文章编号:1671-4598(2016)04-0206-04 DOI:10.16526/j.cnki.11—4762/tp.2016.04.060 中图分类号:TP314 文献标识码:A

基于 Windows 下 Qt 与脉冲发生器的串口通讯实现

李鹏军，兰殿星，宁文斌，袁 攀，宋军伟
（许继电源公司 特种电源部，河南 许昌 461000）

摘要：串口通讯在工业领域有着重要的作用，大部分工业设备都使用串口通讯和上位机远程通讯；VC6.0 的简单易用在工业中经常被使用制作各种复杂的监控控制软件，但是 VC 的图形绘制、API 接口功能和所制作软件移植性存在很多不方便的地方，该实现方法成功使用了 VC 嵌套 Qt 的方法解决了该问题；以脉冲发生器仪器为下位机举例，在 Windows XP 环境下，使用 VC6.0 嵌套 Qt3.3，制作上位机 RS232 通讯控制图形软件；使用该软件通讯功能用户可以设置不同的脉冲发生器参数，输出不同的脉冲序列，该软件所使用的界面、线程、串口支持类全部由 Qt 支持，不使用 MFC 功能，最终完成软件编写；结果表明，在很低的电脑配置中，Windows 环境下，使用 VC 嵌套 Qt 的方式也能快速的设计出一款界面完整、功能齐全、稳定性高、移植性强的工业控制软件。

关键词：嵌套；Qt；Vc；QextSerialPort 类；线程；串口编程

Implementation of Serial Communication to Pulse Generator Based on Qt in Windows Environment

Li Pengjun，Lan Dianxing，Ning Wenbing，Yuan Pan，Song Junwei
（XJ Power CO.，Xuchang 461000，China）

Abstract：Serial communication plays important role in industrial area，most of industrial equipment using serial communication connect to PC terminal. VC6.0 is easy to use in industry and often used to monitor variety complex software production，but VC has many inconvenient places in graphics rendering，API interface function and software portability. Implementation method successfully used VC nesting Qt way to solve this problem. Pulse generator equipment as lower-computer，for example，use VC6.0 nested Qt3.3，making PC graphics software by RS232 communication in Windows XP。Using communication users can set different parameters to let pulse generator outputs various pulse sequences. Interface ，thread and serial support classes of Software all supported by Qt not using MFC. Implementation finally completed software development. Results showed that low computer configuration，in Windows environment，using VC nested Qt way can quickly design a complete interface，full-featured，high stability，easy portability industrial control software.

Keywords：nest；Qt；Vc；Class QextSerialPort；thread；serial communication

0 引言

串口通信在工业中使用范围很广，有很多机器及仪器都大量使用串口通讯的方式，所以作为上位机和下位机之间的通讯[1]，串口通讯的功能是在软件的使用中是一个重要的部分。同时在工业软件的设计中，由于大部分设计软件不能应用于 Windows XP 以上的系统，VC6.0 和 Qt3.3 两个具有内存占用小，编程方便的特点，能完美应用于 XP 系统。但是 VC6.0 的图形编辑能力较差，不能较好地实现人性化人机界面，也不具备很强的移植性[2]。所以这里提出了 VC 嵌套 Qt 的概念，Qt 是一个多平台的 C++图形用户程序框架，因为其面向对象、易扩展、可实现组件编程等特点，已经成为热门的编程软件。由于 Qt 同样采用 C++语言，方便了 VC6.0 和 Qt 的嵌套使用，这样设计的软件程序不是以 MFC 为基础，减少了程序的出错率。该实现方法结合开发实例对带有图形界面的串口通信程序设计过程进行了详细的阐述。同时以脉冲发生器（575 Series Pulse）SCPI 语言和上位机通讯为例介绍。

