

基于 LabVIEW 的 ATLAS 编译器设计与实现*

吴 奈 潘文平 王剑楠

(南京航空航天大学民航学院 南京 210016)

摘要: 针对目前 ATLAS 语言编译器国内开发较少,而国外 ATLAS 编译器价格昂贵不易普及的问题,提出一种在 LabVIEW 开发环境下进行 ATLAS 测试语句编译的 ATLAS 编译器,该编译器采用 LabVIEW 面向对象的编程方式对各个部分进行开发,减少 ATLAS 编译器开发的重复性工作,满足编译器的扩展性与可维护性。同时,采用 IVI 规范实现编译器对各个仪器驱动函数的管理,完成了 ATLAS 测试语句到实际 LabVIEW 可执行语句的转化。实践证明,该编译器开发周期短,扩展性强,开发成本较低,对于 ATLAS 语言编译器的进一步开发具有借鉴意义。

关键词: LabVIEW 面向对象技术;IVI 技术;ATLAS 语言

中图分类号: TP311 **文献标识码:** A **国家标准学科分类代码:** 520.4060

Design and implementation of ATLAS compiler based on LabVIEW

Wu Nai Pan Wenping Wang Jiannan

(College of Civil Aviation, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China)

Abstract: Facing on the problem of less domestic ATLAS compiler and the one abroad which is expensive and difficult to popularize, this article proposed a ATLAS compiler which compiled the ATLAS test statement under the LabVIEW development environment. The compiler adopted LabVIEW object-oriented programming approach to the development of various parts, reducing repetitive tasks and satisfying scalability and maintainability of the compiler. Simultaneously, the compiler adopted IVI norms to achieve the management of each instrument driver function, which completed the conversion of ATLAS test statement to the executable statements. Practice has proved that the compiler possessed short development cycle, scalability and lower development costs, which has significant reference for the further development of ATLAS language compiler.

Keywords: LabVIEW object-oriented technology; IVI technology; ATLAS language

1 引言

通用测试语言(abbreviated test language for avionics system, ATLAS)是一个被广泛应用于军事和电子测试的通用标准测试语言^[1]。用这个语言编写的测试程序不依赖于任何特殊的被测系统,并且它能在自动测试系统(automatic test system, ATS)上执行。使用 ATLAS 编写的测试程序只描述测试流程和信号需求,而不包含任何具体设备的信息,因此是一种面向信号^[2]的测试语言,ATLAS 的这一特性使得使用 ATLAS 语言编写的测试程序可以在不同的测试平台中移植,大大提高了测试程序集(test program set, TPS)的通用性。

但与此同时,如何将 ATLAS 编写的测试流程转化成最终可驱动仪器完成目标测试的代码是在实际工程开发中所遇到的必不可少的问题,这里就涉及 ATLAS 测试语句的编译。目前,已有较为成功的 ATLAS 测试语言编译器有 TYX 公司开发的 PAWS 系统和 ARINC 公司开发的 SMART,但这些编译器价格昂贵,不易于普遍推广,国内也有研究人员尝试自主研发 ATLAS 编译器,也取得了一定成果,但离商业化的投入使用仍有一定距离^[3]。

对此,提出一种基于 LabVIEW^[4]开发环境的 ATLAS 编译器,利用 LabVIEW 开发效率高、与仪器配合使用灵活等特点,采用 LabVIEW 中面向对象的编程方式开发 ATLAS 语言编译器,结合 IVI 规范对仪器驱动进行统一

收稿日期:2015-09

* 基金项目:南京航空航天大学青年创新基金(XAA15071)、南京航空航天大学研究生创新基地(实验室)开放基金(kfjj20150709)、南京航空航天大学创新实践工程自由探索项目(ZT2014072)项目

管理,实现了 ATLAS 信号语句到 LabVIEW 代码的编译与执行,对实现 ATLAS 语言编译器的进一步开发提供了一种解决方案。

2 LabVIEW 面向对象技术

编译器的开发是一个大型的软件构建过程,如果采用面向过程的编程方式会带来软件后期修改扩展困难、重复性工作量大、针对 LabVIEW 中多参数的大型复杂软件开发,一般采用面向对象的开发方式。

