2011环境)\USB-CAN Tool.lvproj”:

	2014-04-19 14:45	文件夹	
data	2014-04-19 14:45	文件夹	
DeviceOP	2014-04-19 14:45	文件夹	
Documents	2014-04-19 14:45	文件夹	
Drivers	2014-04-19 14:45	文件夹	
Vis	2014-04-19 14:45	文件夹	
CAN_ParamShow.vi	2012-09-13 21:40	LabVIEW Instru...	25 KB
CAN-SET.vi	2013-03-07 22:27	LabVIEW Instru...	92 KB
Global_Application Path.vi	2012-06-27 20:16	LabVIEW Instru...	9 KB
Main[Top VI].vi	2013-06-18 2:24	LabVIEW Instru...	420 KB
reboot.ico	2013-05-07 23:28	图标	6 KB
Reboot.vi	2013-05-08 0:43	LabVIEW Instru...	24 KB
Scan time Input.vi	2013-06-10 20:29	LabVIEW Instru...	13 KB
USB_CAN_ABOUT_DLG.vi	2012-09-13 21:40	LabVIEW Instru...	14 KB
USB_CAN_VerInfo.vi	2013-05-09 21:49	LabVIEW Instru...	6 KB
USB-CAN Tool.aliases	2014-04-21 19:14	ALIASES 文件	1 KB
USB-CAN Tool.lvps	2013-08-16 0:22	LVLPS 文件	6 KB
USB-CAN Tool.lvproj	2013-08-16 0:22	LabVIEW Project	39 KB

在LabVIEW工程中打开Main[Top VI].vi:



该vi就是USB CAN Tool的主界面vi，其操作方法请参见“软件使用说明书”。



## 7. VB.NET Example

VB.net下的示例，其编译生成的exe文件分别位于：VB.NET Example\USB\_CAN

Demo\bin\Release\USBCANDemo.exe:

ControlCAN.dll	2014-04-12 2:42	应用程
USBCANDemo.exe	2014-04-21 0:23	应用程

界面截图如下：



本示例中默认使用设备索引号为0。

其操作流程，首先在设备参数区选择设备类型，设备通道号，并且配置好初始化CAN参数，然后点击【连接设备】以连接USB\_CAN设备，然后点击【启动CAN】，设备就可做数据收发操作了。设置好要发送的数据帧的格式等信息，点击【发送】按钮即可发送数据，数据的接收在点击了【启动CAN】按钮后将自动进行，基本的操作信息及发送和接收数据等都会在【信息】列表中显示。