1 实现方法说明

软件采用 PC 端为上位机[3]，脉冲发生器为下位机的，串口互联 SCPI 语言通讯的方法。软件编译环境为 VC6.0，软件设计实现过程如图 1 所示。

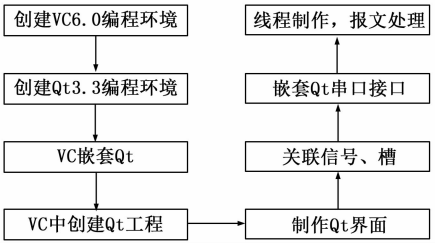

图 1 实现过程图

软件调用 QextSerialPort 对象，并使用线程实现串口数据的接收，使用定时器实现串口数据的发送。发送数据和接收数据全部使用堆栈式列表数据缓冲区，保证接收数据不会出现漏针、发送数据严格按照固定间隔时间发送，避免下位机通讯过快死机问题[4-5]。软件实现方法如图 2 所示。

2 VC6.0 建立 Qt 工程

这里直接使用 VC 嵌套 Qt 后产生的工具栏生成一个 Dialog 的工程，工程名字命名为“PulseGen”；工程会自动生成“pulsegendialogimpl”为名的执行头文件和源文件，以后所有的程序都需要写到该执行文件中，可以看到这里可以直接定义 Qt 信号和槽，头文件代码如下：

收稿日期：2015-10-23； 修回日期：2015-11-29。

作者简介：李鹏军（1985-），男，河南焦作人，大专，工程师，主要从事脉冲功率方向的研究。

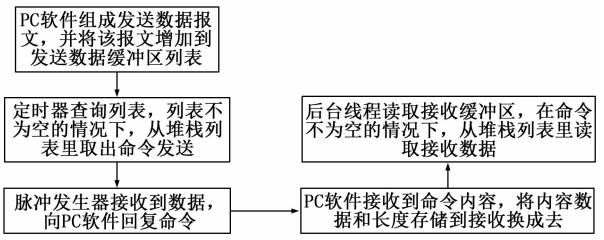

图 2　软件实现方法图

```
include "pulsegendialog.h"
class PulseGenDialogImpl : public PulseGenDialog
{
Q_OBJECT
public:
PulseGenDialogImpl( QWidget * parent = 0,const char * name = 0,bool modal = FALSE,WFlags f = 0 );
};
```

源文件代码如下：

```
include "pulsegendialogimpl.h"
PulseGenDialogImpl::PulseGenDialogImpl ( QWidget * parent, const char * name,bool modal,WFlags f )
: PulseGenDialog( parent,name,modal,f )
{// Add your code}
```

使用 VC 编译后，会产生“PulseGen.pro”文件，使用“Qt Designer”可以打开该文件添加项目文件、修改界面布局、增加界面空间等功能，修改完成后，直接使用 VC 编译即可生产 exe 文件。

3　Qt 界面制作

使用 Qt Designer，打开 pro 工程而文件，可以方便的设计各种复杂界面，先根据功能大致做好界面，然后添加 QextSerialPort 串口接口文件。Qt Designer 可以灵活的设置界面控件颜色及字体，如图 3 所示。