LabVIEW 面向对象的编程方式^[5-6]将任务分解为一系列“对象”组成的实体,其最基本的概念是“类”和“对象”,“类”是创建“对象”的模板,“对象”是“类”的特定显示,具体来说就是将各种测试任务分解成不同的“类”,在“类”中定义私有数据,编写对私有数据进行操作的成员 VI。在 LabVIEW 中,类的基本结构如图 1 所示。

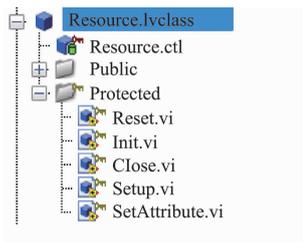


图 1 类的基本结构

Resource 类的数据存放在 Resourc.ctl 中,Resource 类的方法使用 Reset.vi、Init.vi、Close.vi、Setup.vi、SetAttribute.vi 这几个 VI 来表示。这几个 VI 是类的特定实例,指出某一类资源通过这几个 VI 可以进行特定的操作,是 Resource 类的一个对象。

3 编译器面向对象设计方法

测试程序要实际运行,必须要有实际设备的支持,才能完成设备分配、信号输出、数据采集等任务,因此需要对测试环境进行描述。测试环境大致可以用两部分内容概括:设备信息以及路由信息。设备信息包括了设备资源名称、物理通道名称、虚拟资源名称、测试能力描述等内容,路由信息主要是对测试系统中被测件与测试设备之间物理连接关系的描述。除此之外,还需要完成信号语句的测试动作(即单动作语句、多动作语句)到实际 LabVIEW 代码的转化,并且在编译器运行时对虚拟资源状态进行监视及更新。针对这些工作,在编译器中创建了 Factory、Signal、Resource、Sensor、Source、Routing 6 个“类”,每个“类”完成其中一部分工作,对各模块的参数进行管理,降低参数之间的耦合度,将各部分的功能模块化,这样做的好处是便于以后编译器功能扩展时不需要对整体进行修改,只需要对其中一个模块进行修改,降低了工作量^[7],这 6 个“类”之间的关系如图 2 所示。

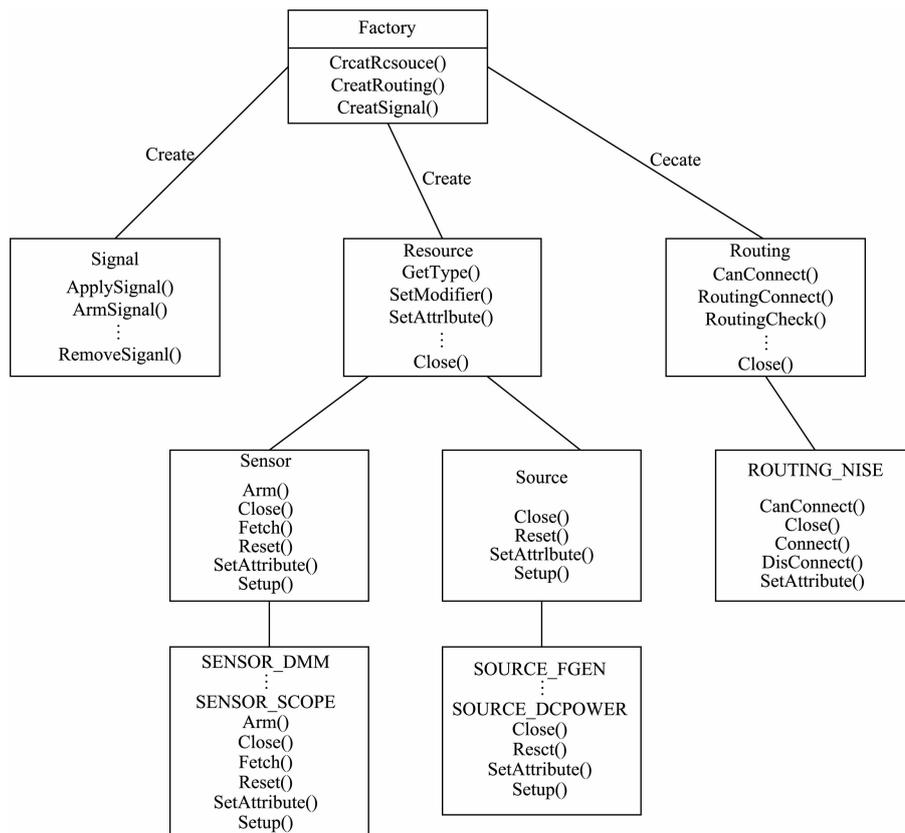


图 2 编译器中各“类”之间的关系

1) Factory 类说明

Factory 类提取出系统的路由信息、虚拟资源的信号信息、系统设备资源信息,并通过这些信息分别完成路由资源的创建、信号资源的创建以及设备资源的创建,这些资源的创建是后续其他几个类对这些资源进行操作的基础,相当于完成测试系统初始化的功能。

