使用QT开发客户端界面，在客户端上实现按钮控制和按键控制小车，以及在客户端上显示摄像头传输的视频信息。

#include "dialog.h"

#include "ui\_dialog.h"

Dialog::Dialog(QTcpSocket \*cmd,QWidget \*parent) :

QDialog(parent),

ui(new Ui::Dialog)

{

ui->setupUi(this);

skt\_cmd=cmd;

qDebug()<<"accept............";

ui->radar->setScaledContents(true);

QMovie \*movie = new QMovie(":/images/2.gif");

ui->radar->setMovie(movie);

movie->start();

ui->camera->setPixmap(QPixmap(":/images/b.jpg"));

ui->camera->setScaledContents(true);

}

Dialog::~*Dialog*()

{

delete ui;

}

void Dialog::on\_pushButton\_5\_clicked()

{

CMD\_ROLL\_FORWARD = "+";

skt\_cmd->write(CMD\_ROLL\_FORWARD,CMD\_SIZE);

qDebug()<<CMD\_ROLL\_FORWARD;

}

void Dialog::on\_pushButton\_6\_clicked()

{

CMD\_ROLL\_FORWARD = "-";

skt\_cmd->write(CMD\_ROLL\_FORWARD,CMD\_SIZE);

qDebug()<<CMD\_ROLL\_FORWARD;

}

void Dialog::on\_pushButton\_7\_clicked()

{

CMD\_ROLL\_FORWARD = "w";

skt\_cmd->write(CMD\_ROLL\_FORWARD,CMD\_SIZE);

qDebug()<<CMD\_ROLL\_FORWARD;

}

void Dialog::on\_pushButton\_8\_clicked()

{

CMD\_ROLL\_FORWARD = "x";

skt\_cmd->write(CMD\_ROLL\_FORWARD,CMD\_SIZE);

qDebug()<<CMD\_ROLL\_FORWARD;

}

void Dialog::on\_go\_pressed()

{

CMD\_ROLL\_FORWARD = "w";

skt\_cmd->write(CMD\_ROLL\_FORWARD,CMD\_SIZE);

qDebug()<<CMD\_ROLL\_FORWARD;

}

void Dialog::on\_go\_released()

{

CMD\_ROLL\_FORWARD = "x";

skt\_cmd->write(CMD\_ROLL\_FORWARD,CMD\_SIZE);

qDebug()<<CMD\_ROLL\_FORWARD;

}

void Dialog::*keyPressEvent*(QKeyEvent \* event)

{

switch(event->key())

{

case Qt::Key\_W:

CMD\_ROLL\_FORWARD = "w";

break;

case Qt::Key\_A:

CMD\_ROLL\_FORWARD = "a";

break;

case Qt::Key\_Q:

CMD\_ROLL\_FORWARD = "q";

break;

case Qt::Key\_E:

CMD\_ROLL\_FORWARD = "e";

break;

}

skt\_cmd->write(CMD\_ROLL\_FORWARD,CMD\_SIZE);

qDebug()<<CMD\_ROLL\_FORWARD;

}

void Dialog::*keyReleaseEvent*(QKeyEvent \*event)

{

char CMD\_ROLL\_FORWARD[1];

switch(event->key())

{

case Qt::Key\_W:

case Qt::Key\_A:

case Qt::Key\_Q:

case Qt::Key\_E:

CMD\_ROLL\_FORWARD[0] = 'x';

skt\_cmd->write(CMD\_ROLL\_FORWARD,CMD\_SIZE);

qDebug()<<CMD\_ROLL\_FORWARD;

break;

}

}

void Dialog::on\_pushButton\_3\_pressed()

{

CMD\_ROLL\_FORWARD = "q";

skt\_cmd->write(CMD\_ROLL\_FORWARD,CMD\_SIZE);

qDebug()<<CMD\_ROLL\_FORWARD;

}

void Dialog::on\_pushButton\_3\_released()

{

CMD\_ROLL\_FORWARD = "x";

skt\_cmd->write(CMD\_ROLL\_FORWARD,CMD\_SIZE);

qDebug()<<CMD\_ROLL\_FORWARD;

}

#include "widget.h"

#include "ui\_widget.h"

Widget::Widget(QWidget \*parent) :

QWidget(parent),

ui(new Ui::Widget)

{

ui->setupUi(this);

}

Widget::~*Widget*()

{

delete ui;

}

void Widget::on\_login\_clicked()

{

serverip = ui->ip->text();

cmdport = ui->port->text().toInt();

vedioport = ui->video->text().toInt();

qDebug()<<"listen............";

skt\_cmd = new QTcpSocket(this);

skt\_cmd->*connectToHost*(serverip, cmdport);

if(skt\_cmd->*waitForConnected*(1000))

{

qDebug()<<"connect success............";

connect(skt\_cmd, SIGNAL(connected()), this, SLOT(listen()));

Dialog \*p = new Dialog(skt\_cmd,this);

p->show();

this->hide();

connect(p,SIGNAL(mysignal()),this,SLOT(myslot()));

}

else

{

qDebug()<<"connect failed......";

}

}

void Widget::listen()

{

qDebug()<<"listen success......."<<endl;

}