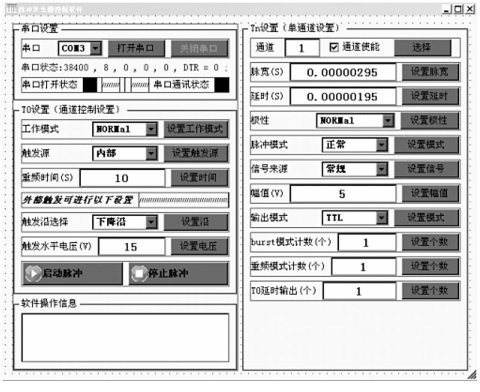

图 3　界面制作示意图

这个界面主要提供了串口设置功能、T_0 的触发功能及每个触发通道的参数设置。

4　设备类线程创建、代码编写

使用 Qt Designer 增加 deviceCe 类文件，该类继承 QThread，该线程负责串口的数据发送和接收处理功能[6]。

在 deviceCe 头文件类中增加串口的对象声明，为了方便串口的数据发送和接收，同时定义串口发送、接收的列表缓冲区和结构体，在发送和接收数据的时候，将发送字节和接收字节增加到缓冲区列表中，处理的时候会从第一行往下依次处理数据，处理一次数据删除一行数据，这样不会造成缓冲区数据的积累和出错，为了保证串口数据接收稳定，在每次发送完数据 50 ms 以后读取下一次数据，并强制发送数据间隔时间为 500 ms，这样串口数据接收非常稳定。deviceCe 头文件添加代码如下：

```
struct cmdStruct //接收命令的结构体
{
unsigned char m_buff[ 300 ];
int m_length;
};
struct cmdSend //发送命令的结构体
{
char m_buff[ 300 ];
int m_length;
};
class deviceCe : public QThread
//继承线程
{
public:
QValueList<cmdStruct> m_listCmd ;
//接受命令列表缓存
QValueList<cmdSend> m_listSend;
//发送命令列表缓存
void AppendChar( const char * cmdChar ,short lengthCmd );//发送命令缓存到列表中
void OpenCom( QString theComStr);
//打开串口
void CloseCom(); //关闭串口
void readMyCom( int theLength);
//读取串口数据
void SendMessage(); //发送数据
void GetLength(); //获取接收数据长度
private:
QTime reTime; //接收数据的强制时间
Win_QextSerialPort * myCom ; //串口对象声明
};
```

源代码中添加打开串口命令代码：

```
//存放串口参数结构体
PortSettings myComSetting = {FLOW_OFF,PAR_NONE,DATA_8,STOP_1,BAUD38400,0,500}; //定义串口对象,并传递参数,并初始化
myCom = new Win_QextSerialPort(theComStr,myComSetting);
//以可读写方式打开串口
if(myCom->open(IO_ReadWrite))
{…}
```

读取串口数据代码如下，GetLength 函数调用：

```
char theReChar[300];
myCom->readBlock( theReChar ,theLength); //按照个数读取数
cmdStruct m_scmd ;
memcpy( m_scmd.m_buff ,theReChar,cmdLength);
m_scmd.m_length = cmdLength ;
m_listCmd.append( m_scmd ) ; //接收数据到缓冲列表中。
```

发送数据添加到缓冲列表代码如下(由界面调用)：

```
void deviceCe::AppendChar( const char * cmdChar ,short lengthCmd )
{
cmdSend m_setValueStuct ;
memcpy( m_setValueStuct.m_buff ,cmdChar,lengthCmd );
m_setValueStuct.m_length = lengthCmd ;
m_listSend.append( m_setValueStuct );
}
```

在串口存在数据时，读取数据，由 reTime 定时器调用，代码如下：

```
void deviceCe::GetLength()
{
int m_length = myCom->bytesWaiting();
if( m_length > 1)
{
readMyCom( m_length );
}
}
```

线程函数编程如下：

发送数据函数，该函数由界面定时器定时调用：

```
void deviceCe::SendMessage()
{
QString msgStr ;
cmdSend sendStuct ;
if ( ! m_listSend.isEmpty() )
{
sendStuct = m_listSend.first() ;
myCom ->writeBlock( sendStuct.m_buff,sendStuct.m_length );
m_listSend.pop_front();
reTime.restart(); //强制时间开启,有发送命令的时候,接收数据
}
else
reTime.restart();
}
```

线程编程如下：

```
void deviceCe::run()
{
cmdStruct reCmdDo;
while(m_stop == false)
{
if ( reTime.elapsed() > 50 )
{
GetLength();
}
if ( ! m_listCmd.isEmpty()) //接受判断
{
reCmdDo = m_listCmd.first();
m_listCmd.pop_front();
}
}
```

5 界面执行文件编程

界面编程主要是增加信号槽、关联 deviceCe 对象、界面刷新编程几个内容。

按钮、定时器槽函数相应代码定义如下：

```
class PulseGenDialogImpl : public PulseGenDialog
{
Q_OBJECT
public:
PulseGenDialogImpl( QWidget * parent = 0,const char * name = 0,bool modal = FALSE,WFlags f = 0 );
protected:
QTimer * m_pTimerProcessUI;//界面刷新时间
QTimer * m_pTimerSendMsg; //发送命令时间
deviceCe * p_device ; //设备类对象
//根据不同的功能组成 SCPI 命令,一般都是:PULSE 开头的
void btSetCmd( QString m_msg );
protected slots:
void processUI(); //界面刷新
void sendMsg(); //发送数据
void btOpenCom(); //打开串口
void btCloseCom(); //关闭串口
/* 其它的按钮相应 */
}
```