2) Signal、Resource、Routing 类说明

Signal、Resource、Routing 3 个类分别对前面 Factory 类创建完成的信号资源、设备资源以及路由资源使用其相应的私有 VI 进行操作,Signal 类中对应的操作 VI 主要是 ATLAS 语言中的各种信号语句的测试动作,例如:APPLY、ARM、FETCH 等;Resource 类中对应的私有 VI 主要是设置和获取虚拟资源信号的类型、修饰语、参数等信息;Routing 类中对应的操作主要是完成路由关系的连接、断开,当前路由状态的获取以及更新等操作,包括了 ATLAS 语句中的 CONNECT、DISCONNECT 语句,也涉及到类似于 RoutingCheck.vi 这样对当前路由状态监测的操作 VI。

3) Sensor、Source 类说明

Sensor、Source 类主要完成将 ATLAS 信号语句的测试动作到具体仪器驱动函数的转化,具体测试动作包括 SETUP、ARM、FETCH、RESET 等,这两大类主要涉及到了对仪器 IVI 驱动的使用,由于 LabVIEW 中有 IVI 驱动函数可以直接调用,因此这两大类中私有 VI 可以直接使用这些 IVI 驱动函数进行编写。

4 IVI 技术

为了实现互换性,IVI^[8]基金会将同类仪器的共性提取出,并作了规范。在 LabVIEW 中每一类的仪器都有各自的类驱动程序(IVI Class Driver),类驱动程序包含了该类仪器通用的各种属性和操作函数。运行时,类驱动程序通过调用每台仪器的专用驱动程序(IVI Specific Driver)中相应的函数来控制仪器。使用 IVI 规范管理底层驱动函数,总结下来有下面 3 点好处^[9-10]:

1)易于使用。所用的 IVI 驱动程序都使用通用的接口,易于理解,也就不再要求应用程序的开发人员必须掌握某一特定仪器的编程方法,从而系统开发获得了更大的硬件独立性。

2)降低了系统的维护和升级费用。IVI 构架系统可以适用不同的仪器,当仪器陈旧或者有了升级的、高性能或低造价的仪器时,可以任意更换,而不需要改变应用程序。

3)代码共享。IVI 构架允许部门和设备之间方便地复用及共享测试代码,并且不需使用相同型号仪器硬件。

5 ATLAS 信号语句分析

5.1 单动作语句与多动作语句的关系

ATLAS 信号语句中单动作语句有 SETUP 语句、CONNECT 语句、DISCONNECT 语句、ARM 语句、FETCH 语句、CHANGE 语句和 RESET 语句,这些语句在编写测试程序需要组合使用且具有固定的出现顺序,针对不同类型的仪器(Sensor 或 Source),其组合顺序也各不相同。例如,对于 Source 类型仪器而言,SETUP 语句应跟在 CONNECT 语句之后。多动作语句在本质上相当于一系列单动作语句的组合,且这些组合是固定的,因此在使用多动作语句编写测试程序时就不用需要像使用单动作语句那样考虑语句之间的顺序问题,故一般在编写测试程序时推荐使用多动作语句。以 APPLY、REMOVE、MEASURE 这几个多动作语句为例,表 1 说明了多动作语句与单动作语句之间的关系。

表 1 多动作语句与单动作语句关系

多动作语句	Source 类仪器	Sensor 类仪器
APPLY	CONNECT	CONNECT
	SETUP	SETUP
	RESET	DISCONNECT
REMOVE	DISCONNECT	RESET
		SETUP
		CONNECT
MEASURE	无	ARM
		FETCH
		DISCONNECT
		RESET

5.2 信号语句状态转移分析

ATLAS 语言中单动作语句和多动作语句的执行会引起测试系统中虚拟资源状态的改变,而这种改变的顺序是有一定顺序的,因为 ATLAS 信号语句的执行顺序具有一定的顺序,这种顺序在 ATLAS 中称为虚拟资源的状态转移^[11]。信号语句可以描述源、传感器等类型的仪器,对于不同类型仪器的信号语句,其对虚拟资源的状态转移也是不同的。下面通过对某一 SENSOR 类型虚拟资源的状态转移加以说明,如图 3 所示。

