

# 某型飞控组件综合测试软件的设计与实现

## Design and Realization of Testing Software for a Flight Control System

中国空空导弹研究院 符冰

**[摘要]** 飞行控制组件是某型号产品的控制核心,其检测的通用性、可靠性在产品研制过程中有着重要的作用。介绍了基于面向对象方法的某型号飞控组件测试软件的开发思想、测试算法和具体实现。此软件可按照特定算法控制硬件自动对飞控组件的各内外接口进行测试,具有友好的界面、良好的可维护性、可扩展性,目前已成功应用于某型号飞控组件的测试中。

**关键词:** 飞控组件 测试软件 多线程

**[ABSTRACT]** Flight control system is the role of a style of missile. It is very important of its testing in the life of research of the missile. The design principles, key arithmetic and implementation method for testing software are introduced. The software are introduced can control the hardware to test the interface of control system automatically. The software has the characteristic of friendly interface and good maintainability. Now the software is used in the test of flight control system successfully.

**Keywords:** Flight control system Testing software Multithreading

某型号飞行控制组件是整个导弹系统的控制核心,在导弹的稳定控制、航迹规划等方面起着重要的作用,同时它也是各种信号的交联中心,对其可靠性、稳定性要求极高,因此如何对飞控组件进行全面充分的测试就显得比较重要。而整个完善的测试系统离不开测试软件的配合,本文介绍了基于 VC++6.0 开发的某型飞控组件自动化测试软件的组成、测试算法、关键技术及具体实现。

### 1 测试原理

由于飞控组件功能复杂,对外接口形式众多,因此对其测试时所需的硬件模块也较多,主要包括 1553B 通信模块、429 通信模块、422 通信模块、AD/DA 采集模块、高速并行数据接收通道、电源控制模块等,这些硬件模块放在一个 PXI 的工控机箱中,配合特定的软件来完成对飞控组件的测试,如图 1 所示。产品的测试就是通过测试软件对这些硬件板卡的操作来模拟载机、GPS、舵机等外部环境,并按照特定的测试算法,自动完成对飞

控组件各功能的测试。整个测试流程由综合测试软件控制,驱动软件是硬件和测试软件之间的桥梁。

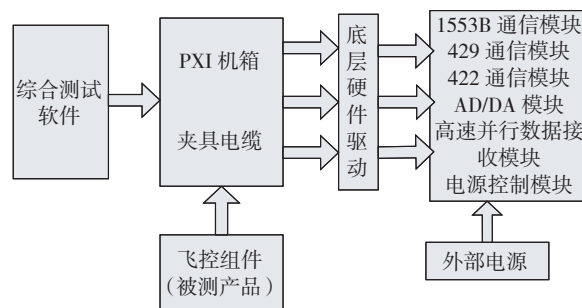


图1 飞控组件测试原理框图

Fig.1 Principle of flight control system

### 2 软件总体设计

某型号飞控组件测试软件是一个自动化的测试软件,根据测试的需求,结合飞控产品的工作特点,按照模块化的思想设计软件的各测试模块。软件在功能上应能满足:(1)合理控制试验流程;(2)能记录并保存试验过程中的各种数据;(3)能对数据进行分析处理;(4)对一些关键的变量能在屏幕上显示并给出评估结果;(5)对产品的测试应全面、可靠,降低虚警率;(6)具有友好的操作界面、人机接口。

利用 C++ 语言面向对象的技术进行系统开发,对数据结构的定义、模块的划分、接口的定义均体现在类及其层次关系内,这样可以提高软件设计的安全性、可靠性和可维护性。

根据模块化的设计思想,软件主要可分为以下模块:对外接口模块、数据采集处理模块、测试数据显示模块、测试流程控制模块、硬件驱动模块,如图 2 所示。测试流程控制模块控制测试整个流程,包括对产品上电,发送、接收数据,对数据的处理和流程控制等;数据显示模块用于对采集到的数据进行显示、管理并按照预先设定的规则进行正确性校验;数据采集处理模块是将测试过程中的数据以特定的格式存储起来以供事后分析;接口模块用于将底层的函数进行打包、处理供上层函数调用;驱动模块在底层工作,用于和硬件打交道,进行数据读写。

