摘要

计算机技术自20世纪90年代以来飞速发展，尤其是网络通信技术的发展给世界带来极大的改变。与此同时，计算机远程控制技术也慢慢走近人们的视线中，其在计算机技术的发展过程中起到重要的作用，应用在国家的军事、工业以及公司的商业、办公等多种领域。

目前已经研究并实现的远程控制技术应用主要有远程办公、远程教育、远程维护、远程协助、远程控制等领域，在这些传统的远程控制技术主要是应用在PC机一类的终端上。自2008年起，智能手机的崛起在全世界掀起了一股热潮，另外移动互联网的发展也促使智能手机的普及，由此基于移动智能终端的远程控制在技术人员的不断创新中发展起来。在电子提花机的控制系统里，目前还没有使用移动智能终端进行远程控制的相关研究与开发，已有的都是利用嵌入式开发板对其进行远程控制。

本文提出了基于Android移动终端的电子提花机远程控制系统的实现方案，详细阐述该远程控制系统的整体系统需求、总体设计、服务器端设计与实现、控制端设计与实现等环节。论文首所实现的远程控制系统整体采用C/S模式，远程控制系统的Server（服务器）是电子提花机控制器，Client（客户端即控制器）是Android智能手机。在服务器端利用QT在Linux环境下编程并编译，在基于M3352工控核心板上运行，其中QT是跨平台C++图形用户界面应用程序开发框架，其编程语言基于C++面向对象开发。在客户端利用Delphi语言在Rad Studio XE7集成开发环境下编写，Rad Studio XE7具有跨平台开发的功能，可以快速实现应用界面功能。系统的通信机制是根据TCP/UDP协议进行控制端与服务器端之间的数据传输。该系统经过设计与研发，可以成功很好地实现对电子提花机进行远程控制。

**关键字：**远程控制，Android，Delphi，Delphi XE7，QT，QT Creator，TCP/UDP

目录

[摘要 1](#_Toc433654147)

[目录 2](#_Toc433654148)

[**第一章** **绪论** 4](#_Toc433654149)

[**1.1** **研究背景** 4](#_Toc433654150)

[**1.2** **国内外研究现状与发展动态** 5](#_Toc433654151)

[**1.2.1** **电子提花机的发展现状与发展趋势** 5](#_Toc433654152)

[**1.2.2** **远程控制的发展现状与发展趋势** 6](#_Toc433654153)

[**1.2.3** **基于Android移动终端应用程序的发展现状与发展趋势** 6](#_Toc433654154)

[**第二章** **相关技术介绍** 7](#_Toc433654155)

[**2.1** **系统的硬件平台相关介绍** 7](#_Toc433654156)

[**2.1.1** **系统服务端硬件平台介绍** 7](#_Toc433654157)

[**2.1.2** **系统客户端硬件平台介绍** 10](#_Toc433654158)

[**2.2** **系统软件平台相关介绍** 11](#_Toc433654159)

[**2.2.1** **系统服务端软件平台介绍** 11](#_Toc433654160)

[**2.2.2** **系统控制端软件编程介绍** 13](#_Toc433654161)

[**2.3** **系统服务端与控制端的通信机制** 16](#_Toc433654162)

[**2.3.1** **TCP和UDP协议** 16](#_Toc433654163)

[**2.3.2** **SIP-P2P技术** 18](#_Toc433654164)

[**2.3.3** **Delphi XE7网络通信** 18](#_Toc433654165)

[**2.3.4** **QT网络通信** 19](#_Toc433654166)

[**第三章** **远程控制系统的设计与实现** 20](#_Toc433654167)

[**3.1** **远程控制系统总体设计** 20](#_Toc433654168)

[**3.1.1** **系统C/S模型** 20](#_Toc433654169)

[**3.1.2** **系统核心架构** 21](#_Toc433654170)

[**3.1.3** **系统服务端与客户端模块分析** 21](#_Toc433654171)

[**3.1.4** **系统工作原理与工作流程** 22](#_Toc433654172)

[**3.2** **远程控制系统服务端实现** 26](#_Toc433654173)

[**3.3** **远程控制系统客户端实现** 26](#_Toc433654174)

[**第四章** **远程控制系统的客户端的设计与实现** 26](#_Toc433654175)

[**4.1** **远程控制系统服务端需求分析** 26](#_Toc433654176)

[**4.2** **远程控制系统服务端设计模型** 26](#_Toc433654177)

[**4.3** **远程控制系统服务端实现** 26](#_Toc433654178)

[**第五章** **基于移动终端的电子提花机远程控制系统的实验测试与数据分析** 26](#_Toc433654179)

[**第六章** **总结与展望** 26](#_Toc433654180)

1. **绪论**
   1. **研究背景**

2013年，德国政府正式提出工业4.0战略，并且将这个战略工业领域提升到国家战略的高度。此后，全球多个经济大国纷纷提出进入工业4.0时代，中国也不例外。从习近平主席在2014年访问德国时对德国工业4.0战略的表述中，可以看出中国将会打造升级版的“中国工业4.0”。（工业4\_0\_中国的机遇与挑战\_何懿文）工业4.0的重点放在信息互联网技术与传统工业制造相结合。（德国打造工业4\_0\_信息互联与传统工业相结合）在工业4.0时代，生产制造的过程已经不仅仅是人与人之间的交流过程，更是机械之间通过互联网进行数字信息交流的过程。根据德国国家科学机构预估，达到工业4.0后,生产企业的生产效率将提高30%以上。

纵观历史，我国的纺织业有着悠久的历史。在我国这一个纺织大国的国民经济中，纺织机械产业占据着重要地位。在纺织机械中，电子提花机是由计算机来控制提花机的电磁选针，同时结合纺织机实现对织物的提花织造。（我国电子提花机产业现状及发展前景分析）提花机是将传统的提花织物转为机械控制的装置，可以按照各种要求来织造图案，现如今，随着科技的发展，电磁选针的方式慢慢代替了初始的机械选针的方式，对于电子提花机的控制成为提花机行业的一个热潮。（基于ARM的电子提花机嵌入式控制系统设计\_冯思轶）

目前电子提花机的控制主要由基于ARM的控制器进行设计的，本文所提出的电子提花机远程控制系统的服务端也是这种基于ARM开发板设计的控制器。

在计算机领域中，远程控制其实并不是指在远距离以外进行控制，而一般指的是通过网络控制计算机设备。对于传统的远程控制，往往是指在LAN（局域网）中进行控制，然而在近年来，随着互联网不断的发展与革新，除了在LAN中控制被控端计算机以外，也可以通过WAN（广域网）对控制端计算机进行远程控制。在远程控制中，在各种先进技术支持的情况下，控制者可以启动被控制端计算机的应用程序，读写所存储的文件资料，甚至可以通过被控制端计算机去访问外网，与生活中的电视遥控器的功能有着异曲同工之处。远程控制中，说得简单些就是被控端计算机一直处于监听状态，监听控制端发送过来的控制信号，一旦有信号就做出相应的操作，之后将操作结果反馈给控制端，所以一切操作仍然是被控端计算机自己完成的。

目前，在远程控制领域中，国内国外都有很多突破性的发展。小到腾讯公司研发的通讯工具中所具有的远程桌面的功能，大到航天事业中地球上的操作人员通过无线信号对星球探测车的远程控制。远程控制技术的发展为人们的平时生产生活带来很多便利，甚至为人类的未来也做出极大的贡献。

（http://baike.baidu.com/link?url=dCC02740MQ-UxtwIGsDhg2aY-3NMW4ylVImnxaO86Da9mEfq5BBtKQH-n0a1zmWl2Vb8-ZdubSyoes11LsR\_6q#3）

随着移动互联网与无线通信技术的发展，智能终端使用群体不断扩大，互联网已经进入移动互联网时代，移动智能终端的发展带动着移动应用市场的发展，丰富多彩的移动应用程序在Android、IOS操作系统的APP Store中如雨后春笋一般日益增多。相对于传统的PC机，移动终端最大的特点就是便捷的移动性，与此同时，无线互联网覆盖的范围越来越广，加上各种移动应用的推广与使用，使人们随时随地享受互联网给日常生活带来的便利与娱乐。2014年Android操作系统占有智能手机市场的81.5%。

（<http://tech.163.com/15/0225/07/AJ9HL4H7000915BD.html>）

通过对以往PC远程控制技术以及Android系统特点的分析与研究，提出基于C/S架构的远程控制模型。

工业4.0时代的到来，使得工业发展与互联网技术紧密联系起来，用互联网迅速发展的技术来带到世界工业发展到另一个历史性的阶段，使其形成一个更加高效快捷的生产体系。另一方面，计算机技术与互联网技术也会在这一过程中因为需求的更大挑战而得到进一步的提高。目前对电子提花机的远程控制大都是由PC机对其进行远程控制，与此同时，随着智能手机的功能越来越强大，所支持的功能远远超过打电话发短信的功能，可以连接互联网，利用互联网的便捷改变以往人们的生活常态。本文利用智能手机的强大功能，研究其对电子提花机这样的工业生产机械的远程控制，这是利用智能手机的网络化来改变人们生产工作的常态，提高生产效率，为操作人员提供一个快捷便利有效的工作环境，这必将是一项具有技术创新性的研究。

* 1. **国内外研究现状与发展动态**
     1. **电子提花机的发展现状与发展趋势**

在目前的国际市场上，有三个国家的制造商所生产的电子提花机产品最具代表性，分别是瑞士STAUBLI公司、德国GROSSE公司、英国BONAS公司。这几家公司已经成为该行业的龙头企业，其生产出的电子提花机是世界公认的差错率低、提花速度快、在花型设计和品种翻改中方便实用，并且具有较高的可靠性、安全性。

国内目前非常重视电子提花机的研发工作，但因为起步较晚，相较于国外实力强大的公司，国内所生产的电子提花机普遍具有自动化程度不高、纹针数低、不匹配于国外先进无梭织机等缺点。虽然在技术上落后于国外，但是这也引发国内电子提花机的不断创新与改革。

（我国电子提花机产业现状及发展前景分析\_冯华峰）

近年来，随着纺织工业的发展，电子提花机有了很大技术进步，逐步向连续化、自动化、短流程、高质量的方向发展。另外一方面，网络通信的发展也使得电子提花机的发展需要呈现出智能网络化。

在电子提花机的发展过程中，与计算机网络技术的结合已经成为现代电子提花机研发工作中的一部分，这类工作使其具备智能化的控制功能和网路化远程控制功能。电子提花机的由来就是因为将电子计算机技术应用在提花机上，实现远程控制，达到高精准度和高效率的效果。

