2017



**CanTool Windows App**

**设计说明书**

现代软件工程第五组：

组长：2017218044 王若琪

组员：2017218053 杨丽芸

2117218010 李林峰

2117216024 焦年红

指导教师：章亦葵

# 项目介绍

## 项目背景

在现代汽车控制技术中，汽车中会使用多个电子控制装置（ECU：Electronic Control Unit）对整车进行控制。而ECU之间的信息交换更多地依赖于CAN(Controller Area Network)总线的网络连接方式来完成。为了检测和控制CAN bus的信息内容，需要使用CAN bus检测设备。CanTool装置是完成CANbus检测的工具。为了实现CAN数据的显示及控制，需要使用本文提出的CanToolApp软件。因此，我们开发了CanToolApp的Window App，能够帮助用户查看CAN数据信息、控制CAN装置。

## 项目介绍

CanToolApp能够帮助用户直观地了解CAN bus检测的信息。该软件可以将连接在CAN总线上的CanTool装置采集的CAN信息发送到上位机（这里是Windows PC），并由运行在上位机中的CanToolApp软件接收这些信息，显示在用户图形界面上。同时在CanToolApp的界面上还可以设定CAN信息，通过GUI按钮将信息发送给CanTool装置，CanTool装置将按照规定的信息格式，将信息发送的CAN总线上。CanTool装置与上位机通过串口进行通信。

于是，在这份报告中，我们对本组开发的CanToolApp进行了更为详细的分析，首先，我们对业务进行了需求分析，以帮助我们更为深入和全面地理解业务需求，进行下一步开发，其次，我们对系统进行总体分析和设计，再次，对各个模块进行详细设计。

# 团队分工

在现代软件开发过程中，团队分工必不可少，下面先介绍小组成员的分工情况，见表2-1。

表2-1 成员分工表

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 学号 | 角色 | 分工 |
| 王若琪 | 2017218044 | 组长 | 分析作业要求，组织项目进展，并组织讨论、分配任务。在开发过程中主要负责接收和发送CAN信息并进行数据相关处理，文件操作相关工作以及最后的代码整合工作。在文档编写中主要负责设计说明书和配置安装及用户手册，整合文档。 |
| 杨丽芸 | 2017218053 | 组员 | 参与讨论，完成自己的任务。在开发过程中主要负责设置CAN装置部分。在文档工作中负责单元测试文档以及文档的校正审核工作。 |
| 李林峰 | 2117218010 | 组员 | 参与讨论，完成自己的任务。在开发过程中主要负责设置串口与串口通信部分。开会过程中负责会议记录，在文档工作中编写会议记录文档。 |
| 焦年红 | 2117216024 | 组员 | 参与讨论，完成自己的任务。在开发过程中主要负责前端界面的开发。还负责了对程序的测试工作。 |

# 需求分析

## 3.1 功能性需求分析

根据CanToolApp开发的要求（Problem Statement文档），我们通过讨论，得到了CanToolApp的相关需求内容：

①设置串口并可以进行通信；  
② 设置CanTool装置；  
③解析并实时显示Can信息；  
④选择仪表盘方式；  
⑤显示实时物理值曲线；  
⑥保存数据文件格式为CSV或自定义；  
⑦指定发送的CAN信息；  
⑧发送信息给CanTool装置；  
⑨完成CAN信号数据解析及CAN发送信号合成；  
⑩提供树状结构显示Can信息；  
⑪Can信息和信号另存为其他格式；  
由相关需求，我们可以得到用户基本的业务用例如下图所示：

