1. 先把串口通信类添加到项目中，总共六个文件qextserialbase.h qextserialbase.cpp

qextserialport.h qextserialport.cpp win\_qextserialport.h win\_qextserialport.cpp文件

2.槽函数connCom，进行连接串口

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*定义槽函数连接串口\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void MainWindow::connCom()

{

QString portName=ui->comboBoxPort->currentText();

myCom=new Win\_QextSerialPort(portName,QextSerialBase::*EventDriven*);//定义串口对象，并传递参数对其进行初始化

if(myCom->isOpen()) //如果串口已经打开，返回

{

return;

}

else

{

ok=myCom->*open*(QIODevice::*ReadWrite*); //以可读可写方式打开串口

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*设置比特率\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if(ui->comboBoxBaudRate->currentText()==tr("9600"))

{

myCom->*setBaudRate*(*BAUD9600*);

}

else if(ui->comboBoxBaudRate->currentText()==tr("115200"))

{

myCom->*setBaudRate*(*BAUD115200*);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*设置数据位\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if(ui->comboBoxData->currentText()==tr("8"))

{

myCom->*setDataBits*(*DATA\_8*);

}

else if(ui->comboBoxData->currentText()==tr("7"))

{

myCom->*setDataBits*(*DATA\_7*);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*设置奇偶校验位\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if(ui->comboBoxCheck->currentText()==tr("无"))

{

myCom->*setParity*(*PAR\_NONE*);

}

else if(ui->comboBoxCheck->currentText()==tr("奇"))

{

myCom->*setParity*(*PAR\_ODD*);

}

else if(ui->comboBoxCheck->currentText()==tr("偶"))

{

myCom->*setParity*(*PAR\_EVEN*);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*设置停止位\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

if(ui->comboBoxStop->currentText()==tr("1"))

{

myCom->*setStopBits*(*STOP\_1*);

}

else if(ui->comboBoxStop->currentText()==tr("2"))

{

myCom->*setStopBits*(*STOP\_2*);

}

this->timer=new QTimer(this); //定义一个定时器

connect(timer,SIGNAL(timeout()),this,SLOT(timeUpdate())); //设置

timer->start(300); //0.2秒调用一次

}

if(!ok)

{

QMessageBox::warning(this,"系统提示","打开失败");

return;

}

}

3.读取串口数据函数readCom

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*定义槽函数读取串口数据\*\*\*\*\*\*\*\*\*\*\*/

void MainWindow::readCom()

{

QByteArray temp=myCom->readAll(); //读取串口数据

qDebug()<<temp.toHex();

listFloat.clear();

for(int i=0;i<temp.size();i+=14)

{

if(temp[i]!=0XAA) //判断如果传出来的数据第一个字节不是aa,为不完整数据，i++

{

i++;

}

else if(temp[i]==0XAA&&temp[i+1]==0XAA&&temp[i+2]==0X0C&&temp[i+3]==0X07&&temp[i+12]==0X55&&temp[i+13]==0X55) //判断出障碍高度的信息

{

listFloat<<((BYTE)temp[i+6])\*16\*16+(BYTE)temp[i+7]; //将信息提取出来

}

}

ui->lcdNumberX->display(0); //设置横坐标

if(listFloat.size()<=2) //如果小于两帧，就直接把上次的数据给控件显示

{

ui->ThermoHeigh->*setValue*(remeberData+0.01); //显示到控件上

return;

}

float fTemp=MiddlevalueFilter(); //中值滤波算法

ui->lcdNumberY->display(fTemp); //设置纵坐标

ui->ThermoHeigh->*setValue*(fTemp); //显示到控件上

ui->textEditRead->clear(); //清空信息

QString strTemp;

strTemp.prepend(temp.toHex());

QString str;

for(int i=0;i<strTemp.size();i+=28)

{

str=strTemp.mid(i,28); //每28个字符为一行

str.append("\r\n"); //加入换行符

ui->textEditRead->insertPlainText(str); //写入信息

}

}

4.调用定时器，实时刷新数据

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*实时刷新数据的定时器\*\*\*\*\*\*\*\*\*\*\*\*\*/

void MainWindow::timeUpdate()

{

if(ok) //如果串口已经打开，则读取数据

{

readCom();

}

else

{

return;

}

}

5.中值平均滤波算法

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*中位值平均滤波算法\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

float MainWindow::MiddlevalueFilter()

{

float temp=0,sum=0;

int N=listFloat.size();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*进行冒泡排序\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

for (BYTE j=0;j<N-1;j++)

{

for (BYTE i=0;i<N-j-1;i++)

{

if ( listFloat[i]>listFloat[i+1] )

{

temp = listFloat[i];

listFloat[i] = listFloat[i+1];

listFloat[i+1] = temp;

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*去掉一个最大值,一个最小值\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

for(BYTE count=1;count<N-1;count++)

{

sum += listFloat[count];

}

remeberData=(sum/(N-2))\*0.01; //将最终的到的均值赋给记忆数据 return remeberData; //返回

}

6.将读取到的信息保存到文件中

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*保存文本信息\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void MainWindow::saveFile()

{

if(!ok)

{

QString fileName=QFileDialog::getSaveFileName(this,tr("保存"),QDir::currentPath());

if(fileName.isEmpty())

{

QMessageBox::warning(this,tr("系统提示"),tr("请输入文件名"));

return;

}

QFile\*file=new QFile;

file->setFileName(fileName);

bool fileOk=file->*open*(QIODevice::*WriteOnly*|QFile::*Text*); //以只写方式打开

if(fileOk)

{

QTextStream out(file);

QString strTemp=ui->textEditRead->toPlainText();

QString str;

for(int i=0;i<strTemp.size();i+=30)

{

str=strTemp.mid(i,30); //每28个字符为一行

out<<str; //写入文件

}

file->*close*();

delete file;

}

else

{

QMessageBox::warning(this,tr("系统提示"),tr("保存失败"));

}

}

else

{

QMessageBox::warning(this,tr("系统提示"),tr("请关闭串口后再保存"));

}

}