（浅析电子提花机创新与现代纺织技术\_电子提花机创新项目管理有感\_罗东升）

* + 1. **远程控制的发展现状与发展趋势**

远程控制必须通过网络才能进行。位于本地的计算机是操纵指令的发出端，称为控制端或客户端，非本地的被控计算机叫做被控端或服务器端。“远程”不等同于远距离，主控端和被控端可以是位于同一局域网的同一房间中，也可以是连入Internet的处在任何位置的两台或多台计算机。

早期的远程控制大部分指的是电脑桌面控制，实际的远程控制用安卓，苹果，笔记本，电脑都可以控制马路上的灯，能控制连网的窗帘，能遥控电视机、DVD，能控制摄像机、投影仪。指挥中心，大型会议室等都普遍应用远程控制技术。

随着智能手机的发展，远程控制技术也从PC控制转向移动终端控制，控制端越来越便捷化、智能化。

* + 1. **基于Android移动终端应用程序的发展现状与发展趋势**

2007年Google公司研发的Android手机操作系统，是一个搭载在移动设备上的操作系统，更确切的说，它是一个开放性的移动设备综合平台，除了操作系统外，它还包括中间件和一些关键的平台应用。

（基于Android的移动终端应用程序开发与研究\_公磊）

Android最大特点在于它是一个开放的体系架构, 具有非常好的开发和调试环境, 而且还支持各种可扩展的用户体验, Android里面具有非常丰富的图形系统, 对多媒体的支持功能和非常强大的浏览器。

目前在智能手机市场上，Android系统的手机的占有率远远超过了其他操作系统的手机，2014年Android操作系统占有智能手机市场的81.5%。

（http://tech.163.com/15/0225/07/AJ9HL4H7000915BD.html）

这样的市场占有比例是必然的，因为谷歌公司为了推动Android系统的发展，为系统的开发人员提供了各种支持，因此，在安卓应用市场上，大批应用程序如雨后春笋般涌现，应用的功能更是丰富多彩。相应的，安卓应用市场也为开发人员或者公司提供购买体验，使得他们能够得到开发的报酬，这从另一方面也增大了开发人员或者公司的开发动力。未来，在应用市场上还是会涌现出很多的应用，并且应用的功能会更加强大，给人们的衣食住行带来极大的改变。

1. **相关技术介绍**

本文所研究的基于Android移动终端的电子提花机的远程控制系统是一种C/S框架的系统，在服务端和客户端都有软件系统设计与实现。本章首先介绍服务端与客户端的硬件支持，再根据两端软件开发的IDE（集成开发环境）对两端的软件开发语言做简单介绍，最后详细介绍本文所研究的远程控制系统的整体系统设计细节。

* 1. **系统的硬件平台相关介绍**

系统的服务端是电子提花机控制器，是基于M3352系列嵌入式工控核心板，其系统是开放性良好、易于移植的Linux系统，系统的客户端是Android系统的移动终端。本节将对这两种系统进行介绍。

* + 1. **系统服务端硬件平台介绍**

服务端控制器采用M3352嵌入式核心板，M3352的处理器是AM3352，可以提供相同成本下最高的DMIPS（Dhrystone Million Instructions executed Per Second，指的是计算机计算能力），集成高性能外设来满足工业设计的需求。它可以实现面向包括便捷式导航系统、工业自动化、人机界面等多个行业应用。开发板所搭载的系统的Linux系统，该系统具有多用户、多线程、多任务的特点，它基于POSIX和UNIX。（陈海燕，任松岩.嵌入式Linux根文件系统的研究J7.现代电子技术.2010, 4.）M3352开发板样式如图1-1所示。

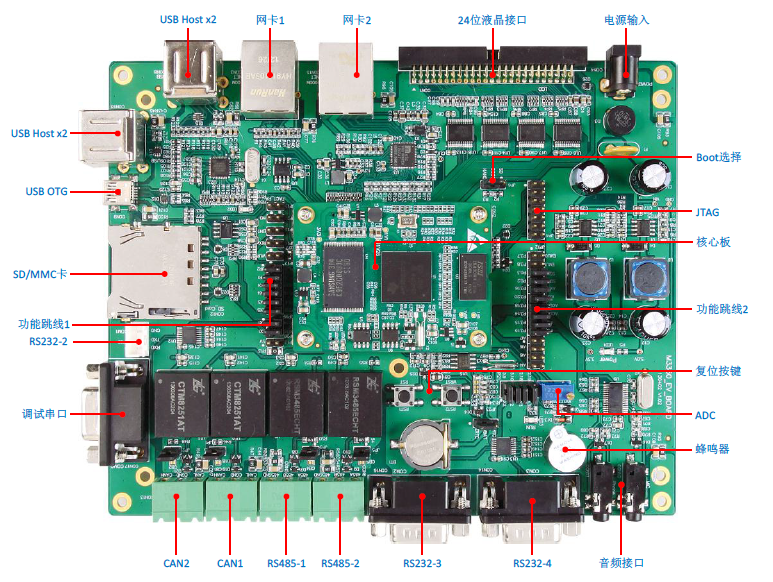


图1-1 M3352工控核心板样图

嵌入式 Linux 系统，由于系统资源的匮乏，通常无法安装本地编译器进行本地开发，而需要在借助一台主机进行交叉开发。一般情况下，主机运行 Linux 操作系统，在主机安装相应的交叉编译器，将在主机编辑好的程序交叉编译后，通过一定方式如以太网或者串口将程序下载到目标系统运行，或者进行调试。一般的交叉开发流程如图1-2所示。

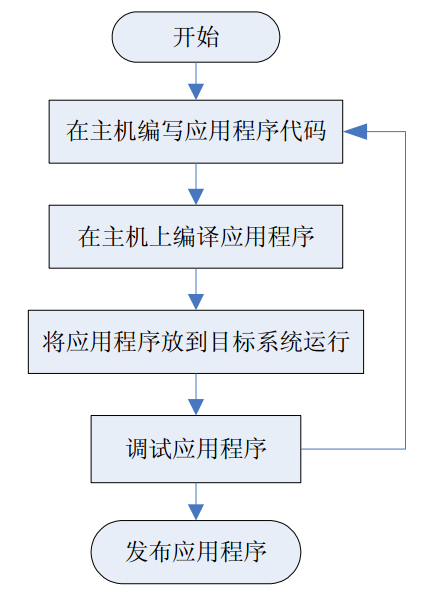


图1-2 嵌入式Linux交叉开发一般流程

嵌入式 Linux 开发的一般模型如图1-3所示。通常需要一台 PC 主机，在其中安装好各种进行交叉编译所需要的软件，通过串口和以太网和目标板相连。在主机上进行程序编辑和编译，得到的可执行文件通过串口或者以太网下载到目标板中运行或者进行调试。

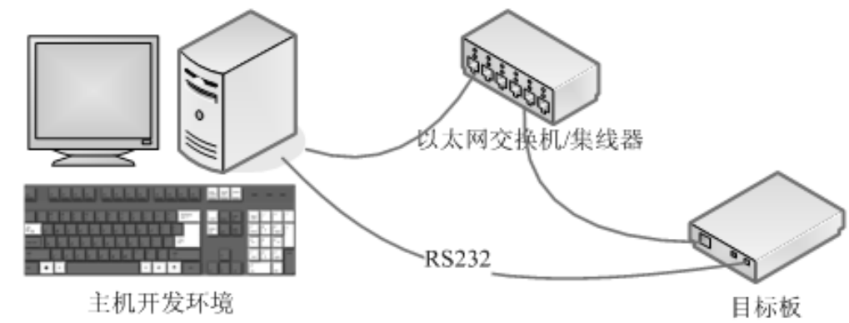


图1-3 嵌入式Linux开发模型

进行嵌入式 Linux 开发，NFS（Network File System）方式应该是最常用的开发方式了。主机开启NFS服务，作为NFS服务器，NFS的客户端是目标板，通过 NFS 方式，将主机 NFS 服务器目录挂载到本地系统，就可以像操作本地文件一样操作远程机器的文件。

对主机的要求，硬件方面，要求有串口和网口；软件方面，操作系统推荐使用主流发行版，如 Ubuntu 等，还需要安装开发相关的软件，同时还需要其它的软件如交叉编译器。主机环境如下图所示。



图1-4 主机环境例图

* + 1. **系统客户端硬件平台介绍**

Google公司在2007年研发出Android操作系统，一年后，第一部Android手机问世。此后Android系统逐渐由手机平台扩展到平板电脑、电视、数码相机、游戏机等设备上，该系统在市场上的使用率一跃领先，风靡全球。

（http://baike.baidu.com/link?url=-ND2-lgcpecw4ibx3UxrpyPvP9NbvKx2jvTs5OQJwS6qQWFXX2NZCN0I\_VhdzcMW1zjJF4My3M4pfBYLhCkZAhm1UiZQmaLnsLXn\_SJuwEa）

Android操作系统的内核是Linux系统，支持一部分基本的 Linux 命令，采用了与 Linux 类似的权限管理。Android平台由四部分组成，由低到高分别是操作系统、中间件、用户界面、应用软件。（E2Ecloud工作室.深入浅出Goggle Android[M]，北京:人民邮电出版社，2009:3.）

Android采用了分层架构，共分为四层。最底层是Linux内核层；Linux 内核层之上是系统运行库，为开发者提供一些基本的库支持；再上层是应用程序框架，这个层里提供的主要是开发者和 Android 相关的一些组件的实现；顶层是应用程序层，这层负责与用户进行直接交互。Android体系结构图如下：

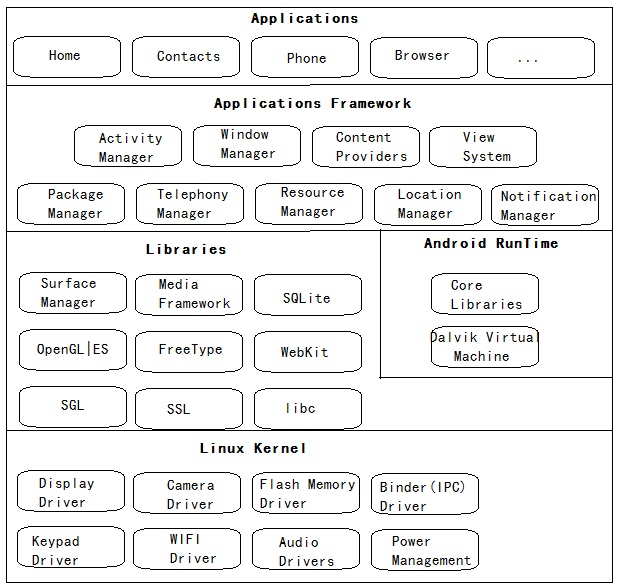


图1-5 Android体系结构图

* 1. **系统软件平台相关介绍**

服务端控制器所开发的控制应用程序是在前面搭建的主机中进行编程开发的，属于嵌入式的GUI系统，所采用的IDE是QT Creator。客户端Android终端的应用程序是在Delphi XE7中进行Delphi语言开发的。