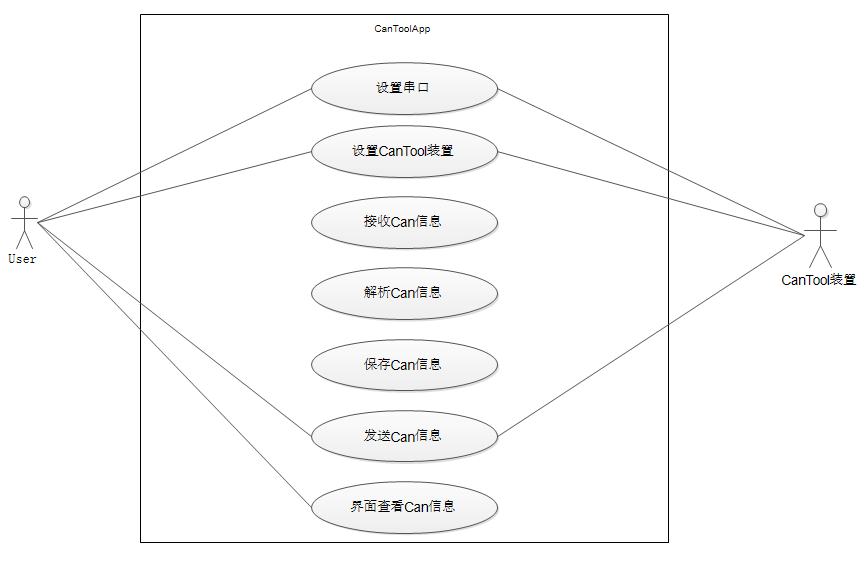


图3-1 业务用例图

根据业务用例图，我们将整个系统进行了功能分解，得到功能分解图，将CanToolApp划分为各大功能模块。

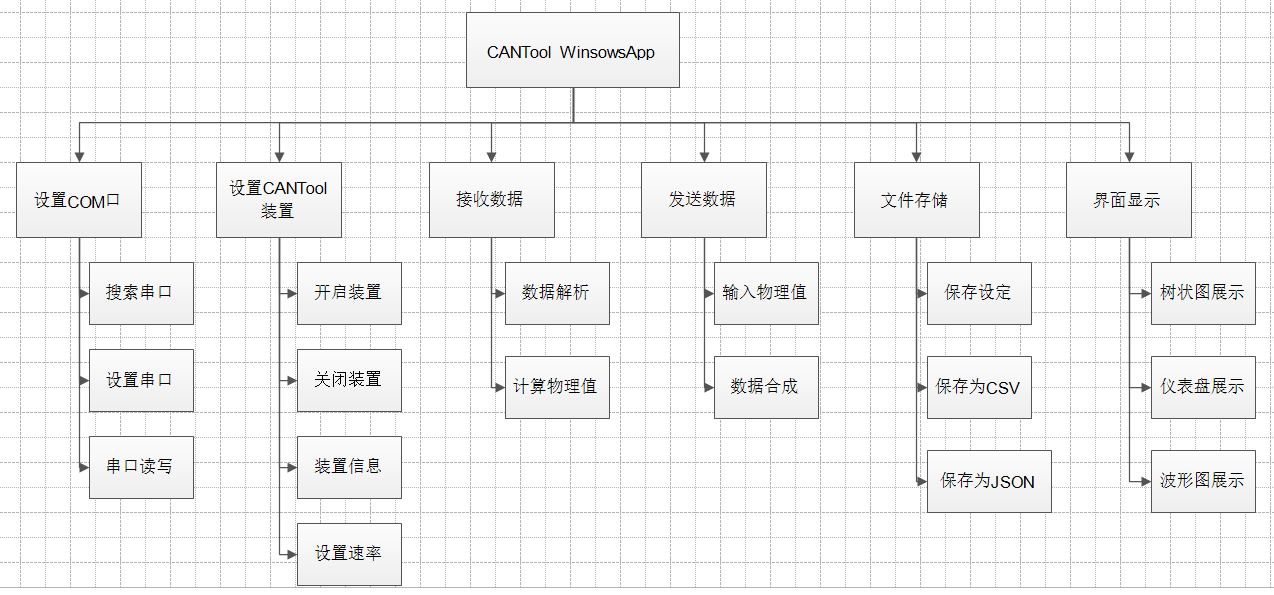


图3-2 功能分解图

## 3.2 非功能性需求分析

除了功能性需求，为了满足用户业务需求，系统也必须具备一些非功能性需求。本节对本系统的给功能性需求进行一些说明。

表3-1 非功能性需求表

|  |  |  |
| --- | --- | --- |
| 编号 | 名称 | 条目 |
| 1 | 性能需求 | 能够接收数据较为快速的传输，程序不崩溃。 |
| 2 | 可用性 | 软件的布局设置合理，用户容易上手，可以进行操作 |
| 3 | 可靠性 | 软件应有一定的容错性，在接收到非法的字符串时不崩溃，不能解析某条数据时也不崩溃。 |
| 4 | 支持性 | 能够支持虚拟CANTool装置和真实CANTool装置的使用。 |
| 5 | 可移植性 | 能够通过安装包安装在其他电脑上正常使用。 |