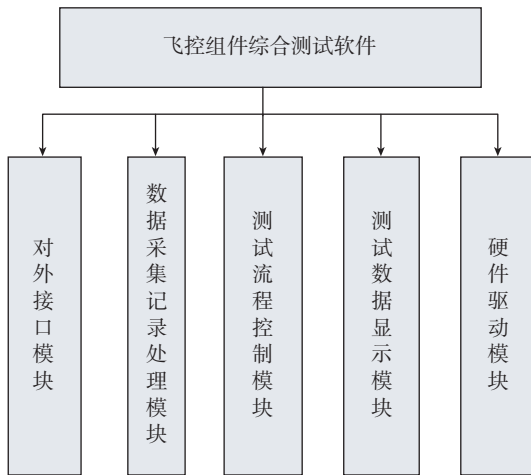


图2 测试软件总体结构  
Fig.2 Overall structure of testing software

### 3 测试算法

某型飞控组件的测试主要分为3个测试项目,分别为内外接口测试、传感器精度测试、导航精度测试,这些测试项目从不同的侧重点对产品的各部分功能进行测试。

#### 3.1 内外接口检测算法

内外接口测试主要对飞控组件的各种对内对外接口进行测试,在飞控软件专用测试模块的配合下完成对各接口通道进行测试。主要流程如图3所示。

- (1)设备初始化,各种初始参数赋值;
- (2)为产品上电,检测产品电压和电流是否正常;
- (3)设备向产品发送飞机描述,并接收产品返回的悬挂物描述,若正确,表明握成手功;

(4)检测 1553B 通道和相应的存储器;测试设备向飞控组件发送任务加载数据块和传递对准参数块<sup>[1]</sup>,工作程序将这些数据块分别转发到任务数据查询块和悬挂物监视数据块,通过比对若都正确则认为飞控组件与设备的 1553B 通道通讯正确,同时也完成了对存放这些数据块的缓冲区的检测;

(5)检测 429 发送和接收通道以及遥测通道:设备通过 429 总线分别发送测试数据,工作程序接收到这些数据后,同时通过 429 发送通道和遥测通道将这些数据转发出来,然后测试设备接收这些数据并判断与发送的是否一致,若有错误,则错误则计数加 1,并将最终结果显示在屏幕上;

(6)检测 IO 指令通道:通过 IO 卡顺次将需要检测的一次性输入指令置位,然后通过遥测接收一次性指令的状态,如果二者一致,则认为一次性指令输入通道正确,通过飞行任务把一次性指令输出通道的相应位置 1 发给产品,然后接收一次性指令,判断接收的一次性指

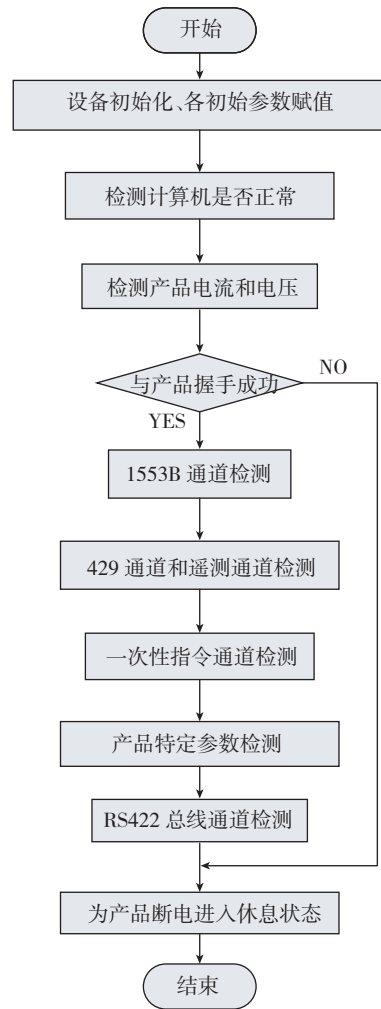


图3 内外接口检测算法流程图  
Fig.3 Flow chart of inner and outer interface testing algorithm

令与飞行任务书中是否一致,每次判断不仅要判断为 1 的指令,也要判断为 0 的指令,这样可以检测产品发送一次性指令和取消一次性指令的功能。有一个出错就把错误计数器加一;

(7)通过高速并行接受通道读取产品特定参数,判断产品的特定数值是否符合要求;

(8)检测 422 通信通道:422 通道的检测方法同 429 通道的检测方法;