按下【复位CAN】按钮后，需要重新点击【启动CAN】才能执行CAN的发送和接收操作。

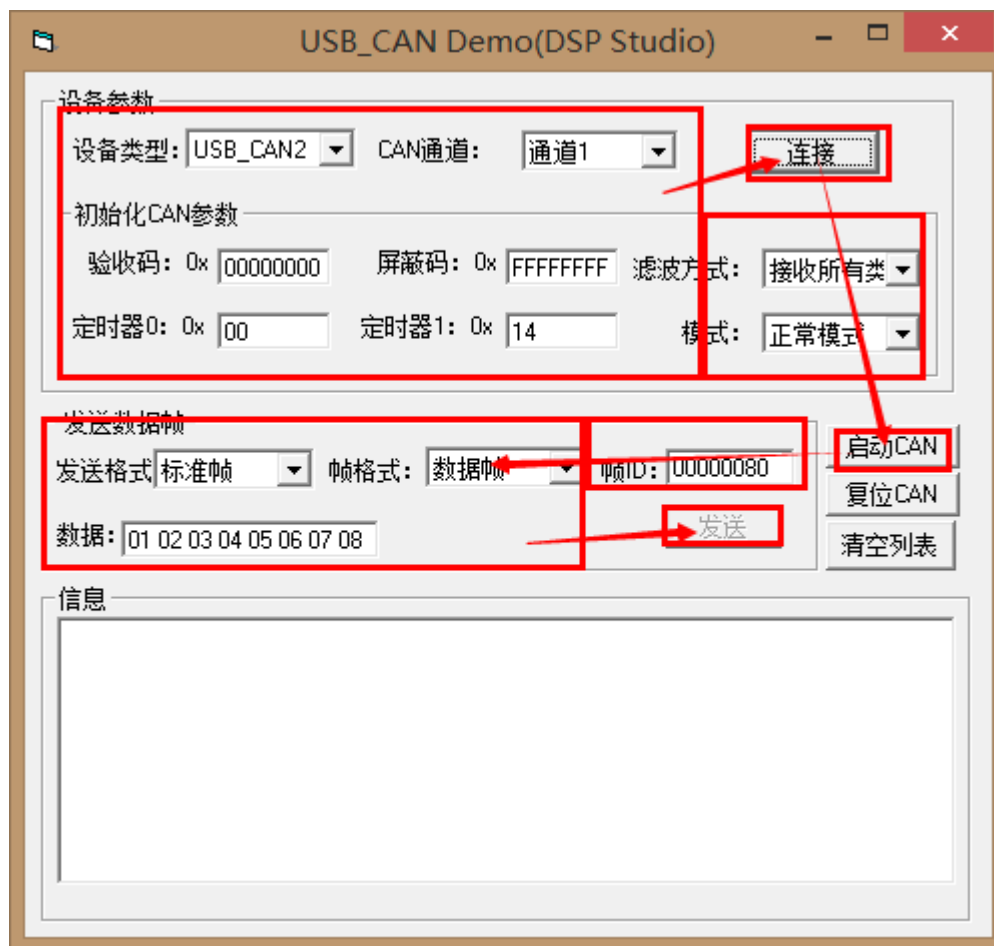
更多内容请参见源代码及注释。

## 8. VB\_Example

VB6.0下的示例，其编译生成的exe文件分别位于：VB\_Example\ 工程1.exe：

名称	修改日期
ControlCAN.bas	2012-09-13 20:1
ControlCAN.dll	2014-04-12 2:42
Form1.frm	2014-04-19 14:4
Form1.frx	2014-04-19 14:4
MSVBVM60111.DLL	2012-07-12 20:2
工程1.exe	2014-04-19 14:4
工程1.vbp	2014-04-19 14:4
工程1.vbw	2014-04-19 14:4

其界面截图如下：



本示例中默认使用设备索引号为0。

其操作流程，首先在设备参数区选择设备类型，设备通道号，并且配置好初始化CAN参数，然后点击【连接】以连接USB\_CAN设备，然后点击【启动CAN】，设备就可做数据收发操作了。设置好要发送的数据帧的格式等信息，点击【发送】按钮即可发送数据，数据的接收在点击了【启动CAN】按钮后将自动进行，基本的操作信息及发送和接收数据等都会在【信
















