

**课 程 设 计**

**课 程：物联网工程开发与应用实例**

**题 目：基于Zigbee的智能温湿度检测系统-windows端程序**

**学生姓名：冯贺彬**

**学 号：15115061044**

**二级学院：信息科学与工程学院**

**专 业：物联网工程**

**班 级：2015级01班**

**指导教师姓名及职称：刘晓樑 讲师**

**起止时间：2018年 9 月—— 2019年 1月**

（教务处制）

**基于Zigbee的智能温湿度检测系统**

**摘 要**： 本设计主要针对环境的温湿度进行监测，在传统设备的功能基础之上进行改进和创新。温湿度传感器模块检测数据与协调器无线组网，协调器通过串口通信把数据上传到服务器端，服务器端通过TCP通信把数据传送给用户，用户可以在数据库中查看接收数据以来各个时段的温湿度数据。数据传输模块包含ZigBee传输，串口传输，socket。传感器终端通过单播的形式与协调器联网，监控终端通过建socket通道与用户端进行通信。在连接数据库方面，本系统用到了Qt自带的一个数据库，也就是Sqlite数据库，用户可以在系统中查看数据库数据，也可以在独立的Sqlite数据库对温湿度数据进行查看。系统用到的技术主要有ZigBee无线组网，串口通信，TCP/IP通信，数据库的连接管理等。

**关键字**： ZigBee；嵌入式；温湿度监测；QT

**Intelligent Temperature and Humidity Detection System Based on Zigbee**

**Abstract:** This design mainly monitors the temperature and humidity of the environment, and improves and innovates on the basis of the function of traditional equipment. Temperature and humidity sensor module detects data and coordinator wireless networking. Coordinator uploads data to server through serial communication. Server transmits data to user through TCP communication. Users can view temperature and humidity data in the database at all times since receiving data. Data transmission module includes ZigBee transmission, serial transmission, socket.Sensor terminals are connected to the coordinator through unicast, and monitoring terminals communicate with users through socket channels.In connection with the database, the system uses a database that comes with Qt, that is Sqlite database. Users can view database data in the system, and also can view temperature and humidity data in an independent Sqlite database.

The main technologies used in the system are ZigBee wireless networking, serial communication, TCP/IP communication, database connection management and so on.

**Key words:**  ZigBee; Embedded; Temperature and Humidity Monitoring; QT

**目录**

[1 温湿度监控终端的设计相关技术 1](#_Toc30260)

[1.1 ZigBee传输技术 1](#_Toc24763)

[1.2 Z-Stack协议栈 2](#_Toc23425)

[1.3 IAR开发环境 3](#_Toc17295)

[1.4 终端及QT图形应用技术 3](#_Toc11255)

[1.5 MySql数据库技术 4](#_Toc21492)

[2 系统设计方案 5](#_Toc12177)

[2.1 功能流程框图 5](#_Toc31853)

[2.2 模块功能介绍 5](#_Toc393)

[3 windows端程序的设计与实现 6](#_Toc6575)

[3.1 建立通信和停止通信 6](#_Toc32618)

[3.2 实时数据显示 7](#_Toc19618)

[3.3 IP输入框清空 8](#_Toc6788)

[4 界面设计 8](#_Toc20214)

[4.1 主界面设计 8](#_Toc34)

[5 功能测试 9](#_Toc17731)

[5.1 连接测试 9](#_Toc24414)

[5.2 接收数据测试 10](#_Toc15578)

[6 总结及展望 10](#_Toc5182)

[6.1 总结 10](#_Toc16362)

[6.2 展望 11](#_Toc16228)

[参考文献 12](#_Toc19368)

**基于Zigbee的智能温湿度检测系统**

# **1 温湿度监控终端的设计相关技术**

## 1.1 ZigBee传输技术

ZigBee传输技术是本设计十分重要的组成，本设计是立足于无线特点开发的多参数监护仪，那么，如何选择无线传输技术就显得相当重要。

下面本文通过比较几种常见的无线技术来说明本设计选择ZigBee的理由。

红外技术是比较早出现的无线技术，虽然有成本低、简单易用的特点，但是，有物体阻挡时，便显得无可奈何了，且仅可以直线传播，所以，不符合本设计需求；WiFi是应用的最多、最流行无线通信技术，传输最远范围是100-300米，它的最高速率将近300Mbps，功耗在10-15mA之间；Bluetooth的有效覆盖半径仅有2-30米，功耗小于WiFi，但大于ZigBee；ZigBee有效覆盖半径在50-300米之间，功耗5mA，还可以自组网，节点数为216个。显然不管是在功耗还是成本，或是传输距离方面ZigBee都是最适合本设计的。