(9)测试结束,为产品断电。

#### 3.2 传感器精度检测算法

传感器精度检测主要检测飞控组件中惯性测量组合的精度,观察惯性元件工作是否正常。流程图见图 4。

- (1)设备初始化,各种初始参数打包;
- (2)为产品上电,检测计算机是否正常;
- (3)设备与产品进行首次通讯;
- (4)设备向产品通过 1553B 总线发送自检指令;
- (5)初始化遥测通道,接收遥测信息,从中解码出

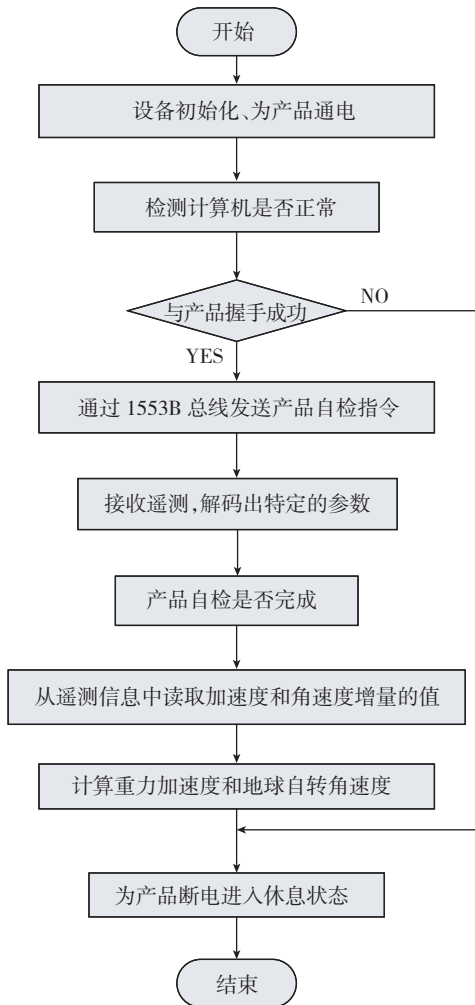


图4 传感器精度检测算法流程图

Fig.4 Flow chart of sensor accuracy testing algorithm

需要的数值;

(6)检测飞控组件自检是否完成;

(7)通过高速并行通道接收产品在 10s 内累加的速度增量值( $svx$ 、 $svy$ 、 $svz$ )和角度增量( $sbx$ 、 $sby$ 、 $sbz$ )值,并通过公式 1 和 2 计算重力加速度和地球自转角速度的值,看其是否在允许的范围之内,从而判断敏感元件的精度:

$$g = \frac{\sqrt{svx^2 + svy^2 + svz^2}}{10.000}, \quad (1)$$

$$d = \frac{\sqrt{sbx^2 + sby^2 + sbz^2}}{10.000}; \quad (2)$$

(8)给产品断电,进入休息状态。

### 3.3 导航精度检测算法

导航精度检测主要包括飞控组件调度时序的检测、初始对准的结果检测以及发射后产品的姿态精度和位置精度检测。流程图见图 5。

(1)设备初始化,各种参数初始化;

(2)为产品上电;

(3)检测计算机是否正常;

(4)设备与产品进行首次通讯;

(5)握手成功后,发送自检指令;

(6)产品自检完成后,发送对准指令和传递对准参数块和任务加载数据块;

(7)检测挂飞段遥测通道是否正常,依次接收各遥测字,有一个接收不到则错误数加 1;

(8)延时一段时间,等待产品对准,接收悬挂物监视,判断产品对准状态,如果产品对准好则发送激活电池指令,产品接收到激活电池指令后发出一次性指令激活电池;

(9)电池电压正常后设备发送投放指令;

(10)产品收到投放指令后则进入自主飞状态;

(11)接收进入自主飞后的遥测数据,从中分离出导弹四元数、速度和位置,并计算出导弹的姿态角并显示到屏幕上,此数据是经对准修正之后的信息;

(12)检测计算机运行信号的周期和占空比;通过定时器设置和读取一次性指令和检测工作周期信号的周期和占空比是否在允许的范围之内;

(13)进行产品的导航计算通过遥测接收产品的导航结果,包括位置和姿态误差,并实时显示;