# 总体设计

首先根据需求分析和功能分解，我们可以对系统整体的设计进行分析，进行流程上的把握，讨论实际开发的方法。

## 系统流程

通过对功能的分析归纳出系统大致的流程图：

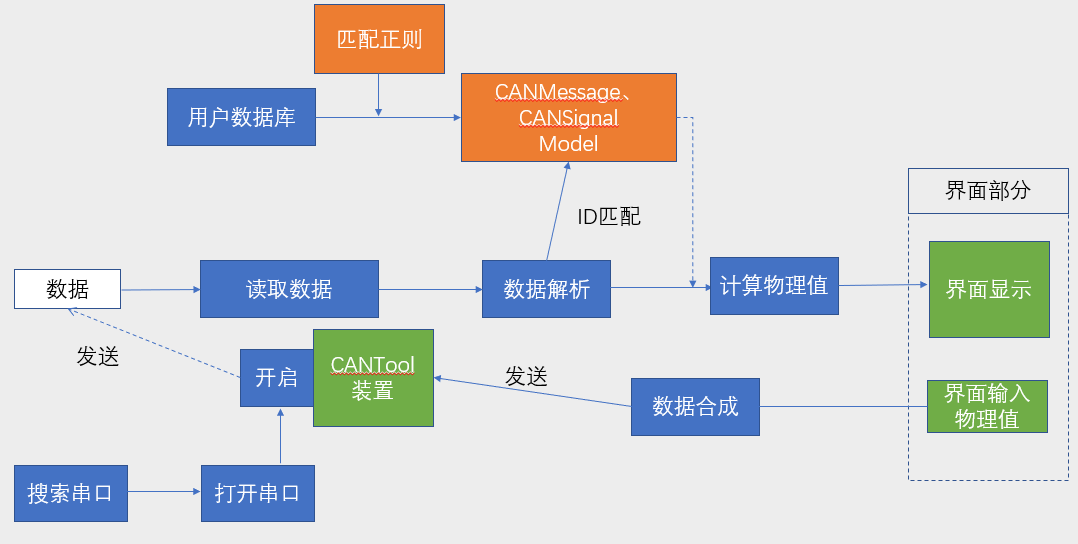


图4-1 系统流程图

在实际使用中，活动图如下：

## 4.2 开发方法

经过我们的讨论，在本项目中，我们采用面向对象的分析方法和面向对象的设计方法。最终确定使用的开发语言为JAVA，使用的开发工具为MyEclipse和Eclipse，在开发过程中引入的Jar包有SerialPortIO.jar、RXTXcomm.jar、gson.jar、javacsv.jar，采用的相关技术有正则表达式等。在开发过程中我们的关键步骤是：

* 确定类及相关属性
* 识别业务对象到类
* 完善对象模型
* 识别用例和事件到具体方法

最终我们将系统分为五个模块来开发，分别是设置串口模块、设置装置模块、接收发送数据模块、文件存储模块、前端界面模块。

# 5.接口设计

## 5.1 内部接口

本系统主要是通过静态变量的方法，让各处的方法去调用同一个变量，主要是用在对数据库文件进行解析，通过正则表达式进行解析后，将解析出的CANMessage和CANSignal存在静态变量messageList和signalList中，和在系统接收到传输来的数据后，将解析后的结果存在静态变量resultList中，保证系统的各个模块调用的都是同一个CAN信息信号集，同一个结果集。通过这种方法在系统内部之间传递。

当然，除了使用静态变量的方法，也用了一些基本的方法，比如设置get、set方法，可以取到对象内部的属性。另外与本系统特定功能无关的工具类，统一放在tool包里并提供相关方法的接口方便直接使用。也能实现外部引入的jar包中的接口。