经科学地分析和比对，本系统选择了ZigBee技术作为本系统的无线传输技术支持。

另外ZigBee通信需要至少需要两个CC2530模块，如图1-1所示，左边的节点是用于与传感器相连接的，右边的协调器是用于接收节点数据或者发送指令的，与它相连接的是监护控制终端（即嵌入式设备）。在本系统中，用了三个节点（节点是指连接传感器的主控模块），和一个协调器模块来搭建无线网。三个节点分别是体温传感节点、血压传感模块、心电和血氧二合一模块，协调器完成数据的汇总以及与监护控制终端的数据交互。



图1-1 ZigBee组网套件

## 1.2 Z-Stack协议栈

Z-Stack是用于CC2530组件的小型的操作系统，它是一个ZigBee模块化的协议栈，能用于于多个平台开发。它是一种小型的操作系统，它里面集成了很多函数方法，比如、、、等，可以十分容易的初始化操作系统，配置硬件，还可以十分方便的使用定时、串口、点灯等功能，可以大大的提高程序员编程效率。它主要进行了CC2530模块硬件的初始化，如WatchDog的使能、中断的初始化、显示屏的初始化等。Z-Stack协议栈架构如图1-2，分为物理层、MAC、NWK、APL层面。其工作流程如图1-3所示。

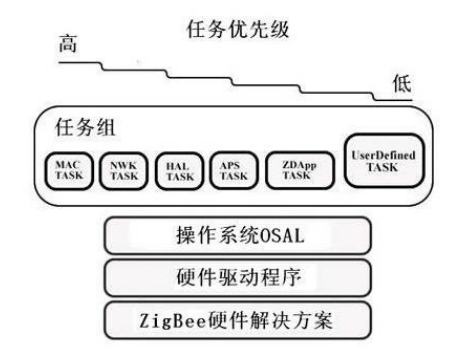


图1-2 Z-Stack软件架构

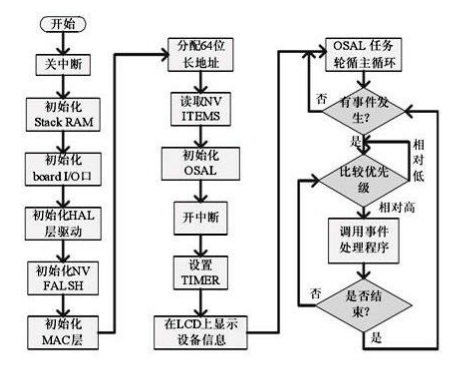


图1-3 Z-Stack软件流程图

## 1.3 IAR开发环境

硬件开发用的是Windows环境下的IAR9.10版本，IAR可以用于CC2530的开发，它是优秀的C编译器，它能够支持非常多半导体企业的芯片，例如ARM、CC2540、CC2530等芯片。它能够打开Z-Stack协议栈，在Z-Stack协议栈的配合下，可以很方便的开发出ZigBee+传感器框架的各种产品，开发软件界面如图1-4所示。