息】列表中显示。




















按下【复位CAN】按钮后，需要重新点击【启动CAN】才能执行CAN的发送和接收操作。

更多内容请参见源代码及注释。

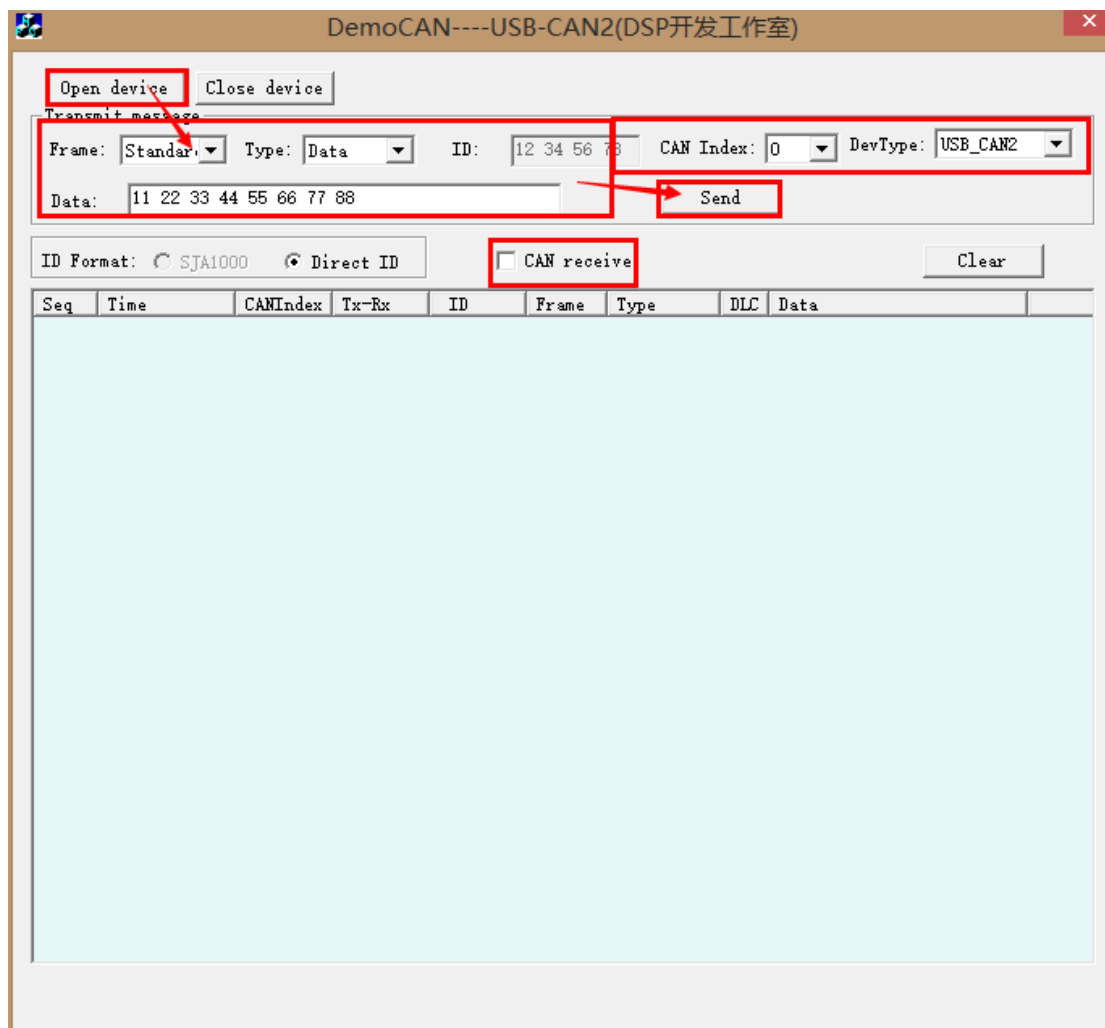
## 9. VC\_Example

VC的例程包含两种环境：VC6.0和VS2008，其操作界面是相同的。其编译生成的exe文件分别位于：VC\_Example\VC\_Example\_VC6.0\和VC\_Example\VC\_Example\_VS2008\Debug\下，文件名称为：USBCAN\_Demo.exe：

	DemoCAN.cpp	2007-12-05 20:45	CPP 文件	3 KB
	DemoCAN.dsp	2012-09-12 23:34	DSP 文件	5 KB
	DemoCAN.dsw	2007-12-05 20:45	DSW 文件	1 KB
	DemoCAN.h	2007-12-05 20:45	H 文件	2 KB
	DemoCAN.ncb	2014-04-19 14:44	NCB 文件	169 KB
	DemoCAN.opt	2014-04-19 14:44	OPT 文件	48 KB
	DemoCAN.plg	2014-04-19 14:44	PLG 文件	3 KB
	DemoCAN.rc	2012-08-23 20:11	RC 文件	8 KB
	DemoCANDlg.cpp	2013-10-08 0:35	CPP 文件	15 KB
	DemoCANDlg.h	2012-08-22 23:23	H 文件	2 KB
	ReadMe.txt	2007-12-05 20:45	文本文档	4 KB
	resource.h	2012-08-22 23:23	H 文件	2 KB
	StdAfx.cpp	2007-12-05 20:45	CPP 文件	1 KB
	StdAfx.h	2007-12-05 20:45	H 文件	2 KB
	USBCAN_Demo.exe	2014-04-19 14:44	应用程序	28 KB

 BuildLog.htm	2013-10-08 0:34
 ColoredListCtrl.obj	2013-10-08 0:34
 ColoredListCtrl.sbr	2013-10-08 0:34
 ControlCAN.dll	2014-04-12 2:42
 DemoCAN.bsc	2013-10-08 0:34
 DemoCAN.exe	2013-10-08 0:34
 DemoCAN.exe.embed.manifest	2013-10-08 0:34
 DemoCAN.exe.embed.manifest.res	2013-10-08 0:34
 DemoCAN.exe.intermediate.manifest	2013-10-08 0:34
 DemoCAN.ilc	2013-10-08 0:34
 DemoCAN.obj	2013-10-08 0:34
 DemoCAN.pdb	2013-10-08 0:34
 DemoCAN.res	2013-10-08 0:34
 DemoCAN.sbr	2013-10-08 0:34
 DemoCANDlg.obj	2013-10-08 0:34
 DemoCANDlg.sbr	2013-10-08 0:34
 mt.dep	2013-10-08 0:34
 StdAfx.obj	2013-10-08 0:34
 StdAfx.sbr	2013-10-08 0:34