(14)为产品断电进入休息状态。

## 4 软件设计实现与关键技术

整个测试系统需要实时通讯,而且各测试模块的功能相对独立,同时需要与用户进行较多的信息交互。因此本软件是在 Windows2000 系统,采用 VC++6.0 语言,基于“文档/视图”程序结构开发出来的一个应用软件。VC++6.0 是功能强大的面向对象程序设计语言,适合底层开发,开发灵活,使用 VC 实现上述各功能模块和构建系统使开发过程变得简洁、清晰、自然。用 VC 类机制可以方便地建立各功能模块相应的类。

### 4.1 多线程技术的实现

对于同一个程序,它可以分为若干个独立的执行流,称之为线程;线程提供了多任务处理的能力,可以访问程序的资源,共享程序的虚拟地址空间,线程间处于并行执行状态。本程序需要与硬件长时间打交道,因此可以建立工作线程完成后台操作,如利用多线程实现了设备定时发送飞行任务的功能。

### 4.2 视图分割技术

由于在显示窗口数据时需要将窗口拆分,可以通过树型控件显示数据。调用 CListPane 类对象进行窗口列表拆分,在左窗口中显示树型控件,在右窗口中管理需要显示的多数数据类。通过响应树型控件被选择的消息,

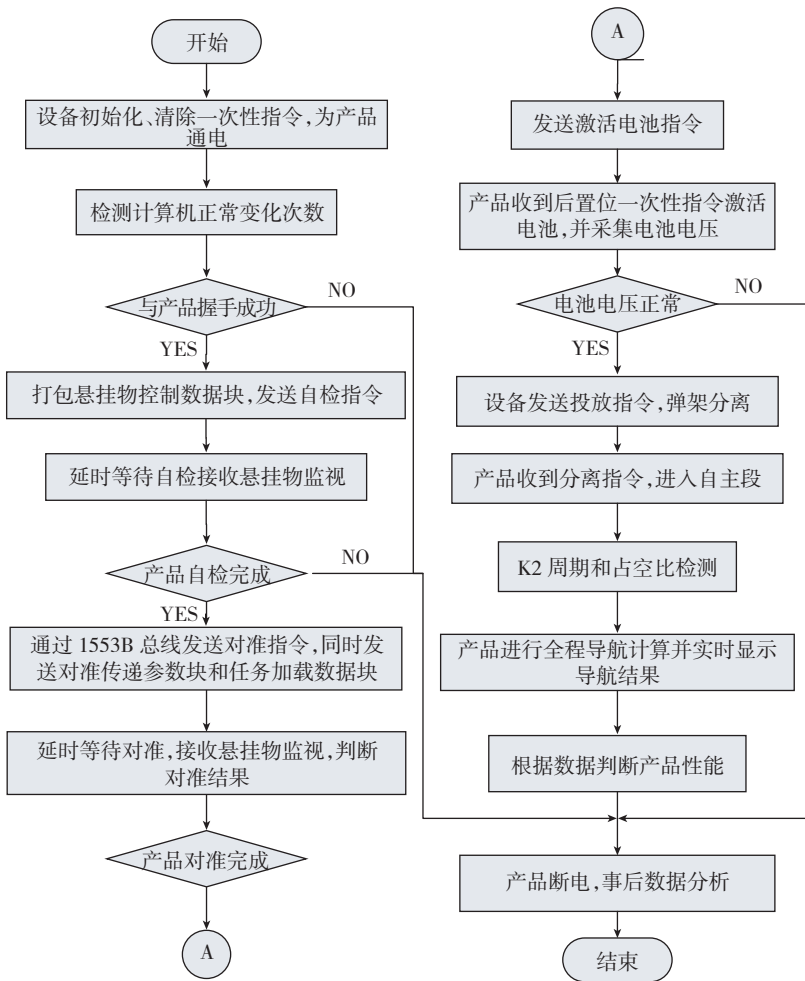


图5 导航精度检测检测算法流程图

Fig.5 Flow chart of navigation accuracy testing algorithm

调用 CMainFrame 类中的 SwitchToView 函数来显示不同的窗口内容。

### 4.3 定时器的应用

为了更直观地看到产品的通电时间等信息,在窗口下面添加一个状态条实时显示时间信息,此时就需用到 windows 的定时器,在 Visual C++ 中采用 SetTimer() 函数创建定时器,向调用 SetTimer() 的窗口发送 WM\_TIMER 消息,同时利用 windows 的消息处理机制来处理该消息<sup>[2]</sup>。Window 的定时器用在不太精确的场合还可以,但在产品的测试过程中常常需要通过时间对程序的流程进行控制,时间要求也比较精确(ms 级别),此时 windows 本身的定时器就显得有些无能为力,因此在硬件上设计了另一个定时器,在需要更高精度的地方就需用到此定时器,此定时器的分辨率为 10μs,可以满足需求。调用 IO\_Start\_Timer(time) 函数来设置定时时间,单位为 10μs。同时可以通过 IO\_Read\_Timer() 函数来获取定时器运行的时间值,从而来控制程序的运行。

### 4.4 程序实现

软件中各种框架、数据的整理、显示主要通过设计各种类来实现。在软件的具体编制中基于 VC++ 设计了多个类函数,具体如下所示。

#### 4.4.1 应用程序类 CHXP2\_App

应用程序类 CHXP2\_App 是基于 CWinApp 的一个类,代表一个独立的应用,通过调用全程对象 App 起作用,CHXP2\_App 定义了类的全部行为<sup>[3]</sup>。

主要在里面定义了每一个测试项的文档模板、总的测试结果对应的文档模板、视图显示模板等,并设置了一些测试控制标志。