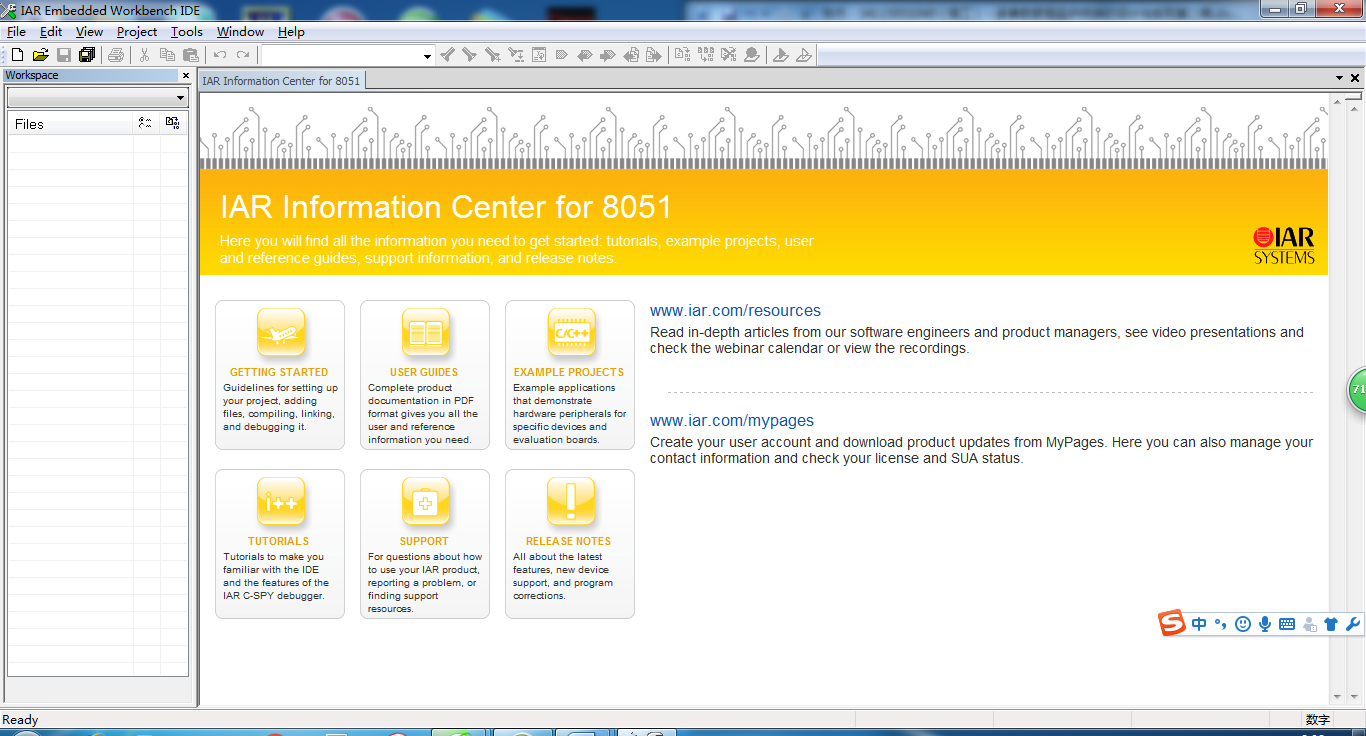


图1-4 IAR集成开发软件界面

## 1.4 终端及QT图形应用技术

被经常用作终端机的主要设备分别是三种手机、电脑及嵌入式设备。首先手机不能直接和ZigBee通信，需要通过复杂的转换，且难以保证24小时不间断监护，所以本文不推荐使用手机作为终端，另外由于手机在本设计中，用于拓展功能远程控制端；而考虑到并不是所有家庭都有电脑，他也跟手机一样具有缺点，所以也排除了电脑，嵌入式设备具有软硬件可裁剪、价格、质量、耗能可控等特点，用它作为终端再适合不过了，因此本设计采用的嵌入式板作为终端设备，配置7寸的电容屏，提供用户交互的功能。

显示方面，本设计使用Qt图形应用界面开发技术，它最大的特点是一个平台开发，可以多个平台编译，可以十分方便的移植到嵌入式设备上去。Qt是一个面向对象的C++架构，方便扩展，被普遍应用于linux平台、Windows平台应用程序的开发上，当然，它也支持Android软件应用的开发，它的开发平台是Qt Creator。

Qt Creator是由美国某企业研发的一个多方向的C++语言图形应用界面程序的集成开发软件。它能够开发GUI（图形用户界面）程序，还能开发服务器等其他非GUI的应用。它的软件界面如图1-5所示。