脉冲发生器使用 SCPI 命令，下面函数负责组成 SCPI 命令，发送函数只需要填入内容即可，如下面代码所示：

```
void PulseGenDialogImpl::btSetCmd( QString m_msg )
{
const char * theSendBuff ;
//组成字符串,并且末尾增加换行符
QString strBy = tr(":PULSE%1<cr><lf>/n").arg( m_msg );
//发送 SCPI 命令
theSendBuff = strBy.latin1();
p_device->AppendChar( theSendBuff ,strlen( theSendBuff ) );
}
```

比如需要通道 1 设置延时，可以直接使用代码：

```
int chID = textchID->text().toInt(); //通道号
QString str ;
double theWid = textDelay->text().toDouble() ;
str = tr("%1:DELAY %2").arg( chID).arg( theWid );
btSetCmd( str ) ;
```

在这里，pTimerProcessUI 关联界面相应，pTimerSendMsg 关联发送函数，如下代码：

```
connect(m_pTimerProcessUI, SIGNAL(timeout()), this, SLOT(processUI())); connect(m_pTimerSendMsg, SIGNAL(timeout()), this,SLOT(sendMsg()));
m_pTimerProcessUI->start(400); m_pTimerSendMsg->start(500) ;
```

sendMsg 负责按照 500 ms 发送一次数据，代码如下：

```
void PulseGenDialogImpl::sendMsg() //定时发送数据
{  p_device->SendMessage();}
```

6 软件功能验证

试验方法通过界面设置 T0 参数如图 4 所示。脉冲发生器回复串口数据如图 5 所示。实际脉冲发生器输出 10 μs 周期脉冲，如图 6 所示。

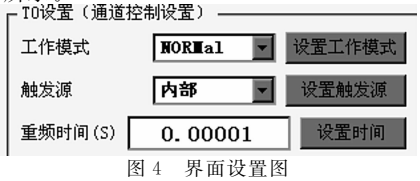

图 4 界面设置图

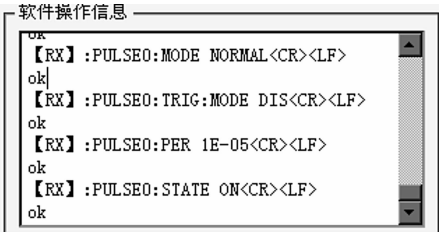

图 5　回复串口数据图

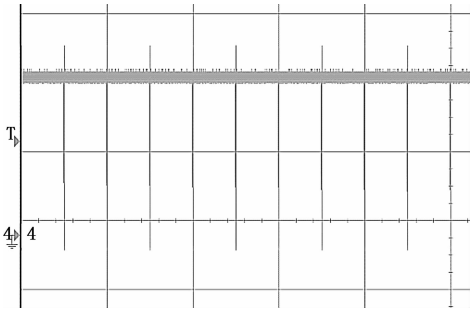

图 6　输出脉冲图（时间：20 μs 一格）

7　结果与分析

通过上位机软件串口通讯功能成功的设置了脉冲发生器的重频时间，并且通过示波器成功监视到符合要求的波形，脉冲发生器通过回复“OK”命令表示命令执行成功，完善软件其它通道参数设置通信内容，实现了同步机全部参数设置功能。

8　结论

该实现方法通过 Qt 与 VC6.0 的结合方式，完美的解决了 VC6.0 的图形编辑能力的弱点，集成了线程、接口，这样编程的方式大大提高了在工业软件监控软件中的开发效率。

参考文献：

[1] 佘　立，李志强．基于 Qt 的多串口多协议数据接收解析软件的设计与实现 [J]．电子质量，2011，01：34－36.