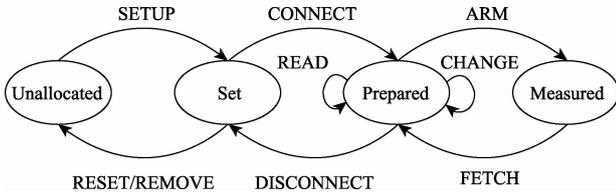


图3 SENSOR状态转移

从图3中可以看出,虚拟资源的状态转移是由每句单动作语句的执行来触发的,当某一句单动作语句执行完毕之后,该虚拟资源的状态应该由前一状态转变成当前状态,而在测试系统中,始终对该状态进行监视,并及时更新虚拟资源的当前状态,即状态 State→State'。与虚拟资源状态同时改变的还有系统的路由连接状态,这里涉及到系统的物理连接关系的转变。编译器在对 ATLAS 语句进行上下文语义分析的时候,对于前后两句测试语句中的测试动作之间是否符合虚拟资源状态转移规则进行检查,若不符合状态转移规则,编译器将提示错误信息。

需要注意的是,并非所有的动作语句都会改变虚拟资源的状态,如 CHANGE 语句,只改变已分配设备的一些特性,但不会改变设备状态。

同样需要注意的是,对于某些测试动作,源或者传感器类型的信号语句都具备,如 SETUP,但不同类型的虚拟资源,即使相同的测试动作,编译部分最终将其转化为可执行代码语句时必然是不相同的。对于 SENSOR 类型仪器的 SETUP 动作,应当是进行采集部分函数的参数配置,而对于 SOURCE 类型仪器的 SETUP 动作,则应当是进行输出部分函数的参数配置,由于本编译部分最终的目标代码是 LabVIEW 的形式,如图4所示,以 DAQ 函数为例,对于 SENSOR 类型的 SETUP 动作,其代码如图4所示。

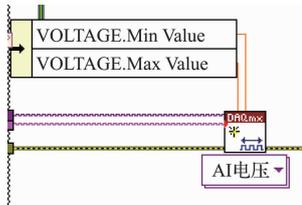


图4 SETUP的LabVIEW代码

5.3 单动作语句到 LabVIEW 代码的转化

由于多动作语句是一系列单动作语句的集合,因此只需要完成单动作语句到 LabVIEW 代码的转化,也就相当于完成了多动作语句到 LabVIEW 代码的转化。

其具体的转化过程在 Signal 类中完成,以单动作语句作为最底层的执行语句,在编译的时候需要直接编译成为可供 LabVIEW 执行的代码,而多动作语句的编译则是先转化成多步单动作语句的集合,然后再将这

些单动作语句集合编译成 LabVIEW 代码,最终执行^[12]。ATLAS 7 条单动作信号语句在 Signal 类中对应 SetupSignal. VI, ResetSignal. VI, ArmSignal. VI, FetchSignal. VI, ConnectSignal. VI, DisconnectSignal. VI, ChangeSignal. VI 7 个 VI,以 SetupSignal. VI 为例,编译器的分析过程如图5所示。

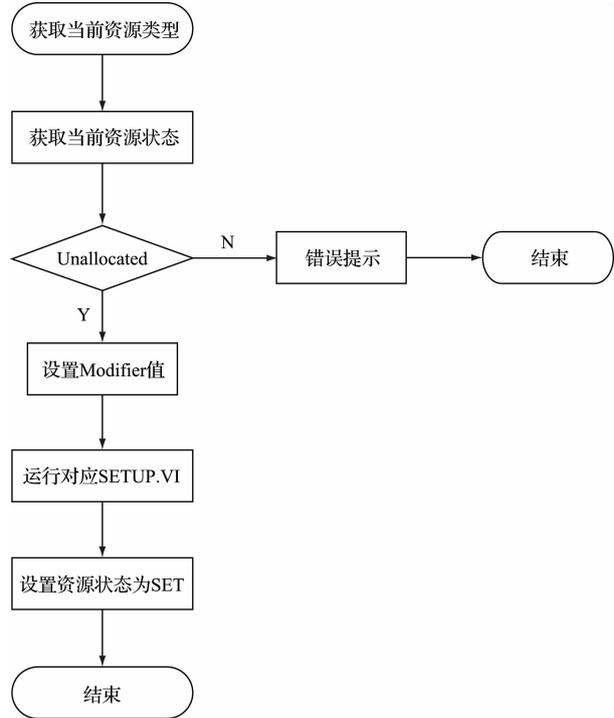


图5 SetupSignal 编译器分析过程

1) 获取当前所操作虚拟资源的设备状态,若目前状态不是 Unallocated,则编译部分会提示错误,若为 Unallocated,则进行下一步;