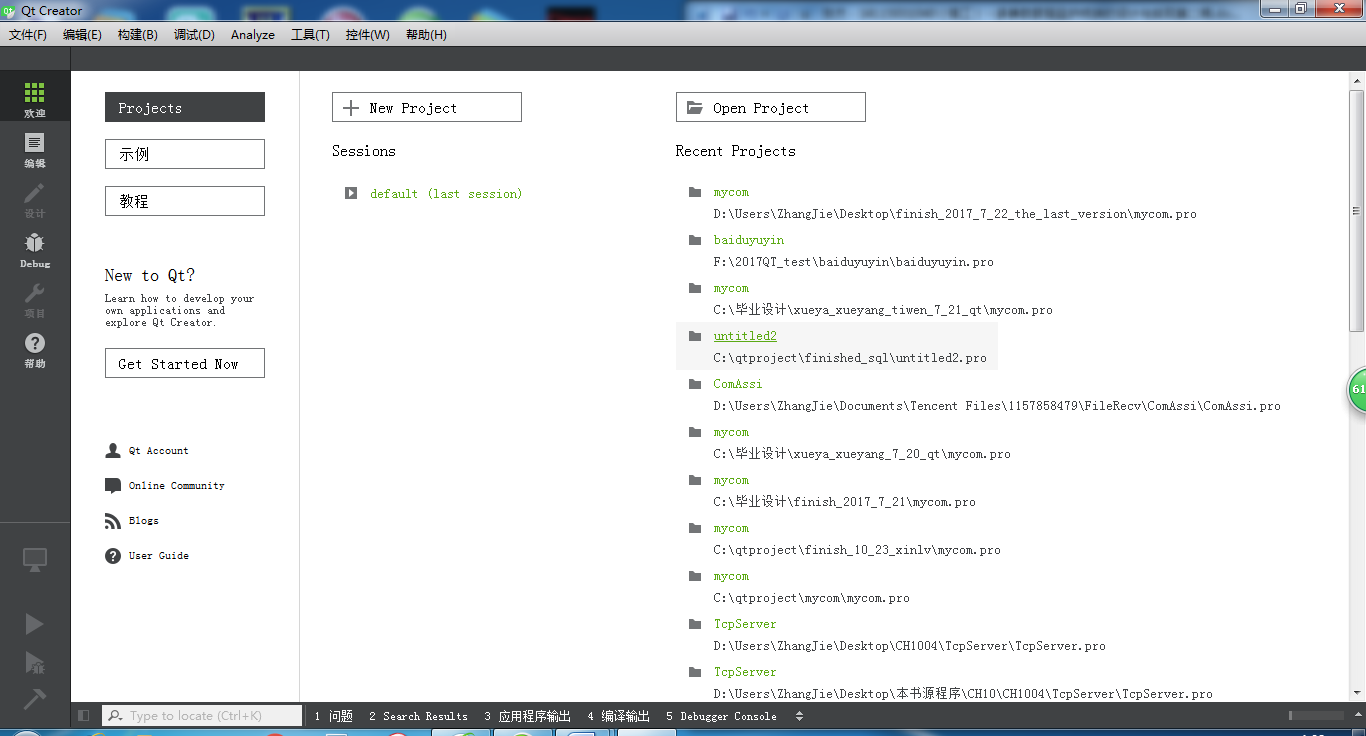


图1-5 Qt Creator集成开发环境界面

## 1.5 MySql数据库技术

本设计使用的windows端数据库是Mysql，在Qt里搭建Mysql数据库，用来存储温湿度的实时数据以及接收时间。

MySql数据库管理系统把用户的数据存放在不同的表格中，这种存放的方式极大程度的提高了存取速度和它的使用灵活性。它访问数据库的方式与其他常用大型数据库（Sql Sever2008等）访问数据的方式相同，采用SQL语句，这极大程度减轻了本设计的工作量。它还具有的特点有体积小、高速、省成本。对本系统其最大的好处是开放资源，是一个不用钱数据库。

它有下面几种类型的应用情景：

1）单点式（Single），用作小规模应用情景；

2）复制式（Replication），用作中小规模应用情景；

3）集群式（Cluster），用作大规模应用情景。

# **2 系统设计方案**

## 2.1 功能流程框图

Linux终端接受数据并进行显示，处理，传输

DHT11

采集实时数据，通过zigbee网络传输

接收到的数据存入数据库，并设立管理员进行管理

Windows端程序与linux终端进行通信，接收数据

图2-1 功能流程框图

如图2-1功能流程框图所示，本设计利用DHT11温湿度传感器采集温湿度的实时数据利用CC2530搭建的ZigBee无线传感网络进行数据的实时传输，所有数据统一聚合到linux的显示控制终端上显示，处理，数据再经过局域网转发至windows上，再由windows端的Qt数据库来存储。

## 2.2 模块功能介绍

数据采集模块是使用DHT11温湿度传感器进行采集数据的，对采集到的温湿度数据进行校准和包装，一次传输的温湿度数据为40位；8bit湿度整数数据 + 8bit湿度小数数据 + 8bint温度整数数据 + 8bit温度小数数据 + 8bit校验位。然后通过与协调器组成的zigbee无线网络进行数据传输。