## 5.2 用户接口

系统给用户提供了可以操作的界面，可以通过图形界面对系统的各项功能进行操作，重要的界面如下图：

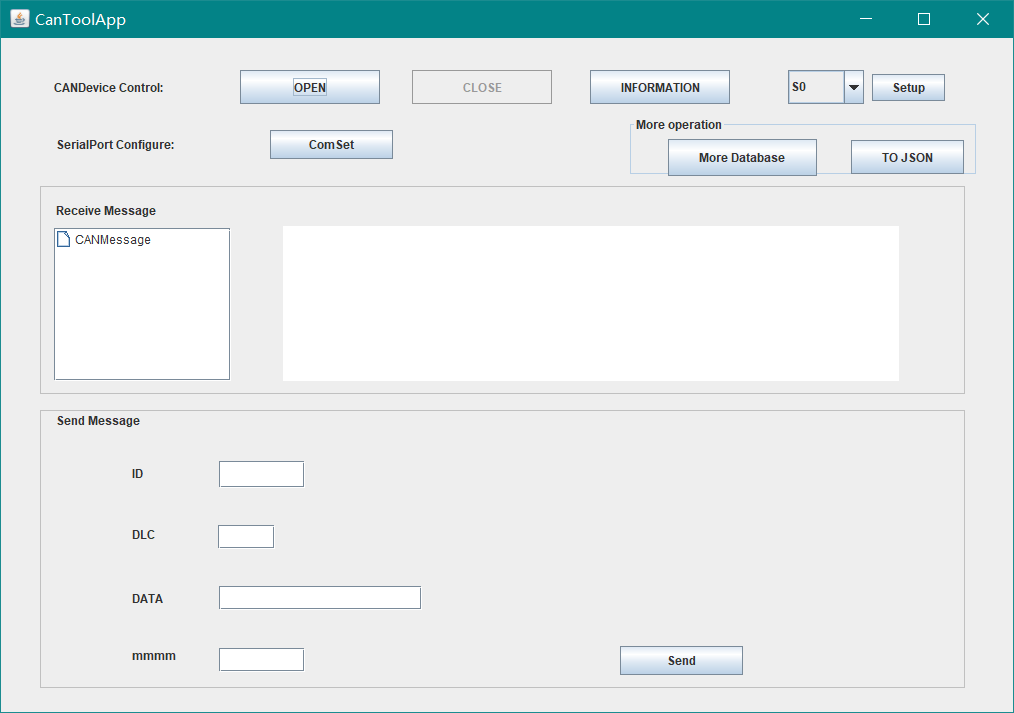


图5-1 主界面

在此界面上可以操作设置串口，设置CAN装置，发送CAN信息。

具体设置串口的参数的界面如下：

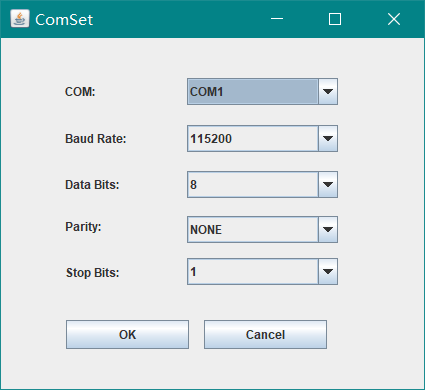


图5-2 设置串口界面

接收并显示CAN信息信号物理值的界面如下：

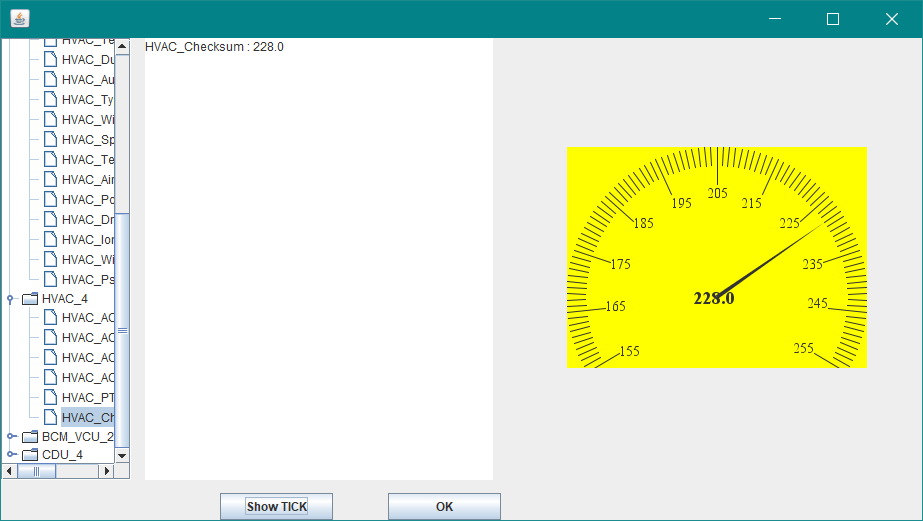


图5-3 显示物理值界面

# 6.各模块详细设计

## 6.1 设置串口模块

设置串口模块主要实现了搜索本机可用的串口，设置并连接相关串口，通过串口传输数据等功能。

在本模块中主要使用了SerialPortIO.jar、RXTXcomm.jar两个jar包里提供的方法，serialTool是静态变量，每次启动程序只创建一个。findPort()方法负责搜索本机的串口，openPort()方法负责打开相应串口，sendToPort()方法负责发送数据，readFromPort()方法负责读取数据，对串口设置监听SerialListener。本模块相关的类图如下：

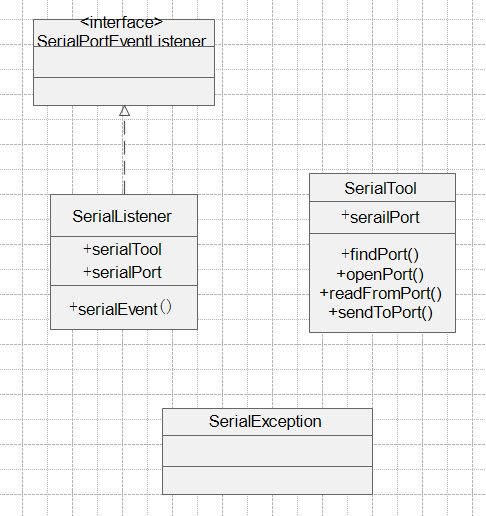


图6-1 设置串口模块类图

## 6.2 设置装置模块

设置CAN装置模块主要实现了通过CanToolApp打开CAN装置、关闭CAN装置、查看CAN装置的版本信息、设置CAN总线速率等功能。