2) 获取该虚拟资源对应信号的 ModifierName、Max-Value、MinValue;

3) 进行 SETUP 动作,首先获取信号类型(Sensor 或 Source),然后将 Setup 动作根据前面所获取的设备信息,转换成 LabVIEW 代码。

6 实例分析

以某一型号的电板测试为例,其需求的测试流程如下:

- 1) 向被测电路板供电;
- 2) 延时 1 s;
- 3) 通信功能检测(串口 RS-232);
- 4) 被测板模拟输入功能检测;
- 5) 被测板模拟输出功能测试;
- 6) 被测板频率生成功能检测;
- 7) 被测板频率测试功能检测;
- 8) 断电;

9)合格或失败显示。

在该例中,测试流程较为简单,但涉及模拟信号采集、模拟信号发出、频率功能测试以及串口通信,因此具有一定的代表性。

6.1 硬件配置分析

系统硬件平台采用 NI(National Instrument)公司的多功能采集卡 PCI 6220 作为采集设备,并采用 Pickering 公司的矩阵开关卡来进行路由切换,使用工控机上的自带 RS-232 串口进行串口通信。基本硬件连接关系如图 6 所示。

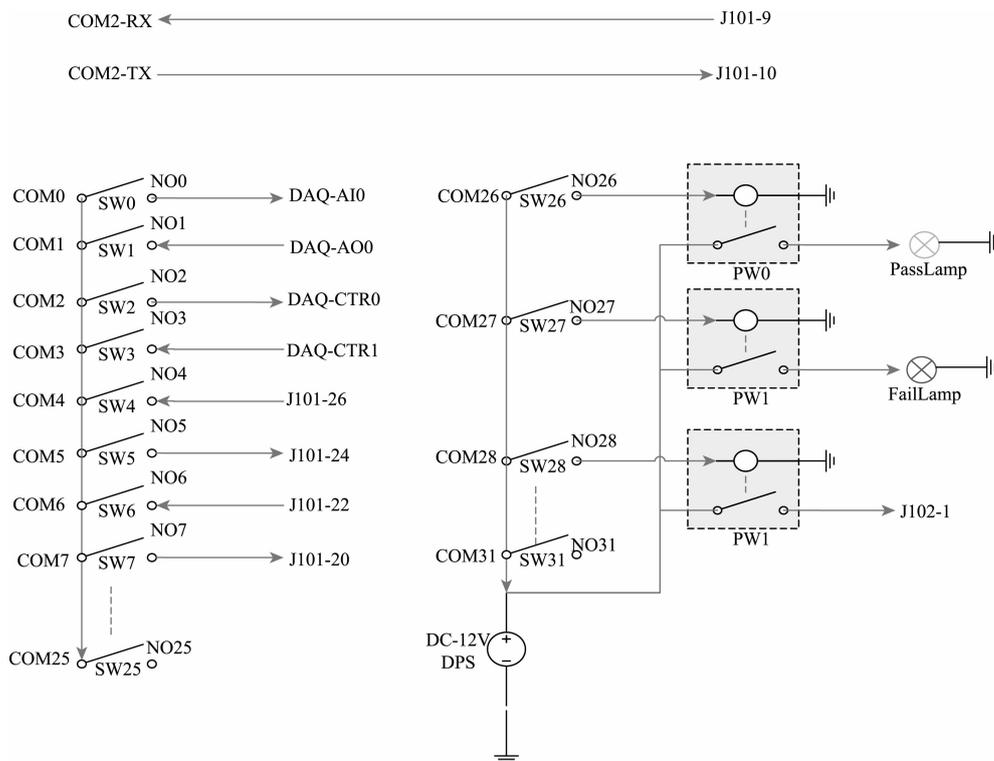


图 6 基本硬件连接

在矩阵开关卡配置文件中,将引脚与对应信号连接完成,然后进行后续测试流程的编辑。

6.2 测试流程编辑分析

对应上述测试流程的需求,编写了下面 ATLAS 标准的测试流程,以部分代码为例。

1)检测开始: MONITOR, (VALUE), LOGIC DATA USING 'DIGITAL METER', LOGIC-ONE 4V, LOGIC-ZERO 0.4V, EQ'B'1", CNX J101-11/12 \$

2)检测夹具: MEASURE, (VALUE), LOGIC DATA USING 'DIGITAL METER', LOGIC-ONE 4V, LOGIC-ZERO 0.4V, CNX J101-9/10 \$