```
CMultiDocTemplate *pCKTemplate,*pBKTemplate,*pPKTemplate;
```

```
CMultiDocTemplate *pDocTreeTemplate;
```

```
CMultiDocTemplate *pDisplayTemplate。
```

#### 4.4.2 应用程序框架类 CMainFrame

应用程序框架类 CMainFrame 是由多文档框架窗口类 CMDIFrameWnd 派生而来。工具条、状态条等所有与主框架窗口有关的操作都在这里定义。作为菜单响应函数的产品三项测试也在这里启动。另外还对应用程序运行时的窗口状态进行了保存和恢复。

在 OnCreate 函数一开始调用 LoadWindowPlacement 来装入上一次退出时的窗口状态,它是在 OnClose 函数中调用 SaveWindowPlacement 中保存到注册表的。

还有一个成员函数比较重要,就是 SwitchToView,它的功能是生成一个以 ChildFrame 为基础的、与规定的文档模板相联系的新窗口,并作为当前窗口。这样在每一个测试项菜单响应函数中就只需一条语句 SwitchToView(theApp.pAppTemplate) 就可以了。

#### 4.4.3 测试子窗口框架类 CChildFrame

子窗口框架类 CChildFrame 是由多文档子窗口类 CMDIChildWnd 派生而来。有关子窗口的操作都在这里进行。有子窗口的标题形成、子窗口的初始大小、以及 Start 按钮的显示和状态等等。具体如下:

在重写的 PreCreateWindow 中用 CS 结构定义了子窗口的初始大小。在 OnCreate 中以 DialogBar 的形式创建一个工具条,并显示一个按钮 Start,用来启动每一次测试。通过重写 OnUpdateFrameTitle 在子窗口标题栏上显示“文档名: 测试项”形式的标题。通过响应 UI 命令使 Start 按钮允许和禁止。

#### 4.4.4 文档类 CHXP2Doc

文档类 CPkpa702Doc 是由 MFC 类库的文档类 CDocument 派生而来,它主要完成文档的建立、打开和保存。定义了 ResultArray[7][2]、ResultArrayCK[48][6] 等七个二维数组用来保存各项测试结果的。作为试验必须记录的产品号、设备号、参加人、日期、时间等信息也在这里用菜单响应函数 OnTestDevice() 和 OnTestTime() 来完成。

#### 4.4.5 文档显示子窗口框架类 CTreeListFrm

文档显示子窗口框架类 CTreeListFrm 是由多文档子窗口类 CMDIChildWnd 派生而来。它定义了一个带有分格器的子窗口。

通过重写 OnCreateClient,用分割器窗口变量 m\_wndSplitter 来创建一个一行两列的分格器窗口,两个区域分别与 TreePane 和 ListPane 建立联系。在 PreCreateWindow 通过修改结构 cs 对子窗口的大小进行了定义。