* + 1. **系统服务端软件平台介绍**

目前对于嵌入式Linux系统下的GUI系统的主流解决方案有四种：QT、Microwindows、Pure X架构和OpenGUI。其中QT具有强大的可移植性，并且对于入门的开发人员来说很容易上手并运行项目，所以本研究选择使用QT进行嵌入式GUI开发。

QT Creator是跨平台的QT语言的 IDE，使用强大的C++代码编辑器以至于可以快速编写C++代码，它继承了QT Designer、QT Assistant、QT Linguist、图形化的GDB，从而可以为QT开发人员提供快速、轻松地开发应用程序。QT Creator的开发界面如下图所示。目前QT Creator的最新版本是3.1.0，已经完全实现对IOS的支持，逐步实现对Android的支持，可见QT Creator的跨平台性正在逐渐完善提高。

（<http://baike.baidu.com/link?url=TAQBJP9ol_uErL-XRMIqZPCGi5LJ8hi2zYtwTh5VsXIWLwPUsunPgCduOrLVlkE8XbbfBfrw-Smw1fCwurJF9a>）

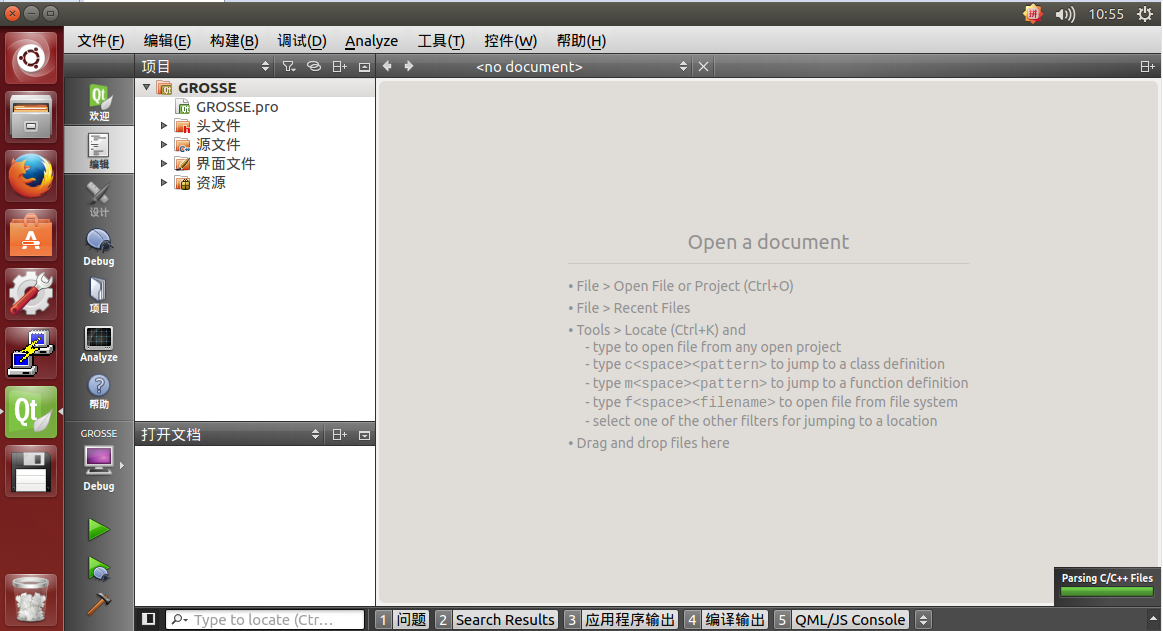


图1-6 QT Creator IDE开发环境图

QT是基于C++的面向对象的框架，属于跨平台C++图形界面应用程序的开发框架，不仅可以开发GUI应用程序，也可以用来进行非GUI程序的开发。它具有以下几点特性：

1. 跨平台

QT语言诞生的一个重要的目的，就是要实现跨平台开发，它是一个适用于UNIX也适用于MsWindows的GUI工具包。

1. 容易入门上手

对于一个熟悉C++这种OOP（Object-Oriented Programming，面向对象）编程语言的开发人员来说，入手QT语言应该属于一件非常简单的事，因为QT属于一种C++工具包，包含了很多的C++封装的类，开发人员只需熟悉这些类的功能，并且在开发过程中去调用即可，所以QT是一门易用性强的OOP语言。

1. 运行速度快

QT除了上述的两个特点之外还有很快的运行速度，它使用多平台的低级绘图函数对MsWindows和Motif进行仿真，另一方面，QT的开发者对产品进行很多的优化工作，这使得QT同时达到了易于使用和运行速度快的两个特点。QT的代码片段如下图所示。

(基于Qt的嵌入式Linux系统GUI的研究与实现\_范朋)



图1-7 QT代码片段

* + 1. **系统控制端软件编程介绍**

系统控制端是基于Android操作系统的移动终端，所采用的开发IDE是Delphi XE7。Delphi XE7也称为Rad Studio XE7，其前身是Borland公司集合Delphi和C++ Builder的可视化集成工具，（智能手机平台浏览器上网记录的提取与分析\_王宇阳）后来Embacadero（英巴卡迪诺）公司收购Delphi和C++ Builder，推出RAD Studio XE系列产品。

（RADStudioXE7[EB/OL].http://www.embarcadero.com/cn/products/rad-studio.）

Delphi XE7是利用单一代码库来为Windows、Mac、IOS、Android快速构建原生应用程序的完整软件开发解决方案，为了无需牺牲应用质量以及应用的性能，使用了相同的源代码库构建应用。新版本的Delphi XE7具有以下几个特点：

* 完美实现跨Windows、Mac、Android和IOS平台开发，而且代码无需改变；
* 具有可视化集成工具，可以使应用程序开发速度提高5-20倍；
* 任何可视化集成工具中的控件都可以通过LiveBindings绑定到其他对象；
* 控件样式的不断更新使得应用程序的界面样式变得更加现代化，适合大众的审美标准；
* 可以访问基于云的BaaS和REST服务。

（智能手机平台浏览器上网记录的提取与分析\_王宇阳）

现在RAD Studio XE7已经逐渐成为一款成熟稳定的跨平台开发IDE，使得Delphi语言又重新燃起希望之火。RAD Studio XE7的开发环境界面如下图所示。

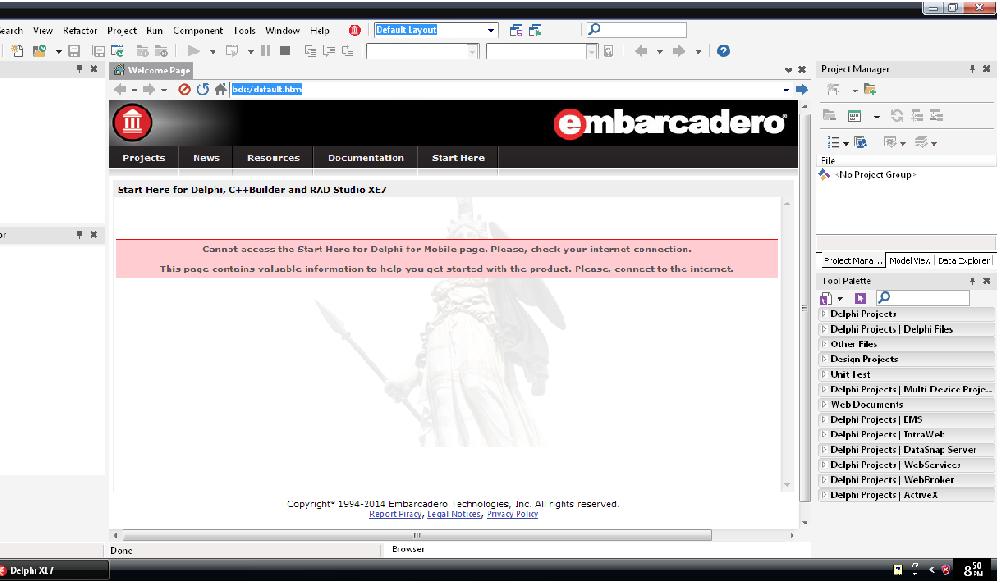


图1-8 Delphi XE7开发环境

2012年Embacadero公司推出FireMonkey技术，它是一种基于CPU、GPU混合架构的应用平台，使用FireMonkey技术可以帮助开发人员只要通过一次编写程序创建所需的应用程序，然后选择不同的目标操作系统就可以开发出不同操作系统下的应用程序，包括Windows 32位/64位、Mac OS、IOS和Android，这样就实现了一套代码应用于不同平台从而简化开发流程。另外FireMonkey技术的GPU动力从而提高用户体验，增强应用程序的视觉效果，提高用户使用程序时的流畅性。FireMonkey技术还支持数据库应用开发，提供控件帮助开发人员访问目前技术领域所用的大部分数据库。本文使用FireMonkey技术主要是运用其跨平台技术以及提高用户体验的流畅性。使用FireMonkey技术创建Android APP项目以及开发Android APP的界面如下图所示。在创建项目之后，选择项目所需要的操作系统平台，然后在IDE的右侧，开发人员可以通过拖动控件的方式快速开发应用程序。（面向招考信息的网络舆情系统设计与实现\_李明）