3)打开电源: APPLY, DC SIGNAL USING 'SOURCE _DPS', VOLTAGE 12.000V, CNX J102-1 \$

4)输出 DC(2): MEASURE, (VALUE-ASCII), RS SERIALS COMMUNICATION USING 'SERIALS COMM ADA-

PTAR', BIT-RATE 9600 BITS/SEC, VALUE-ASCII 'write 2.34', RETURN DATATYPE 'VALUE-ASCII', STOP-BIT-LENGTH 1.0, PARITY-MODE NONE, CNX J101-20/21 \$

5)读取 DC(2): MEASURE, (VOLTAGE), DC SIGNAL USING 'VOLTAGE METER', CNX J101-26 \$

6)关闭电源: REMOVE, DC SIGNAL USING 'SOURCE_DPS', VOLTAGE 12.000 V, CNX J102-1 \$

对于编译部分而言,要做的工作就是将前面编写好的 ATLAS 测试流程语句,结合测试系统中的仪器资源以及测试信号类型、参数等信息,编译成为 LabVIEW 可识别的代码语句,进而进行实际的测试,以“读取 DC(2)”为例,完成了使用 NI PCI-6220 采集卡读取被测电路板上某测试点电压值的功能,其经过编译形成的 LabVIEW 执行代码如图 7 所示。

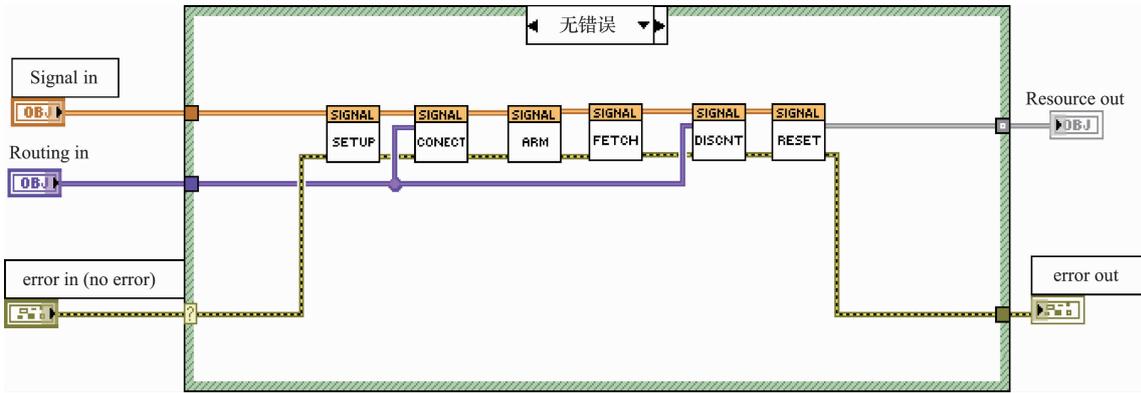


图7 编译完成代码

由于 MEASURE 动作在 ATLAS 中对于 Sensor 类信号语句来说是 SETUP 等一系列单动作语句的集合,而这些单动作语句在编译部分是单独进行编写的,因此