#### 4.4.6 结果显示列表视类 CListPane

列表视类 CListPane 是由 MFC 类库的 CListView 派生而来,它以 ListView (列表)的形式对测试结果进行显示。每一项测试有多个参数,每个参数有名称、代号、允许上限、允许下限、实测值等多个字段,用 ListView 的方式可以把结果直观地显示出来。另外文档的打印和打印预览、打印所需的准备工作都在这里进行。测试完成后的休息定时器设置和显示也在这里进行。

#### 4.4.7 试验条件对话框类 CDevice

CDevice 对话框类由 MFC 的 CDialog 派生而来,它是以对话框的形式,由操作人员输入试验条件。在对话框构造函数中,用定制串行化的方式从一个固定的文件取上次试验时的参数,这样可以避免重复输入,节省时间。在对话框的 OK 响应函数中,再把本次试验的新数据输入到文件。

## 5 结论

本测试软件各模块相互独立,可扩展性强,具有友好的界面,良好的可维护性,方便用户操作,另外对产品的软硬件测试相结合,检测程度深,虚警率低;目前已成功应用于某型号飞控组件产品的测试和各种试验中,为产品的研制节省了大量时间。

### 参考文献

- [1] 《MIL-STD-1553 DESIGNER'S GUIDE》, DDC 公司。
- [2] 洪锡军,陈彩锁,李从心. Windows 下高精度定时的实现. 计算机应用研究, 2000, 17(3): 96-97.
- [3] 武安河,周利莉. Windows 设备驱动程序开发实务. 北京: 电子工业出版社, 2001.

(责编 小城)

(上接第 124 页)

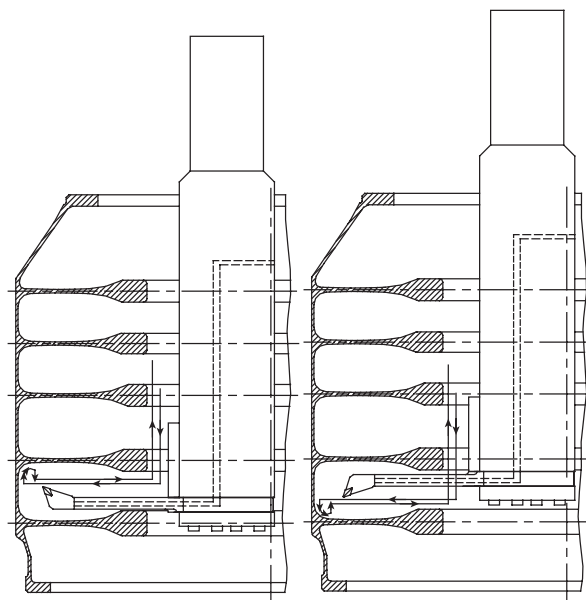


图3 加工上、下部内腔型面走刀轨迹示意图

Fig.3 Diagram of machining trace on upper and lower surface of cavity

图3(右侧)图形中刀具轨迹,调用程序切削深腔下部型面,切削参数:转速 42r/mm;进给速度 0.15mm/r;切深 0.3 ~ 0.6mm,直径方向与加工后的内腔接平,轴向上与零件原有腹板上表面型面接平,加工一级后依次向下加工下一级,完成整个零件内腔型面的加工。

按照改进后的加工方案,进行鼓筒零件内腔型面的加工工作,加工过程中刀具与零件没有出现震动现象,加工后的内腔表面达到了图纸要求的  $1.6\mu\text{m}$  表面质量,避免了原有工艺中加工后的抛光环节,内腔表面的一致性较好,而且内腔直径让刀现象基本消除,测量同一级内腔上下部位的直径尺寸,较改进前的加工后直径相差 0.04mm 左右的情况大为改善,上下部位仅相差不到 0.01mm,加工的精度水平得到了提高,加工的效率较改进前提高了 15% 以上,刀具消耗的成本下降近 40%,改进工作取得了成功。

## 6 结束语

应用改进的刀具方案加工鼓筒零件腹板间内腔型面,改变了传统的加工工艺,满足了全封闭深腔结构鼓筒零件内腔加工的需要,解决了让刀及零件表面振纹现象的技术瓶颈,消除了传统加工方法中需人工抛光的工步,保证了内腔型面表面质量与尺寸要求,提高了鼓筒的加工效率与安全性,节约了加工成本,实现了该项技术在生产中的工程化应用,取得了明显的技术效果和经济效益。

(责编 亿霖)