界面截图如下：



本示例中启动参数使用1Mbps，标准帧，自测模式，当点击“Open device”时，即打开并启动CAN，勾选“CAN receive”来启动接收线程，点击“Send”发送界面配置的CAN数据帧。详情请参见源代码。

## 10. QT

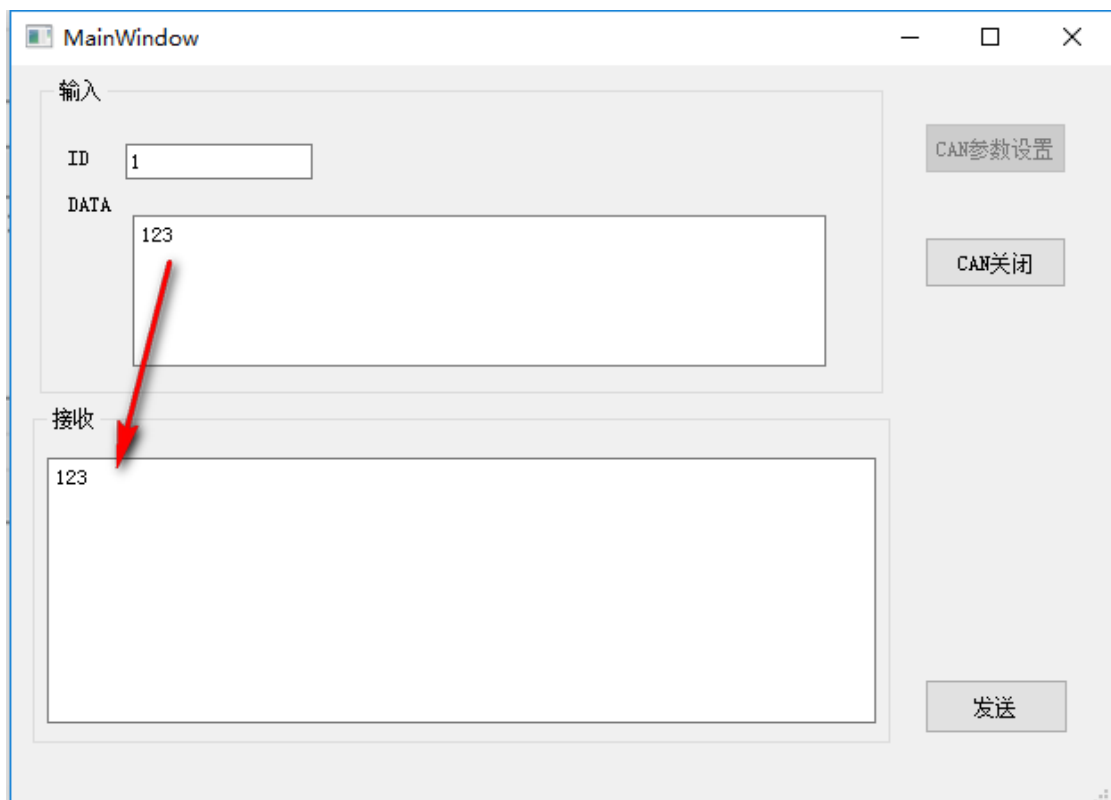
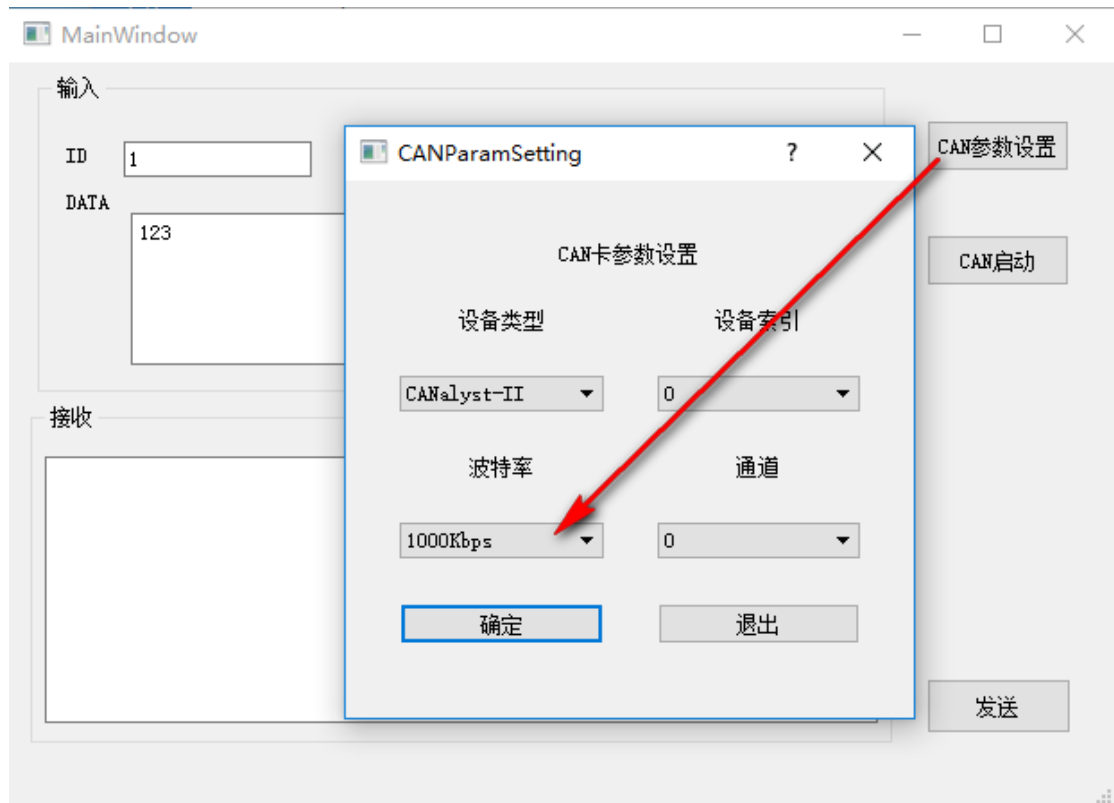
版本：QT5.9.3