本模块主要是通过在Main类中添加按钮的鼠标点击事件监听实现的，也就是点击“OPEN”按钮时即通过sendToPort()方法发送“O1\r”，点击“CLOSE”按钮时即通过sendToPort()方法发送“C\r”，点击“INFORMATION”按钮时即通过sendToPort()方法发送“V\r”，并且能接收到装置返回的版本信息，点击“Set Up”按钮时即通过sendToPort()方法发送下拉框选择的值。除此以外，还设置了按钮可用或者不可用的判断。

## 6.3 接收发送数据模块

本模块实现的功能是本系统的重点功能，是将得到的数据通过数据处理计算出用户需要了解的物理值，和由用户输入数据发送出去。

在接收数据部分，本模块中首先对用户提供的数据库进行解析，通过按行读取文件并用正则解析式解析的方法，将文件中以“BO”开头的行转化为CANMessage对象，将“SG”开头的行转化为CANSignal对象，然后分别存入messageList、signalList里，这两个list是静态变量，只在程序运行时创建一个。然后对接收到的数据进行解析，这里加了一个缓冲区，并且以“\r”对数据进行分解，得到一条数据，然后再解出ID、L、DATA，对这一条数据，以ID去messageList里查找，并根据相应的CANSignal在DATA中取出数据，这里把DATA的16进制字符串转化成了二维数组，再通过数据格式是“0+”还是“1+”，根据startPos和BitLength取成代表二进制数的字符串，然后再计算物理值，CANSignal和物理值的数组存入resultList里，这就是数据解析的结果集，可以在前端界面需要显示时调用。主要是dataprocess包里。这一部分的类图如下：