协调器与Linux端进行数据传输的时候使用到了UART驱动和USB串口通信。通过串联各种数据，使得数据到达Linux端控制显示温湿度数据然后做处理。

显示控制终端主要职责是近距离的无线监视的功能，本功能主要针对缺点为传统设备的拖线多、干扰用户活动等的问题。本部分利用Qt编写终端图形应用程序，然后将应用程序移植到Linux系统的终端上运行。其他功能还有温湿度数据的查看，传感器的控制等。

windows端程序的主要职责是与Linux终端进行通信，接受传输来的实时数据并显示，执行存入数据库的操作。其他功能有用于管理数据库等。本设计我负责的便是windows端的程序设计。

# **3 windows端程序的设计****与实现**

## 3.1 建立通信和停止通信

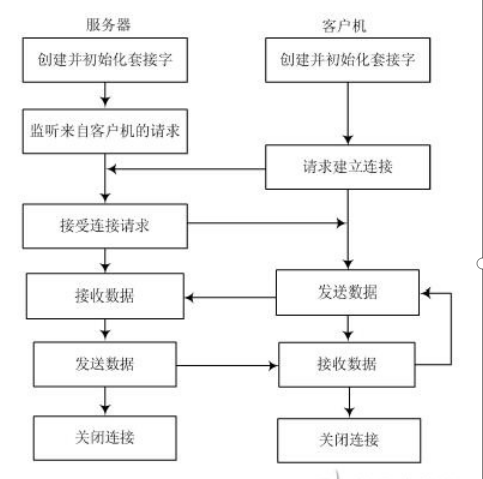


图3-1 socket套接字通信流程图

如图3-1所示Linux端的程序作为服务器监听的一方，在接收到来自windows端的程序的连接请求后，接收连接请求，并且开始进行温湿度实时数据的传输，直至windows端程序发出断开连接的请求后，停止数据的传输。实现代码如下：

void clientWidget::onConnect()

{

ui->pushButton\_connect->setDisabled(true);

socket->*connectToHost*(QString(ui->lineEdit\_socket->text()),9090);

if(!socket->*waitForConnected*(3000))

{

ui->pushButton\_connect->setEnabled(true);

ui->textEdit->append("connect failed!");

return;

}

ui->pushButton\_disconnect->setEnabled(true);

ui->pushButton\_send->setEnabled(true);

ui->textEdit->append("connected to host");

}

void clientWidget::onDisconnect()

{

socket->abort();

ui->textEdit->append("disconnected!");

ui->pushButton\_connect->setEnabled(true);

ui->pushButton\_disconnect->setDisabled(true);

ui->pushButton\_send->setDisabled(true);

}

## 3.2 实时数据显示

## 在接收到来自linux端的温湿度实时数据后，windows端会进行读取信息显示出来并且存入数据库的操作。显示的信息种类有发送方的信息，接收时间，温度和湿度的实时数据等，然后将相关的数据存入数据库。实现代码如下：

void clientWidget::onread()

{

QSqlQuery query;

char bufclient[100];

char w[5];

char s[5];

QString da;

QString tt;

memset(bufclient,0,sizeof(bufclient));

int lengthc = socket->*bytesAvailable*();

if(lengthc>0)

{

socket->read(bufclient,lengthc);

ui->textEdit->insertPlainText("\n");

ui->textEdit->insertPlainText("server : "); /

ui->textEdit->insertPlainText(QDateTime::currentDateTime().toString("yyyy-dd-mm ")); ui->textEdit->insertPlainText(QTime::currentTime().toString("hh:mm:ss"));

ui->textEdit->insertPlainText("\n");

ui->textEdit->append(bufclient);

ui->textEdit->insertPlainText("\n");

da=QDateTime::currentDateTime().toString("yyyy-MM-dd");

tt=QTime::currentTime().toString("hh:mm:ss");

memset(w,0,sizeof(w));

memset(s,0,sizeof(s));

w[0]=bufclient[0];

w[1]=bufclient[1];

s[0]=bufclient[3];

s[1]=bufclient[4];

query.exec(QString("insert into type

values('%1','%2','%3','%4')").arg(da).arg(tt).arg(w).arg(s));

}

}

## 3.3 IP输入框清空

在输入错误的IP地址后，可以进行一键清空的操作，对已经输入的地址进行清除，实现代码如下：

void clientWidget::on\_pushBotton\_clear\_clicked()