MEASURE 测试语句在编译部分就是这些对应单动作语句类下私有 VI 的分步执行。就其完成的实际测试功能代码如图 8 所示。

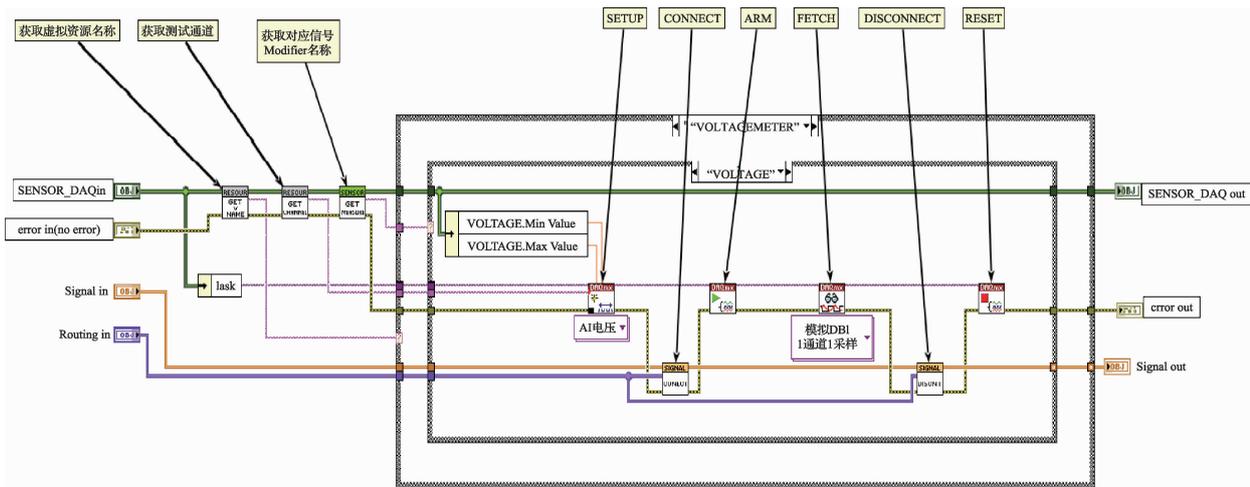


图8 完成的实际测试功能代码

从上述代码中可以看出,单动作语句实际上将 LabVIEW 中的采集函数进行了模块化的封装,每一句单动作语句对应了一个采集函数,在实际运行时,还需要对当前测试状态进行查询,在测试完成之后则更新当前的测试状态,由于篇幅所限,上述代码只写出了测试功能的部分,对其他一些系统测试状态部分的编写并未详细列出。

7 结论

编译器将 ATLAS 测试语句转化成 LabVIEW 的目标代码形式,国内还有其他研究人员开发出了其他编译器,将 ATLAS 测试语句转化成 C++ 代码的形式,也可以进行测试流程的最终执行,但相较于 LabVIEW 而言,C++ 语言的开发环境一般为 VC++ ,需要重新调用测试板卡的驱动函数,与一般数据采集卡以及支持 IVI 驱动的可编程仪器之间的结合不如 LabVIEW 那么灵活。

除此之外,从 ATLAS 语句转化过来的 C++ 代码语句还需要其开发环境进行编译之后才能运行,而 LabVIEW 代码编写完成之后可以直接运行,不需要再次进行

编译,因此,LabVIEW 在代码执行效率要高于 C++。

以 LabVIEW 作为编译器的开发环境,采用面向对象的开发方式,对于整个编译器的开发效率有了较大的提升,同时编译器中各个类的扩展也十分方便,对于以后编译器的升级提供了较大的支持。目前,编译器完成了 ATLAS 单动作测试语句以及多动作测试语句到 LabVIEW 目标代码的转化,经转化而来的目标代码行也可以完成实际的测试任务,但对于一些复杂的测试语句,如循环语句、时序语句等还没有完成这部分的编译工作,因为这部分的工作涉及到了对于目标代码整体运行框架的设计,而不是单动作语句流程化执行的简单设计,怎样去设计循环、时序这些 ATLAS 语句的转化也是后面有待完成的工作。

参考文献

- [1] 杨召,肖明清,胡斌,等. 国外航空自动测试描述语言发展综述[J]. 计算机测量与控制, 2013, 21(4): 833-835.
- [2] 王怡苹,李文海,文天柱. 面向信号测试的路径搜索算