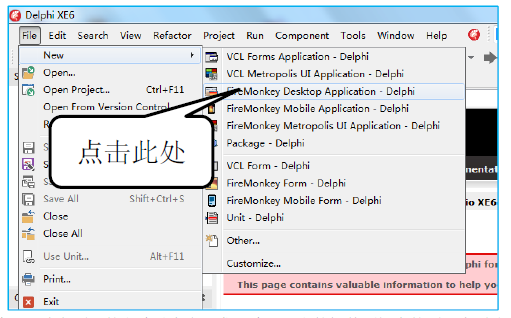


图1-9 Delphi XE7创建多平台设备APP项目

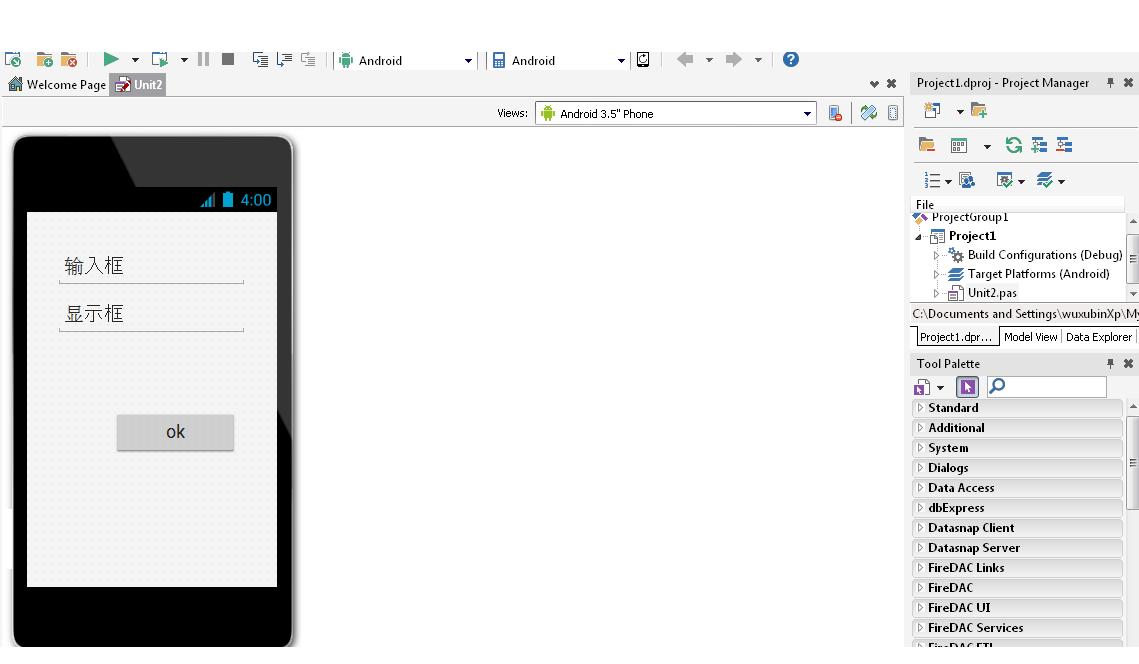


图1-10 Delphi XE7开发Android APP界面

1995年，美国Borland（宝兰）公司开发Delphi语言，其研发初衷是为Windows平台研发一种开发工具。Delphi是由传统Pascal通过加入面向对象的功能后演变而来，从此Delphi成为一门OOP语言，受到Windows开发人员的热捧。经过数年的发展，Delphi从Borland公司转至成为Embarcadero公司旗下的产品。目前Delphi已经成为一个集成开发环境，主要的开发环境是图形用户界面，使得程序开发人员的编程效率大大提高。在Delphi的发展历史中，从Delphi1到现在的Delphi XE8，Delphi由开始的Windows图形开发工具发展成跨平台的图形化用户界面IDE。

Delphi语言具有以下功能特性：

* Delphi作为一个可视化的IDE，采用面向对象的编程思想和基于部件的开发结构框架。开发人员可以修改部件来满足自己的需求。
* Delphi集合了VC功能强大和VB简单易用的特点，成为功能强大并且易用的编程语言。
* Delphi适用于多种数据库结构，具有高效的数据库管理系统和先进的数据库引擎。

（http://baike.baidu.com/link?url=oWHa2jGwV7HRyb2aalb1urqmnPjYQkL3ERo\_yJep7e\_sxhiScLpeW6aoi8I4oJ1Eq3yRv4un\_1g7fdtmNW0p20EyYAYzGug-ujpG7GHdsgO#1）

Delphi是由经典的Pascal语言通过一种面向对象扩展而来，Borland公司在研发阶段一直致力于Turbo Pascal编辑器的研究。Pascal语言相对于大家熟知的C语言来说具有可读性强并且语法十分详细等特点。与目前比较火热的其他OOP种类的编程语言一样，Delphi语言也是遵循相同的OOP扩展方法。（mastering Delphi7）

Delphi XE系列产品在移动端开发上，投入了很多精力，一开始XE系列产品适应IOS系统，对于Android系统是后来慢慢支持的。

* 1. **系统服务端与控制端的通信机制**

系统采用C/S框架来进行架构，服务端与客户端需要实现数据交互，达到实时通信的目的。在系统的通信机制上，本文主要研究了TCP协议和UDP协议、SIP-P2P通信技术以及系统两端的通信编程。

* + 1. **TCP和UDP协议**

目前在网络通信体系中，TCP/IP（Transmission Control Protocol/Internet Protocol）协议是互联网最基本的网络通信基础，由OSI（Open System Interconnection）模型中的网路层的IP协议和传输层的TCP协议组成。

（http://baike.baidu.com/link?url=nzUXFM6RqVYfB2xsfab-kFxswexQNaG326dHEXXqDyfxvjzakPuVCwEpOAv4qq1BDCIY19j0oxGvujUBKBUsYK）

TCP（Transmission Control Protocol）传输控制协议和UDP（User Datagram Protocol）用户数据报协议都是一种在传输层的通信协议。

TCP协议面向连接，基于字节流，并且具有一定的可靠性。基于TCP协议连接的通信双方需要在已经形成TCP连接的基础上才能进行可靠的数据通信，这样的连接就是经典的三次握手，三次握手的连接如图所示。为了确保数据传输的可靠性与准确性，接收数据端在接收数据时需要进行TCP数据校验，一旦数据有差错就会抛弃出错的数据片，接收端在这种情况下不发送确认信息，发送端在检测到超时的时候会重新发送数据。TCP协议的报文格式如图所示。报文主要包含两个部分，分别是报头和数据。其中报头是携带源端口、目的端口、顺序号、确认号等可以用来标识的信息。



TCP三次握手：数据段交换



TCP报文格式

UDP协议面向无连接，提供那些事务简单不可靠信息传送服务。这两种协议各有各的优势，适用场合不经相同。UDP协议的主要作用是压缩网络数据形成数据报形式，将数据报在网络中进行传输。UDP协议的报文格式中的报头所携带的信息量比TCP协议少得多。所以在相同情况下，UDP协议所携带的数据量比TCP协议多。下图是UDP协议的报文格式图。（TCP\_UDP网络传输协议差异之我见\_雷显臻）



UDP报文格式

对于TCP协议与UDP协议的区别，本文列出表格，将二者的区别罗列了出来。根据两个协议的区别，得出TCP协议虽然可以进行可靠传输，但是其丰富的功能反而会使得传输性能下降，本文所研究的系统使用的是UDP协议。



TCP协议与UDP协议区别

* + 1. **SIP-P2P技术**

系统采用C/S框架来进行架构，服务端与客户端需要实现数据交互，达到实时通信的目的。在系统的通信机制上，本文主要研究了TCP协议和UDP协议、SIP-P2P通信技术以及系统两端的通信编程。

* + 1. **Delphi XE7网络通信**

Delphi XE7作为一款跨平台开发IDE，它提供丰富的组件给开发者进行相应的开发。对于网络通信，Delphi拥有非常成熟的组件，Indy是Delphi 7所带的开放源代码组件集，全称是Internet Direct。

Indy是Delphi的一种实现阻塞式Socket模型的网络组件，包括使用TCP协议、UDP协议和原生Socket的供客户端和服务端使用，除此之外，Indy组件还包括更加高级的网络协议。比如HTTP、FTP、SMTP、POP3、NNTP等供客户端和服务端使用。Indy组件是利用Delphi语言来开发的，但是也可以被其他OOP语言所使用，例如C++、C#和.NET语言。Indy可以在不同平台很多运行环境下使用，包括Win32、Linux、.NET1.1、Mono。Indy目前正在不断地更新当中，它属于一种单源库，每个版本的源代码所生成的库和组件都只适应与特定的运行平台和运行环境。

Indy是阻塞式调用，阻塞式调用与文件的读写操作非常相似。如果现在正在执行某个函数对某些数据进行读写操作时，在操作结束之前这个函数将不会有任何返回，一旦操作完成立马返回。区别于文件操作的是，对数据的操作所需要的时间可能会更长，因为数据可能不会立即被执行操作。再比如，我们在使用Indy去调用连接方法时会有一个等待的过程，连接结束时会有一个结果返回，根据连接是否成功，返回的结果也不一样。

Indy阻塞式socket的优点有：

* 编程简单。阻塞式Socket应用程序很容易编写。所有的用户代码都写在同一个地方，并且顺序执行。
* 容易向Unix移植。由于Unix也使用阻塞式Socket，编写可移植的代码就变得比较容易。Indy就是利用这一点来实现其多平台支持而又单一源代码的设计。
* 很好的利用线程技术。阻塞式Socket是顺序执行的，其固有的封装特性使得它能够很容易地使用到线程中。

但是Indy也有它自身的一些缺点。Indy会使客户程序的用户界面“冻结”。当在程序的主线程中进行阻塞式Socket调用时，由于要等待Socket调用完成并返回，这段时间就不能处理用户界面消息，使得Update、Repaint以及其它消息得不到及时响应，从而导致用户界面被“冻结”。

在本文所研究系统的客户端中，网络通信模块所用的是Delphi Indy组件中的TIdUDPServer类，该类的层次结构如图所示。



Delphi XE7中TIdUDPServer类层次结构图

* + 1. **QT网络通信**

QT在网络通信部分为应用做TCP/IP协议相关的开发提供一套API，其中对基于HTTP协议的请求、Cookies和数据的发送都有封装C++类来进行处理。基于TCP/UDP协议，QT分别封装了QTcpSocke类和QUdpSocket类供开发者使用。

TCP协议是一种面向连接、可靠性强的传输协议，适用于传输连续的数据。QTcpSocket类提供TCP协议接口，开发人员可以基于此来实现其他协议的传输。基于TCP协议的传输必须建立在客户端与服务端以及成功连接的基础上，该连接被关闭时，数据的传输也会立即停止。

与QNetworkAccess类一样，QTcpSocket 类的工作机制是异步进行的，通过发送不同的信号来汇报数据传输的状态的变化以及错误信息。对于数据的传输，QTcpSocket 类依赖于循环来检测传入的数据，并且是自动刷新将要输出的数据。

对于UDP这种面向无连接的轻量级并且没有可靠性的协议，比较适用于那些对可靠性没有要求的场合。QUdpSocket类可以发送和接收UDP数据报，该类与QTcpSocket类一眼继承自QAbstractSocket类，所以这两个类共享QTcpSocket类很多的方法，两个类的主要区别在于QUdpSocket类传输数据包而不是数据流。数据报是一个大小有限的数据包，包含被传送数据的IP地址和端口信息。

另外QT的QUdpSocket类还支持IPv4广播，这里的广播就是往往用来实现网络发现目标，当发送端通过广播发送数据报，这些主机在接收到数据报后，发送一个回复给发送端，这样发送端就发现了目标。