{

ui->lineEdit\_socket->clear();

}

# **4 界面设计**

## 4.1 主界面设计

界面程序如图4-1所示，用户在主界面可以看到温湿度的实时数据，及其发送方的相关信息和接收数据的时间。加入了输入框和连接及断开按钮，只要知道服务器的IP，便可自主选择连接的Linux终端并且接收实时数据。加入登录数据库按钮点击后可以进入用户登录界面进行数据库端的相关操作。

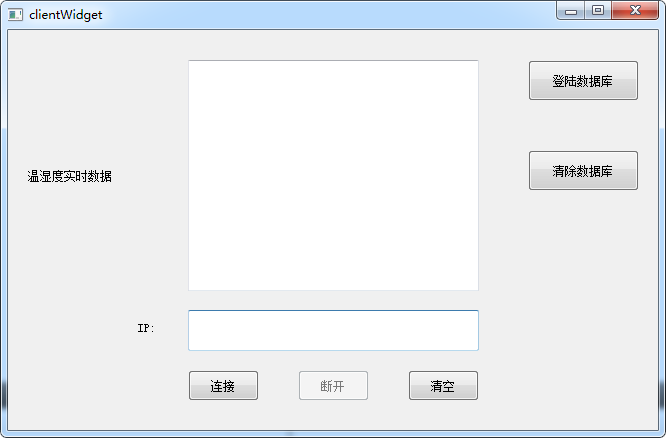


图4-1 主界面设计

# **5 功能测试**

## 5.1 连接测试



图5-1 连接失败

本次测试，Linux端设置的IP地址为192.168.42.119，所以当输入与此不符的IP地址时，点击连接按钮后界面显示如图5-1所示的内容。



图5-2 连接成功

当输入正确的IP地址时，点击连接按钮后界面显示如图5-2所示，此时windows端已经成功与Linux端建立通信。

## 5.2 接收数据测试

## 

图5-3 接收实时数据

建立通信成功后，windows端程序显示发送自Linux端温湿度的实时数据以及相关的时间信息，经对比，确认数据无误，之后，windows端会将这次的实时数据执行存入数据库的操作。

# **6 总结及展望**

## 6.1 总结

本文设计了基于Zigbee的智能温湿度检测系统-windows端程序，完成了其与Linux终端成功通信后显示实时数据并且执行存入数据库的操作的功能的设计与实现。

本次设计做了以下工作：

1. 通过对整体设计的模块化，分析清楚各模块的功能和使用方法；
2. 进行软件的编写，确定通信的流程图和硬件相关函数的使用；
3. 进行主界面的设计，确定界面每个部件的作用以及使用的方式；
4. 通过对相关程序的调试，验证程序和硬件的配套性，最终确定本次设计的可行性和正确性等。

经测试，软件功能能够实现，但仍需要增加程序的实用性研究以及更多功能的添加。

## 6.2 展望

由于设备条件的限制，在测试过程中，实验数据并没有跟真实的多参数监护仪进行比较，主要依据与数据基本特性、曲线特性的对比得出测试结果，所以多参数监护终端可能在精度、准确度上与真实结果有较大的误差，所以我希望本设计今后能进行更严谨的更准确的测试，并对其中的不足进行改进。

另外，在系统中有待改进的地方如下：

1. 补充手机APP端和微信公众平台；
2. 增强系统功能；
3. 增加系统监测的环境参数；
4. 控制终端增加无线模块实现真正意义的无线监测控制等。

# 参考文献

[1]李群芳, 肖看, 关新, 张士军. 单片机微型计算机与接口技术[M]. 北京：电子工业出版社. 2015.2-30.

[2]洛夫. Linux系统编程[M].南京：东南大学出版社. 2009.2-8.

[3]王小强, 欧阳骏, 黄宁淋. ZigBee无线传感器网络设计与实现[M]. 北京：化学工业出版社. 2012.25-35.

[4] Jasmin Blanchette, Mark Summerfield. C++ GUI Programming with Qt 4. 北京：电子工业出版社[M]. 2008. 5-40.

[5]谭浩强.C程序设计(第三版)[M]，北京：清华大学出版社.2011.8-150.

[6]任化敏,王丽，刘仁权.C++程序设计[M]. 北京：中国铁道出版社.2010.10-200.