QT的例程。其编译生成的exe文件分别位于：QT样例

\QT\_Example\QT\_Example\release\CAN.exe下，文件名称为：CAN.exe：

界面如下，首先，波特率要设置为1000Kbps，其它参数无关紧要，后台自动以相同的参数初始化两个通道。把CAN1与CAN2对接起来。点击“CAN启动”，ID随意输入，DATA中填入字符串，点击发送。CAN2可以读到对应的数据。详情请参见源代码。





## 11. python

版本: python3.8.0

python编译器分32位与64位。

32位编译器要用32位DLL，在目录：CAN分析仪资料\CAN分析仪资料20191001\二次开发库文件\ControlCAN.dll详情请参见源代码。

64位编译器要用64位DLL，在目录：CAN分析仪资料\CAN分析仪资料20191001\二次开发库文件\x64\ControlCAN.dll。

程序中有指定DLL目录，要把DLL拷贝到对应的目录下才能运行。

```
CanDLLName = 'D:\ControlCAN.dll' #把 DLL 放到对应的目录下
```

```
canDLL = windll.LoadLibrary('D:\ControlCAN.dll')
```

样例功能：两个通道同时初始化 125K，正常模式。CAN1 发送一帧数据到 CAN2 通道并显示。测试结果如下：

注：一定要按说明书把CAN1与CAN2对接。（参考 CAN分析仪资料\说明文档目录\《5. 插件1: USB-CAN总线适配器测试.pdf》说明书，接线。）



```
Python 3.8.0 Shell
File Edit Shell Debug Options Window Help
Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 14 2019, 19:37:50) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\ttc\Desktop\python64\python3.8.0.py =====
D:\ControlCAN.dll
调用 VCI_OpenDevice成功
调用 VCI_InitCAN1成功
调用 VCI_StartCAN1成功
调用 VCI_InitCAN2 成功
调用 VCI_StartCAN2 成功
CAN1通道发送成功
CAN2通道接收成功
ID:
1
DataLen:
8
Data:
[1, 2, 3, 4, 5, 6, 7, 8]
>>> |
```