QUdpSocket类的bind()方法调用socket接收传过来的数据报，与QTcpSocket类的listen()方法功能类似。当数据报达到时，QUdpSocket类触发readyRead() 信号函数，该信号函数调用了QUdpSocket类的readDatagram()方法来读取数据报。QUdpSocket类同时也支持多重广播，同时与多设备之间进行UDP数据报的传输。

1. **远程控制系统的设计与实现**

自从数字信息时代的到来，网络通信技术推动着计算机技术，特别是远程控制技术不断地飞速发展。在军事、工业、商业、办公等多个日常生活生产离不开的领域中，远程控制系统都得到广泛的应用。本文利用互联网通信技术将嵌入式应用程序开发和客户端应用程序开发连接起来，研究并实现出基于Android移动终端的电子提花机的远程控制系统，本章将对所研究的内容与实现过程进行阐述。

基于第二章详细叙述的基于Android平台的远程控制系统所相关的技术知识，文章将在第三章讲述系统的设计与实现过程。本章首先系统进行总体设计，包含系统模型、核心架构、服务端和客户端模块设计、系统工作原理以及工作流程。然后会分别进行系统服务端、客户端的实现过程进行说明。

* 1. **远程控制系统总体设计**

本文所研究的远程控制系统采用C/S框架来实现总体架构。系统的两端是不同类型的终端，Client端是Android平台的移动终端，Server端是基于M3352核心板嵌入式Linux系统的电子提花机控制器。有效地实现客户端与服务端的网络通信是本文所研究的核心内容，另外在服务端需要对提花机的工作信息的数据进行进一步处理，分析得出提花机的工作状态，将其工作状态发送给客户端，客户端再将状态信息推送给操作人员，从而可以保证操作人员实时掌控提花机的工作状态。本节将从系统的模型设计入手，叙述系统的核心架构以及系统的工作原理和工作流程。

* + 1. **系统C/S模型**

本文所研究的目的在于实现通过网络通信将手机客户端与嵌入式服务端连接起来，形成同一个网络下的整体，实现客户端对服务端的远程控制。目前研究人员与开发人员在研究开发远程控制系统时，对系统模型的选择主要有两种，一种是C/S（客户端/服务端）模型,另一种是B/S（浏览器/服务器）模型。这两种模型的技术已经十分成熟，并且应用也非常广泛。

上个世纪90年代起，C/S模型就开始盛行，迅速成为软件开发领域系统结构的主流模型。在C/S模型中客户端只有一个，服务端可以有多个，客户端与服务端通过网络建立起网络连接，加以进行数据传输通信。这种模型一个最大的特点就是使得任务分配更加明确化，只要能够有效地进行客户端与服务端的通讯，就可以分别在两端进行独立开发。

B/S模型是C/S模型的一种衍生模型，是WEB互联网技术的一种新型模型，B/S模式的发展也为互联网的推广做出了新的贡献。该模型在结构上可以分为表示层、功能层、数据层等三个结构层次。表示层是电脑中的浏览器，在浏览器中用户可以完成与服务器之间的数据交互；功能层是处理表示层与数据层之间的数据交互；数据层就是完成对数据库的增删改查的功能。B/S模型最大特点是开发成本少，维护和升级简单，应用效率高，但是B/S模型会导致服务端处理的事务过繁琐，运行数据符合过重。

本文将要研究与实现的是一种远程控制系统，该系统中拥有客户端与服务端，所以采用传统的C/S模型来进行组织架构基于Android移动终端的电子提花机远程控制系统的设计与实现，其中客户端是基于Android系统的移动终端，服务端是基于Linux系统的嵌入式开发的电子提花机控制器。根据C/S架构的基本思想，结合本文所研究的远程控制系统，笔者设计一套基于Android移动终端的电子提花机远程控制系统的C/S模型，如下图所示。工作人员通过操控基于Android系统的移动终端，通过无线网络连接到局域网中，服务端提花机控制器通过有线网络连接到局域网中，与客户端进行网络通信，从而实现数据传输。当服务端接收到正确命令数据后，与提花机机器进行控制操作，进而操作提花机的工作。



系统C/S模型图

* + 1. **系统核心架构**

根据以上所提出的远程控制系统的C/S模型，在整个系统当中，实现的核心点在于Android终端上的应用程序与提花机控制器的应用程序的开发与实现，通过两端的控制程序来实现整个远程控制系统的核心部分。核心部分的结构图如下图所示。客户端利用Delphi语言进行Socket技术编程，服务端基于QT语言进行Socket编程来实现两端的网络通信。系统客户端是Android移动终端，可以通过无线WIFI连接网络到网络，服务端是基于核心板M3352嵌入式开发板，所搭载的系统是Linux系统，具备网卡，可以通过有线网络连接到网络。在同一个局域网内，两端可以通过Socket编程来进行网络数据传输。



系统核心架构图

* + 1. **系统工作原理与工作流程**

系统的核心部分在于控制端移动终端与服务端控制器之间的数据通信，处理这部分核心内容是本研究的重点。系统两端的数据通信是基于socket技术编程来实现的，在客户端中是通过Delphi XE中的TIdUDPServer类实现网络通信，而在服务端中是通过QT中的QUDPSocket类来实现的。两端通过各自的通信模块来进行数据发送和数据接收。TIdUDPServer类与QUDPSocket类都是基于网络通信技术Socket扩展而来的。Socket技术是进程间的通信技术，适用于网络环境中。作为一种网络通讯类的应用程序的通信编程接口，Socket技术为网络通信的发展带来很大的进步。本文采用基于UDP协议的数据报形式的Socket编程技术（SOCKET\_DGRAM），该技术定义一个无连接的服务，数据通过数据报进行传输，并且是无序，无可靠性保证的数据传输。（基于UDP协议的Socket网络编程\_周丽娟）基于UDP协议的Socket编程的框架如下图所示。



基于UDP协议的Socket编程框架

系统中客户端与服务端的连接是基于UDP协议的，跟TCP这种面向连接的协议一个不同的地方就是UDP协议是面向无连接的协议，在数据通信之前无需进行建立客户端与服务端网络连接，之间进行数据发送和接收。客户端只需要知道服务端的IP地址和所使用的端口就可以通过网络通信模块向服务端发送数据。

客户端与服务端在启动程序后，一直处于监听状态，当客户端有数据发送到服务端时，服务端对接收的数据进行处理，然后将处理结果反馈给客户端。服务端与客户端一直处于阻塞状态直到接收到对方发送过来数据。在这里所说的数据指的是客户端与服务端所进行的控制命令数据，操作人员通过在移动终端对应用程序的界面进行相应的操作，客户端的命令控制模块将操作控制信心整合成相对应的命令数据，当操作人员确认发送命令数据后，命令发送模块将之前整合的命令数据根据两端的数据协议解析成UDP数据报通过基于UDP协议的socket编程技术传输到服务端，服务端一直处于监听状态，当接收到客户端发送过来的数据后，通过网络通信转换模块将UDP数据报根据两端的数据协议解析成可读的命令数据，进而执行命令。



命令数据发送流程

本文所研究的远程控制系统是根据上述网络通信原理来实现的，操作人员手持移动终端，在确保服务端处于启动状态的情况下，并且已知服务端网络通信IP地址以及端口号之后，开启客户端远程控制应用程序，在服务端网络设置界面输入服务端IP地址与port参数，并且测试网络连接是否正常。在主控部分和设置部分进行一些控制操作，控制器端将这些操作通过命令控制模块转化为控制命令，等待操作人员的确认发送。操作人员确认发送后，服务端接收到控制命令后发送给控制端一个确认信号。系统工作流程图如图所示。



系统工作流程图

服务端的网络通信模块是由CUDPTransmission类完成的，该类拥有一个成员变量是QT中的QUdpSocket类的一个对象clientScoket，另外还有IP地址和端口两个成员变量，该类的方法有设置IP、端口的set()方法和发送数据的send()方法，与此同时，根据QT的信号槽机制，编写了两个信号函数sennm()、recvm()和两个槽函数sendmsg()、recvmsg()，将clientSocket当中的信号读取数据函数readyRead()连接到CUDPTransmission类的槽函数recvmsg()，并且将CUDPTransmission类的信号sendm()连接到该类的槽函数sendmsg()。在发送信号时用send()方法，该方法调用信号sennm()，这样就触发了槽函数sendmsg()，在这个槽函数中使用clientSocket进行数据的发送。当clientSocket接收数据时，触发了CUDPTransmission类的槽函数recvmsg()，该函数解析数据并触发信号recvm()。

在Delphi XE7中根据面向对象的编程思想，创建一个网络通信类（TCommunication），这个类一个重要的成员变量是Delphi里的TIdUDPServer类的一个对象IdUDPServer，该对象拥有发送UDP数据报和接收UDP数据报的方法。在封装的TCommunication类当中编写了设置Server端网络参数的方法setServerNet()、数据发送方法sendMsg()与接收的方法receiveMsg()、数据报处理的方法msgOp()，sendMsg()和receiveMsg()分别调用该类成员变量IdUDPServer的发送UDP数据报和接收UDP数据报的方法，msgOp()方法主要是在receiveMsg()方法中被调用，根据所接受的不同的数据ID去进行进一步的操作或者显示。

* 1. **远程控制系统服务端模块代码实现**

系统服务端是基于QT语言在QT Creator IDE中进行开发的，QT语言是基于C++的图形化框架，可以使用OOP的方式进行编程开发。本节根据上述的系统工作流程，对服务端进行模块化分析与设计，进而加以编程实现。

* + 1. **系统服务端模块分析与设计**

服务端提花机控制器要实现对电子提花机的完全控制，服务端的应用程序需要包含对电子提花机的工作过程中状态的设置，对电子提花机的工作管理，另外还有对电子提花机工作之前的测试等一些功能。根据这样的需求，将电子提花机控制器的控制功能分为三大部分，分别是主要控制、设置、工具。其中主要控制部分是包含提花机的几乎所有基本控制功能，包括查看进程信息、管理控制器、设置花型、设置选纬数、选纬前进、选纬后退、设置花型循环、查看状态信息、加载花型等。设置部分主要完成对控制器以及提花机的一些控制，包括加载提花机花型信息或者工作信息，操作文件，设置提花机参数，设置控制器网络参数、时间、语言，转换花型等。本研究对服务端控制器的功能需求进行详细分析，将服务端提花机控制器分为以下四个模块，如图所示。

* 用户界面模块：要实现控制器界面之间的事件机制，花型的输入与输出，连接功能，例如网络连接、USB连接等等；
* 花型和进程模块：主要实现在开发板中读取花型到RAM，编辑花型，转换花型数据的格；
* 配置模块：主要实现定义样本，设置织布机类型，设置某一个织布机的装备；
* 网络通信模块：主要实现开发板网络设置，远程客户端网络地址设置，数据报处理包括数据报的发送、接收和解析处理。