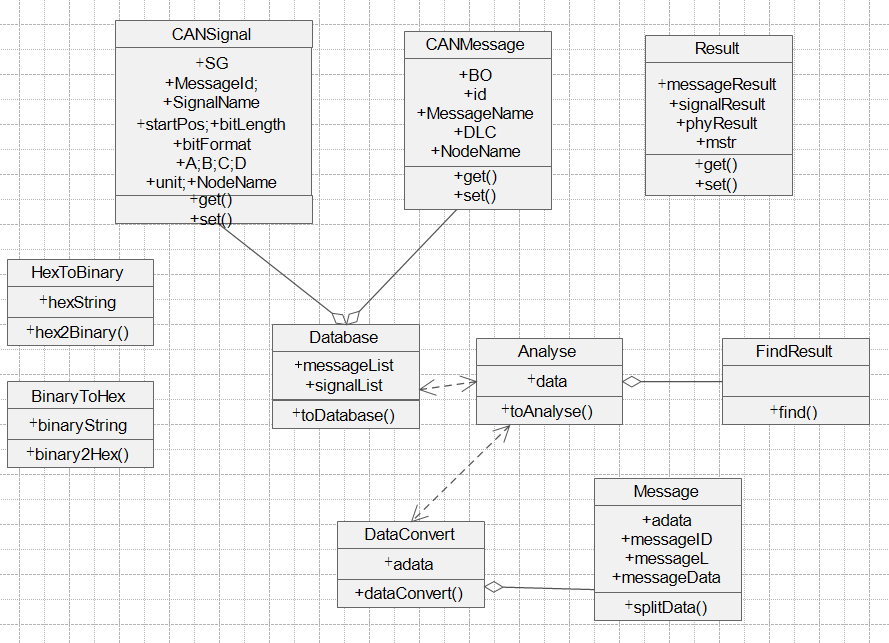


图6-2 数据解析类图

在发送数据部分，是由用户在界面上填写ID、L、DATA、mmmm，再将它们拼接成字符串，通过sendToPort()方法发送出去。

## 6.4 文件操作模块

本模块主要实现的功能是将用户数据库转换为JSON格式，将解析好的数据存成CSV格式和JSON格式的文件，方便用户保存记录之后查看，用户还可以选择其他数据库文件更新数据库。

将用户数据库转换为JSON格式，是将messageList和signalList里的对象一一取出，利用gson.jar包提供的toJson()方法，存成JSON对象。将解析好的数据存成CSV格式和JSON格式，是编写了两个类放在tool包里作为工具类，每解析出一条数据就写入对应文件。通过点击“More Database”按钮，在弹出的文件选择器中选择另外的文件，就以同样方法读取此文件并转换为新的messageList、signalList，之后再接收到的文件就以本数据库来解析了。本模块的类图如下：

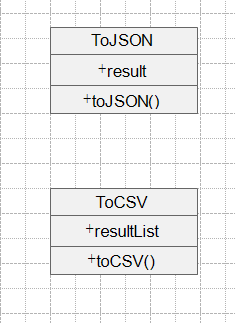


图6-3 转换为json和csv格式的相关类

## 6.5 界面模块

本模块可以分成两个部分，一是给用户提供操作的接口，比如提供设置串口、装置的按钮，二是将解析好的数据，主要是物理值和单位，以各种形式展示出来，方便用户查看。

主要有三个界面，一个是主界面，用户可以点击按钮，给按钮加了鼠标点击的事件响应，来进行响应的操作，一个是设置串口的界面，可以在此界面调整串口名、波特率等，一个是展示解析完成数据的界面，主要是用到了树和仪表盘，树是可以通过update()方法动态更新的，每解析成功一条数据就update()一次，点击对应结点将此结点的信息显示在文本框中，方便用户查看。仪表盘（Tick类）是点击一个signalName后，将该物理值显示在仪表盘上。

# 7.测试结果

在软件开发完成后，我们通过了组内的单元测试和测试组（9组，26组）的功能测试，具体测试内容见“单元测试文档”和由刘敬成组提供的测试报告。

# 8.总结

本报告介绍了根据CanToolApp的相关需求，我们组开发CanTool Windows App的相关情况。首先我们通过讨论得出了系统具体的业务需求，然后经过对系统整体的设计和每个模块的详细设计，进行编程实现，使用迭代开发的方法，最终开发完了CanTool Windows App，符合业务要求，实现了设置串口、设置CanTool装置、接收发送数据等功能。整体运行效果良好。

在项目的开发过程中，我们小组的同学一起研究发现问题、学习开发技术，也向其他小组的同学请教了许多，最终能够开发完成这个项目，离不开大家的努力和其他同学的帮助，在此也非常感谢。当然我们也存在着很多不足，系统还有可以改进的地方，比如实时曲线功能我们没有能完成，一是时间安排的问题，一是开发能力的问题，以后需要注意这些问题并不断提高能力。我们应当把这次课程的经历作为一笔宝贵的财富，在其中归纳总结，努力提升自己的水平。“学如逆水行舟，不进则退”，在开发技术更新发展迅速的今天，需要保持不断进步。