[2] 张新村，严　殊．基于 ARM 的 Linux 系统下 Qt 串口助手的设计 [J]．软件导刊，2011，08：64－66.

[3] 孟繁磊，穆丽红，王小曼．基于 linux 和 Qt 的串口通信调试器调的设计及应用 [J]．科技创新导报，2011，32：24.

[4] 严　贤，韩秀玲．基于 Qt 的串口通信应用研究与实现 [J]．微计算机信息，2012，08：114－116.

[5] 陈旭红，高文学．Qt/Embedded 串口类的设计及应用 [J]．湖北汽车工业学院学报，2010，04：51－53，58.

[6] 胡军锋，宋文杰，尤泽萌．Qt 中基于线程串口采集方式的实现 [J]．科技视界，2014，28：5，28.

（上接第 205 页）

文算法在识别准确度和速度方面都超过贝叶斯分类器和 BP 神经网络算法。

表 2　3 种算法识别结果比较

采用算法	正确数	错判数	准确率/%	计算时间/s
贝叶斯分类器	195	25	88.64	37.90
BP 神经网络	202	18	91.82	26.54
本文算法	207	13	94.09	20.40

6　结束语

本文采用构建 SVM 分类器模型的方法识别室内火灾。首先采用背景差分法进行火焰图像运动区域的检测，在 *YCrCb* 颜色空间分割出可疑火焰区域，然后进行火灾火焰图像特征提取，并将特征向量输入 SVM 训练，并用训练完成的 SVM 模型对火焰及疑似火焰干扰物作分类识别。为了提高 SVM 模型火焰识别的准确率，本文提出一种带变异操作和非线性动态调整惯性权值的 PSO 算法优化 SVM 的两个参数。仿真结果表明，本文改进的 PSO 算法进行 SVM 参数寻优后的分类器模型，在火焰识别准确率和速度方面都有很大的提升，并能有效排除室内常见干扰的影响，降低了火灾探测的误报率。

参考文献：

[1] 王　娜．火灾探测的模糊神经网络数据融合算法研究 [J]．燕山大学学报，2008，32（2）：120－123.

[2] 张　兢．基于多传感器数据融合的智能火灾预警系统 [J]．计算机工程与应用，2006，6：206－208.

[3] 王　菲．基于火焰识别的早期火灾探测技术研究 [D]．广州：华南理工大学，2013.

[4] Cheng C X. One fire detection method using neural networks [J]. Tsinghua Science and Technology，2011，16（1）：31－35.

[5] 谢荣全，徐志胜．基于 BP 神经网络在火灾图像探测技术中的应用 [J]．铁道科学与工程学报，2014，11（3）：140－145.

[6] Truong T X. Fire flame detection in video sequences using multi-stage pattern recognition techniques [J]. Engineering Applications of Artificial Intelligence，2012，25（7）：1365－1372.

[7] Ko，Byoung Chul，Fire detection based on vision sensor and support vector machines [J]. Fire Safety Journal，2009，44（3）：322－329.

[8] 王媛彬．基于特征融合的图像型火灾探测方法 [J]．计算机工程，2011，37（19）：166－168.

[9] 马宗方，程咏梅．基于快速支持向量机的图像型火灾探测算法[J]．计算机应用研究，2010，27（10）：3985－3987.

[10] 付　燕，聂亚娜．PSO-SVM 算法在肝脏 B 超图像识别中的应用 [J]．计算机测量与控制，2012，20（9）：2491－2500.

[11] Chen Yongqi. LS _ SVM Parameters Selection Based on Hybrid Complex Particle Swarm Optimization [J]. Energy Procedia，2012，17（A）：706－710.

[12] Hui ling Chena. Towards an optimal support vector machine classifier using a parallel particle swarm optimization strategy [J]. Applied Mathematics and Computation，2014，239：180－197.

[13] 邬月春．基于自适应变异粒子群算法的物流配送路径优化 [J]．兰州交通大学学报，2012，31（1）：114－117.

[14] 高　磊．一种动态非线性改变惯性权的自适应粒子群优化算法 [J]．科学技术与工程，2011，11（17）：3984－3988.