系统服务端模块图

* + 1. **系统服务端模块实现**

本文对服务端的开发所采用的IDE是QT Creator。利用QT Creator新建项目后，根据OOP思想，建立多个不同功能的类。根据不同的功能的类大致可以分为两种类型的类：界面类、逻辑操作类。逻辑操作类与C++语言所建立的类一样分为两种文件，那就.h后缀的头文件与.cpp后缀的CPP文件。而界面类除了包含于逻辑操作类相同的头文件和CPP文件之外，还有.ui后缀的界面文件。在建立了一个项目之后，项目文件夹下会自动新建一个.pro后缀的项目配置文件，在之后的建立不同类的同时，该配置文件也会跟随着不断的更新。后面会再详细叙述该配置文件。在对界面类进行开发过程中，往往会添加一些图片到界面中，比如按钮背景，窗口背景，应用程序logo等等，这个时候QT Creator将会将资源的配置存储在.qrc后缀的文件中。所以，在QT Creator所建立的项目中会包含五中文件，当然还有一个存储引用资源的文件夹。



QT Creator 项目文件结构图

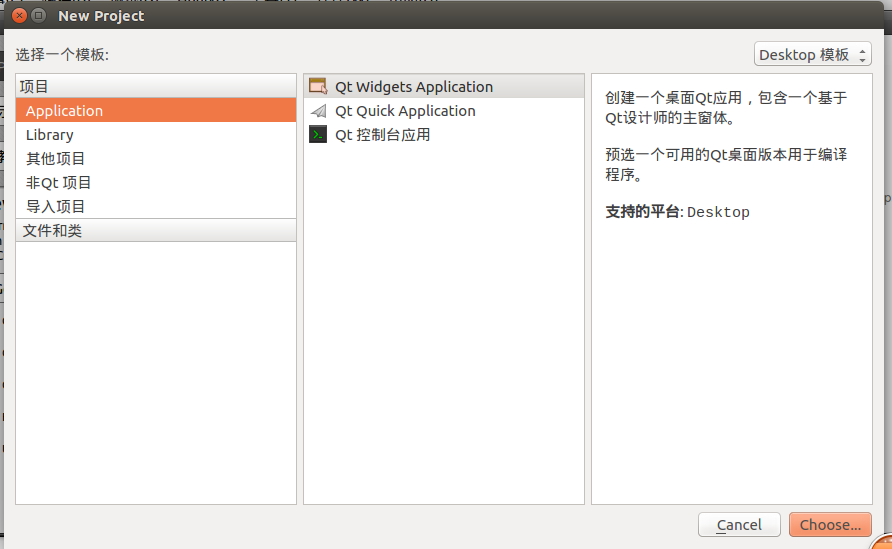
1. 通过QT Creator建立项目

本文对服务费的开发所采用的QT Creator是QT Creator3.4.2版本，基于第5版本的QT，编译器是GCC4.9.2。与其他大多语言IDE一样，在欢迎界面有一个新建项目（New Project）按钮，可以通过这个按钮来新建项目，或者可以在文件菜单中新建项目。



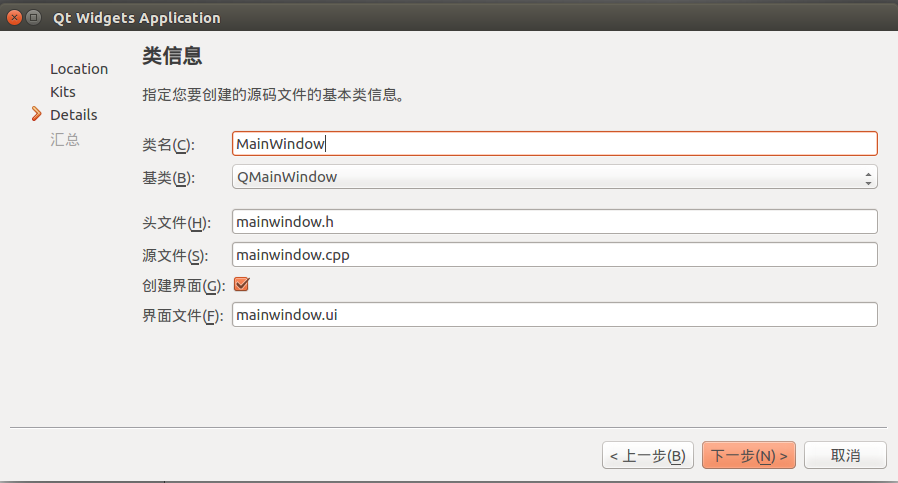
QT Creator新建项目

本文所建立的项目属于嵌入式GUI项目，所以新建Application下的Qt Widgets Application项目。这样所建立的项目会包含QT Creator所带的Qt Designer，这是一个可视化的开发界面，使得开发人员快捷开发应用界面。



QT Creator建立桌面应用

在建立项目之后，QT Creator会自动为开发人员建立一个基于QMainWindow的MainWindow的类，该类属于程序界面类，所以分别包含mainWindow.h、mainWindow.cpp、mainWindow.ui三个文件。



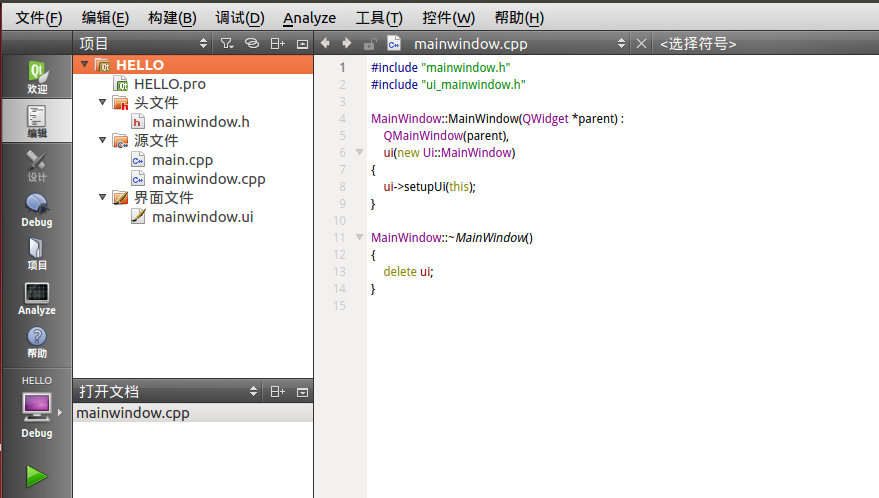
QT Creator新建项目自动生成MainWindow程序界面类

除了自动生成MainWindow程序界面类之外，QT Creator也会生成项目配置文件HELLO.pro（测试项目名称为HELLO）与项目执行主文件main.cpp，与C++项目一样，程序主线逻辑都在main.cpp文件中编写。



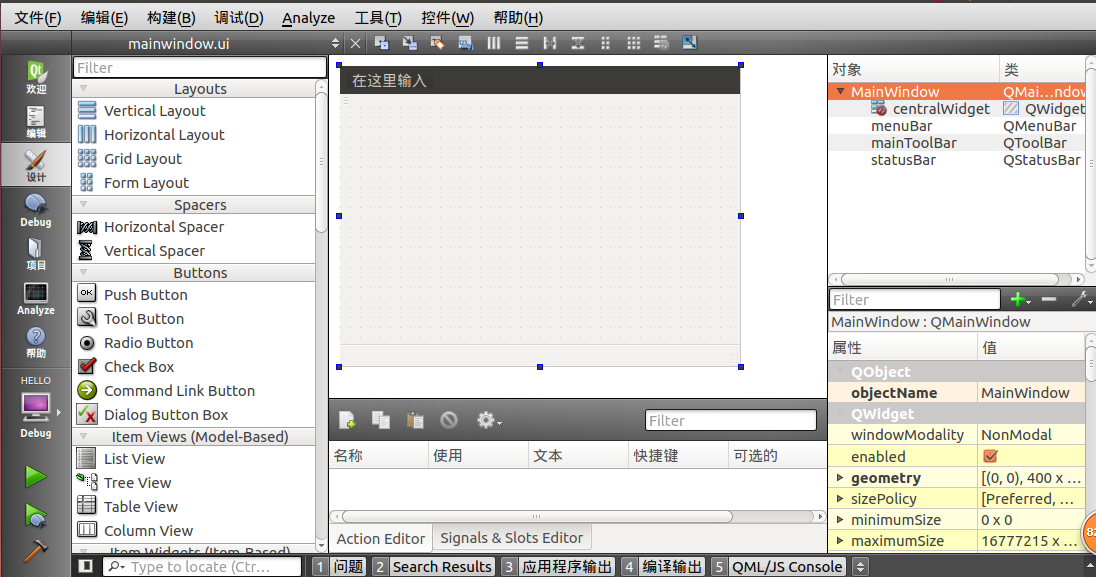
QT Creator新建项目汇总信息

在完成项目的建立之后，QT Creator会自动跳至mainwindow.cpp的开发界面。前面已经详细叙述了QT Creator所建立项目的文件体系，在界面左侧，IDE会自动将不同格式类型的文件放在一个虚拟文件夹中显示出来，方便开发人员查找。所有的.h头文件都在头文件夹内，所有的.cpp文件都在源文件夹中，所有的.ui文件都包含在界面文件夹中。



项目文件系统

在QT Creator中，编辑界面中可以进行代码编程，主要对.h和.cpp文件进行编写。在设计（QT Designer）界面中对.ui文件进行编辑，这里所说的编辑是指设计应用程序的界面。在QT Designer中进行图形化开发，可以从左侧的控件栏中选择所要的控件，拖进中间的编辑栏中，再在右侧的属性栏中对相应的控件的属性进行修改。对于控件的信号槽的编辑是在信号、槽编辑器中进行编辑。



QT Designer界面

1. 服务端用户界面模块

根据电子提花机的功能，将控制器的界面分为三大类：主要部分（MainPart）、设置（Seting）、工具（Tools）。三大类中包含着不同的功能控制界面，例如主要部分中包含设置花型（setPattern）、工作信息（work）、软键盘（keybord）等功能界面。这些功能界面也是一个个类，每个类都有自己的功能控件。下图介绍四大类所包含的控制功能界面。



控制器应用程序的控制界面结构

用户界面模块主要完成整个应用程序的界面设计和界面之间的逻辑跳转。项目中所有的界面都是面向对象中的一个类，拥有各自的属性和方法。下面以电子提花机类和主要部分类为例，来介绍这两个类的核心代码。