- 法研究[J]. 仪器仪表学报, 2013, 34(7): 1650-1658.
- [3] 王胜军. ATLAS 语言编译系统[D]. 吉林: 吉林大学, 2006.
- [4] 王海霞, 颜桂定, 李宝辉, 等. 直线电机运动控制系统的软件设计与实现[J]. 电子测量与仪器学报, 2013, 27(3): 264-269.
- [5] 陈海生, 邓锐. LabVIEW 面向对象编程技术[J]. 现代计算机: 专业版, 2008(4): 70-72.
- [6] 李曼, 赵坤. LabVIEW 程序设计方法在自动测试系统中的应用讨论[J]. 测控技术, 2014, 33(12): 107-109.
- [7] HE Y F, WANG S J, PENG Y. High performance heterogeneous embedded computing: a review[J]. Instrumentation, 2014, 1(2): 1-9.
- [8] 徐卫, 贺正军. 基于 Nimbus 的 IVI-COM 仪器驱动设计[J]. 电子测量技术, 2014, 37(2): 11-17.
- [9] 阮奇楨. 可互换虚拟仪器驱动程序的开发[J]. 测控技术, 2006, 25(5): 78-80.
- [10] MUELLER J. IVI 驱动程序标准[J]. 国外电子测量技术, 2013, 32(6): 11-17.
- [11] 黄双双. 基于 ATLAS 2000 的通用 ATS 的可视化开发软件的设计[D]. 西安: 西安电子科技大学, 2013.
- [12] 刘浩, 朱小平. ATLAS 语言在自动测试设备 ATE 中的应用实践[J]. 计算机测量与控制, 2005, 13(2): 118-119.

作者简介

吴奈, 1990 年出生, 在读研究生。主要研究方向为基于总线的计算机测控技术。
E-mail: wunai1002@aliyun.com

全新源测量单元满足高功率器件测量应用需求

泰克推出 2461 SourceMeter[®] 图形源表仪器, 观察和分析高达 10 A、1 000 W 的待测器件特点

2016 年 1 月 14 日, 泰克科技公司日前宣布, 推出一款简便易用的图形源测量单元 (SMU) 仪器, 用以优化和分析高功率材料、器件和模块的特性。

Keithley 2461 高电流源表 SMU 仪器为创建精确控制的 10 A/100 V、1 000 W 高电流脉冲提供了许多先进的功能, 最大限度地降低功率器件热量效应, 保持器件的完整性。双 18 位高速模数转换器可以简便地测量待测器件特点, 并在前面板上直接以图形方式显示, 立即进行分析。在久负盛名的 2450 和 2460 SMU 平台基础上, 2461 拥有同类最高的 DC 和脉冲源和阱性能, 用户可以更深入地了解设计的特点。

开发下一代高功率材料和器件的科研人员、科学家和设计师必须能够在各种 DC 和脉冲功率上进行测量, 检验待测器件性能, 同时使器件自热效应达到最小, 因为自热经常导致器件或模块故障。这适用于材料研究、半导体器件、电路保护装置、高级照明技术、能源贮存和生成器件及消费电子中使用的功率管理电子等市场。

“全球绿色能源计划和能源效率标准正在推动对更高效的功率半导体器件和系统的需求。”泰克科技公司吉时利产品线总经理 Mike Flaherty 说, “最新的高功率应用是非常苛刻的, 要求测试仪器能够分析比以前明显更高的电流、更高的功率、更高的峰值电流、以及更低的泄漏电流。2461 满足了这一需求, 提供了前所未有的同类领先的性能和界面友好组合。”

与泰克其他吉时利图形 SMU 一样, 2461 提供了简单直

观的 Touch, Test, Invent[®] 用户体验, 最大限度地缩短学习周期, 加快测试设置, 更快地获得所需信息。通过图形触屏界面, 用户可以像在智能手机和平板电脑上一样, 使用自然手势在前面板上直接与结果交互, 迅速放大和缩小数据, 同时进行详细的分析。内置开放源脚本语言使得用户能够为专门的测量应用创建可以重用、可以量身定制的测试软件库。

2461 的 10A/100V、1 000 W 脉冲功率使得工程师可以在更短的时间内对待测器件应用更多的功率, 最大限度地降低待测器件的自热效应, 相比之下, DC 电流测试则可能会掩盖待测器件的真实特点。如果应用电流的时间太长, DC 测试还可能会损坏待测器件。由于 18 位双 1 MS/s 模数转换器, 2461 可以同时测量和查看待测器件特点、波形以及电流和电压的瞬态事件。全新的快速“接触检查”功能可以帮助用户最大限度地降低测量误差, 减少与接触疲劳、探头尖端杂质、连接松动或断开和继电器故障有关的产品误判。这些功能让用户对测试结果更自信, 从而可以更快地做出设计和工程决策。

上市时间

2461 高电流源表 SMU 仪器现已可以在全球范围内订购, 如需更多信息, 请访问: <http://www.tek.com/keithley-source-measure-units/smu-2450-60-graphical-sourcemeter>。