本文根据应用程序需求，将功能分为不同级别的功能类。一级功能类是CMainTooBbar，二级功能类是CMainPart、CSetting、CTools。二级功能类下面是三级功能类，比如属于CMainPart二级功能类的三级菜单类有CSetPattern、CWork、CKeybord,三级菜单类下面还有其他级别的功能或功能类。如下图所示。



系统服务端类结构图

根据需求，本文将不同级别的功能分为不同级别的类，但不同级别类之间是没有继承和被继承的关系的，可以说是所属关系——低级别的类的实例可能是上一级别类的成员变量。据上所述，一级类CMainTooBbar下有三个二级功能类，所以三个功能类的一个实例属于CMainToolBar的公共成员变量：mainpart、tools、setting。CMainToolBar还有一个私有成员变量，是自身的界面对象ui，在QT Creator中建立一个应用界面类时，该类都会包含一个自身ui对象的私有成员变量。另外，CMainTooBbar还有一个运行显示二级功能窗口的方法runningWidget(int widgetNum)。



一级类CMainToolBar类图

根据类图，CMainToolBar类的构造函数实现四个成员变量的定义，代码实现如下所示。

CMainToolBar::CMainToolBar(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::CMainToolBar)

{

ui->setupUi(this);

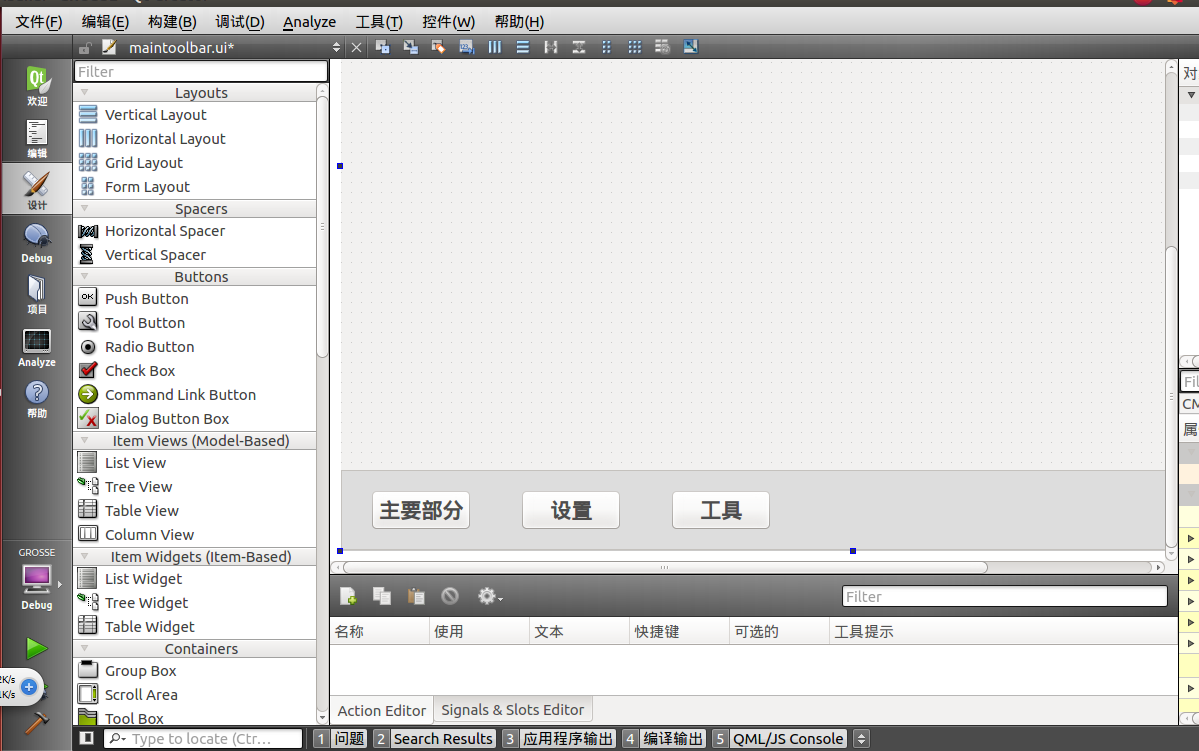
mainpart=new CMainPart(this);

tools=new CTools(this);

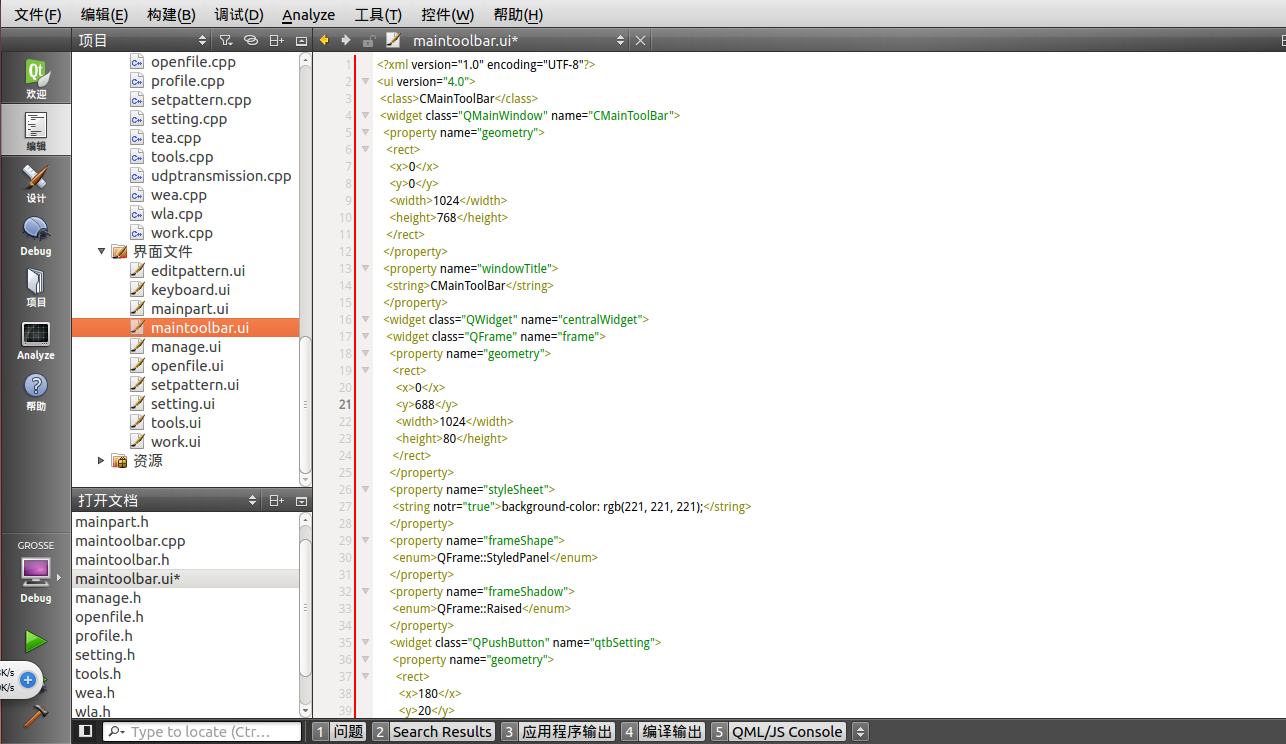
setting=new CSetting(this);

}

CMainToolBar类是用户界面类，所以除了.h和.cpp文件之外还有.ui文件，.ui文件是在QT Designer中进行界面构建时，IDE自动生成的文件。如下图所示，是CMainToolBar类的用户界面，所生成的.ui文件是xml文本，项目在编译时会解析xml文本，将xml文本解析成图形界面。.ui文件如图所示。



CMainToolBar类的用户界面



maintoolbar.ui文件

上图当中的三个按钮的点击事件分别触发以下三个事件：

CMainToolBar::on\_qtbMainpart\_clicked()；

CMainToolBar::on\_qtbSetting\_clicked()；

CMainToolBar::on\_qtbTool\_clicked()。

这三个事件都会调用CMainToolBar类的成员函数runningWidget(int widgetNum)，该函数根据参数不同显示不同的二级功能的界面。具体实现看下面的代码展示。

void CMainToolBar::runningWidget(int widgetNum)

{

switch (widgetNum)

{

case 0:

mainpart->setWindowFlags(Qt::FramelessWindowHint);

mainpart->move(0,0);

mainpart->show();

mainpart->raise();

break;

case 1:

setting->setWindowFlags(Qt::FramelessWindowHint);

setting->move(0,0);

setting->show();

setting->raise();

break;

case 2:

break;

case 3:

tools->setWindowFlags(Qt::FramelessWindowHint);

tools->move(0,0);

tools->show();

tools->raise();

break;

default:

break;

}

}

主要部分包括设置花型、工作信息和软键盘，这三个又是三个界面类，所以将这三个界面类的一个实例作为主要部分的成员变量。主要部分类CMainPart类图如下图所示。



主要部分类CMainPart类图

在主要部分的界面上有输入框，服务端所使用的是基于嵌入式的Linux系统开发板，没有外接键盘。控制器使用的显示器是液晶触摸显示器，所以需要在主要部分界面内开发出一个软键盘。在主要部分中，有工作信息、设置花型两个三级功能，所以这两个三级功能类的实例是主要部分类的成员变量。除此之外，主要部分类还有一个成员函数eventFilter()，该函数主要判断软键盘主要部分界面的聚焦是否聚焦在输入框内从而显示出键盘。另外还有几个槽函数：newValue()、deltext()、on\_qtbResetPattern\_clicked()、on\_qtbPatterManage\_clicked()、on\_qtbJobInfo\_clicked()。前两个是实现输入框的逻辑操作的槽函数，后三个是界面按钮的槽函数。

CMainPart类的构造函数代码如下所示。首先是创建UI界面，然后是定义三个成员变量。其中connect()，方法是QT所带的信号与槽的连接函数，是信号槽机制所定义的一个方法，当信号函数(SIGNAL)发出时，就会触发槽函数（SLOT）。

CMainPart::CMainPart(QWidget \*parent) :QMainWindow(parent),ui(new Ui::CMainPart)

{

ui->setupUi(this);

keyboard=new CKeyBoard(this);

pSetPattern=new CSetPattern(this);

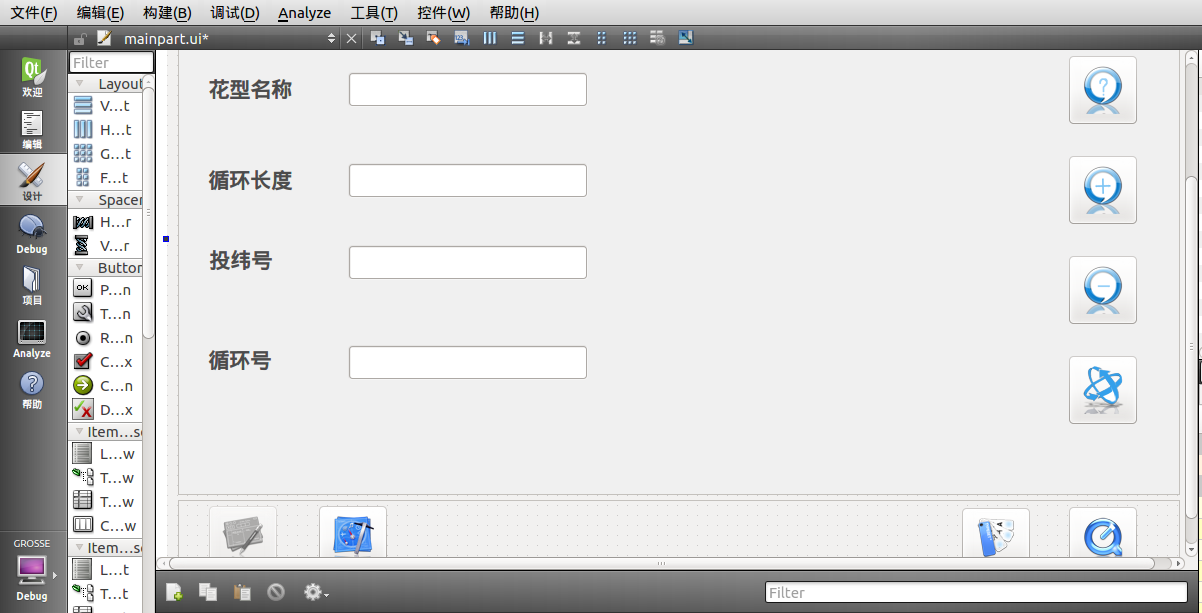
pWork=new CWork(this);

connect(keyboard,SIGNAL(newValue(QString)),this,SLOT(newValue(QString)));

connect(keyboard,SIGNAL(delValue()),this,SLOT(deltext()));

}

CMainPart类的UI界面如下图所示。



CMainPart类的UI界面

CKeybord类的类图如图所示，其中CapsLock是指是否大写，Number指的是是否切换到数字模式。该类拥有两个信号函数：newValue(QString value)和delValue()，分别表示新增一个值和删除一个值。该类的槽函数有很多，是键盘界面上的每个按钮所对应的槽函数。CKeyboard类实现的UI如下图所示。



CKeybord类的类图



CKeyboard类实现的用户键盘界面

当软键盘中的非特殊按钮触发click事件时，该事件会自动触发newValue()信号函数：

void CKeyBoard::on\_qtb\_O\_clicked()

{

emit newValue(ui->qtb\_O->text());

}

在CMainPart中，该信号函数连接了CMainPart类的newValue(QString)槽函数，从而实现了字符的输入，代码如下：

void CMainPart::newValue(QString value)

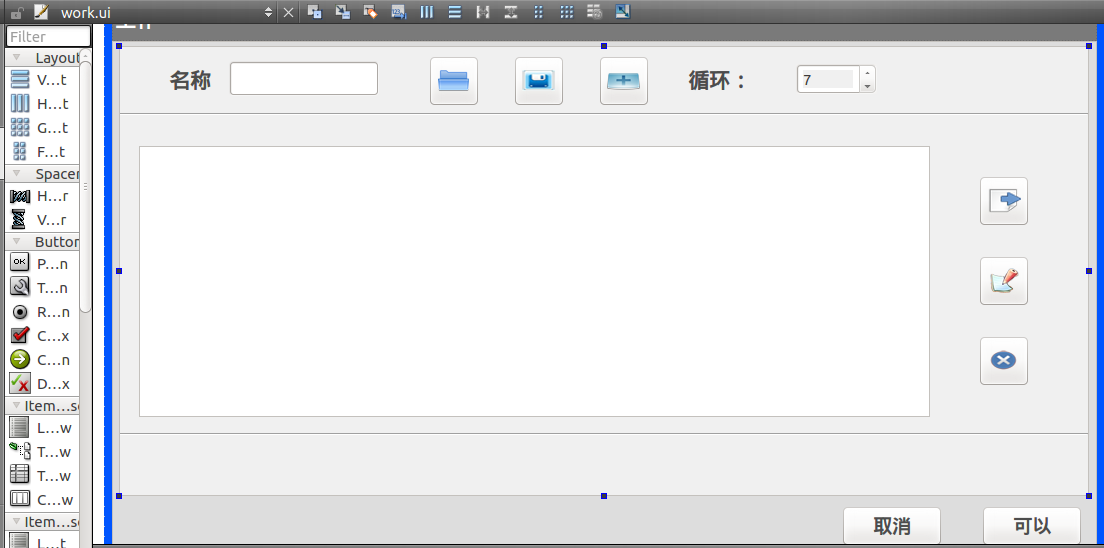
{

QWidget \*focusWidget=this->focusWidget();

((QLineEdit \*)focusWidget)->insert(value);

}

主要部分中还有工作信息二级功能，CWork类的界面如图所示。



CWork类的界面

界面中有一个表格（table）控件，需求设置其尺寸属性以及单行选中，在CWork类中用成员函数setTable()实现上述需求。代码如下所示。

void CWork::setTable()

{

ui->tableWidget->setColumnCount(5);

ui->tableWidget->setRowCount(10);

ui->tableWidget->setColumnWidth(0,25);

ui->tableWidget->setColumnWidth(1,125);

ui->tableWidget->setColumnWidth(2,75);

ui->tableWidget->setColumnWidth(3,75);

ui->tableWidget->setColumnWidth(4,75);

for(int i=0;i<10;i++)

{

ui->tableWidget->setRowHeight(i,25);

}

//选中一行

ui->tableWidget->setSelectionBehavior(QAbstractItemView::SelectRows);

//一次只能选中一行

ui->tableWidget->setEditTriggers(QAbstractItemView::NoEditTriggers);

ui->tableWidget->verticalHeader()->setVisible(false);

ui->tableWidget->horizontalHeader()->setVisible(false);

ui->tableWidget->setShowGrid(false);

}

CSetting类主要实现对文件的一些操作，其中包含openfile成员变量。file是COpenFile类的一个实例。COpenFile类的类图如下所示。



COpenFile类的类图

COpenFile类主要实现打开文件，将文件显示在table中。类图中的三个成员函数和一个槽函数showDirFollow()的具体代码实现如下所示。

void COpenFile::showDir(QDir dir) //显示当前目录dir 下的所有文件

{

QStringList string;

string<<"\*";

QFileInfoList list=dir.entryInfoList(string,QDir::Dirs|QDir::Files,QDir::DirsFirst);

showFileInfo(list);

}

void COpenFile::showFileInfo(QFileInfoList list) //显示目录,文件

{

ui->tableWidget->clearContents();

for(int index=0;index<list.count();index++)

{

QFileInfo tmpFileInfo=list.at(index);

if(tmpFileInfo.isDir())

{

QIcon icon("dirfilePicture/dir.png");

QString fileName=tmpFileInfo.fileName();

qint64 tmpSize=tmpFileInfo.size();

QString fileSize=QString::number(tmpSize);

QTableWidgetItem \*nameItem=new QTableWidgetItem(icon,fileName);

//QTableWidgetItem \*sizeItem=new QTableWidgetItem(fileSize);

ui->tableWidget->setItem(index,0,nameItem);

//ui->tableWidget->setItem(index,1,sizeItem); //目录大小不作处理

}

else if(tmpFileInfo.isFile())

{

QIcon icon("dirfilePicture/file.png");

QString fileName=tmpFileInfo.fileName();

qint64 tmpSize=tmpFileInfo.size();

QString fileSize=transSize(tmpSize);

//QString fileSize=QString::number(tmpSize2);

QTableWidgetItem \*nameItem=new QTableWidgetItem(icon,fileName);

QTableWidgetItem \*sizeItem=new QTableWidgetItem(fileSize);

ui->tableWidget->setItem(index,0,nameItem);

ui->tableWidget->setItem(index,1,sizeItem);

}

}

}

void COpenFile::showDirFollow(QTableWidgetItem \*item) //根据当前选择进入下一目录

{

QString dirName=item->text();

qDebug()<<dirName;

QDir dir;

dir.setPath(path);

//path+="/"+dirName; //此处为路径,须有“/”

qDebug()<<path;

dir.cd(dirName);

path=QString(dir.absolutePath());

qDebug()<<path;

showDir(QDir(path));

}

//将文件大小转化为B,K,M,G函数

QString COpenFile::transSize(qint64 size)

{

unsigned char ch ='B';

qint64 currentSize=size;

if(currentSize>1024)

{

currentSize/=1024;

ch ='K';

if(currentSize>1024)

{

currentSize/=1024;

ch ='M';

if(currentSize>1024)

{

currentSize/=1024;

ch ='G';

}

}

}

QString result=QString::number(currentSize)+QString(ch);

return result;

}

这一部分主要介绍了用户界面模块的实现，

1. 服务端花型和进程模块
2. 服务端配置模块
3. 服务端网络通信模块

在QT Creator中开发应用程序时，一个项目分为四个子文件夹和一个项目配置文件，子文件夹分别是头文件夹、源文件夹、界面文件绝爱、资源文件夹。头文件夹下包含所有的类的.h文件，在源文件夹下包含所有的.cpp文件，在界面文件夹下包含所有的.ui文件。资源文件夹存放着项目应用程序中所需要的图片资源

* 1. **远程控制系统客户端实现**
     1. **系统客户端模块设计与实现**

系统客户端主要实现连接服务端，并且通过数据传输实现控制服务端控制器，通过命令数据间接控制提花机的工作。根据上述的功能需求，笔者将控制端分为以下四个模块。如图所示。

* 用户界面模块：实现应用程序界面之间的事件机制，花型的输入输出；
* 命令控制模块：将界面之间的操作、界面内部的操作转化为相应的数据报以待发送；
* 命令发送模块：将等待发送的命令数据报通过网络通信发送到服务端；
* 数据处理模块：对服务端反馈的数据报进行处理并在相对应的用户界面进行显示或者操作。



系统客户端模块图

* + 1. **系统客户端模块实现**

1. **远程控制系统的客户端的设计与实现**
   1. **远程控制系统服务端需求分析**
   2. **远程控制系统服务端设计模型**
   3. **远程控制系统服务端实现**
2. **基于移动终端的电子提花机远程控制系统的实验测试与数据分析**
3. **总结与展望**