文章编号:1671-4598(2018)11-0158-04 DOI:10.16526/j.cnki.11—4762/tp.2018.11.035 中图分类号:TP311.1 文献标识码:A

基于 LabVIEW 的通用且可定制的数据采集处理软件设计

纳杰斯，丁明惠
（昆明船舶设备研究试验中心，昆明 650051）

摘要：数据采集系统可将外界模拟信号转换成数字信号并输入计算机，随后在计算机上进行后续数字信号处理[1]；在实际应用中，待采集的模拟信号一般来自各类物理场传感器，针对不同的应用场景、不同的传感器、不同的项目需求，一般需要定制化开发数据采集系统；得益于 NI（National Instruments，美国国家仪器）公司开发的通用化硬件采集设备，用户在数据采集项目开发时只需着重考虑软件部分的设计工作，文章提出一种基于 LabVIEW 编程语言的通用且可定制的数据采集处理软件设计方法，即考虑可用于多个数据采集项目的通用性，又满足不同项目的定制化需求，注重软件复用性，提高项目开发效率。

关键词：数据采集；LabVIEW；软件设计

A Customizable Data Acquisition and Processing Software Design Method Based on LabVIEW

Na Jiesi, Ding Minghui
(Kunming Ship Equipment Research Test Center, Kunming 650051, China)

Abstract: The data acquisition system can convert analog signal to digital signal and input it into the computer. In practical applications, in view of different application scenarios, different sensors, and different project requirements, a customized data acquisition system is generally required. Thanks to the universal hardware acquisition device developed by NI (National Instruments), the user only needs to consider the design of the software part of the data acquisition project. This paper proposes a data acquisition and processing software design method based on the LabVIEW programming language. The method considers the versatility, focuses on software reusability, and improves project development efficiency.

Keywords: data acquisition; LabVIEW; software design

0 引言

数据采集系统包括硬件采集设备及配套软件，硬件采集设备用于模拟信号到数字信号的转换，配套软件一般称为“数据采集与处理软件”，负责硬件设备的配置、控制及对转换后数字信号的处理及终端显示等。硬件采集设备的实现一般使用 FPGA（field programmable gate array，现场可编程逻辑门阵列）＋ADC（analog digital convert，模－数转换器）的硬件平台自主开发，或直接选用市场通用产品。在市场产品中，美国 NI（National Instruments）公司开发的 DAQ 系列硬件采集设备采用模块化设计，用户可针对不同项目自由选择组合功能、性能各异的硬件模块实现一套功能完整的硬件采集设备，使得在开发数据采集系统时，节约硬件开发时间，用户只需着重考虑软件部分的设计工作。

LabVIEW（Laboratory Virtual Instrument Engineering Work bench）是 NI 公司针对于该公司硬件采集产品配套开发的一种程序设计语言[2]，LabVIEW 编程语言可与 NI 公司的硬件采集设备无缝连接，实现采集硬件的实时配置、实时控制[3]，同时 LabVIEW 具备高度封装底层逻辑、图形化编程方式、以数据流为导向、可实时插入过程监视探针等特点，使得 LabVIEW 在测量与控制系统的开发中有着得天独厚的优势[4]。

本文以实际项目为例，项目中硬件采集设备均使用市场上成熟的 NI 公司 DAQ 系列产品，项目开发的主要工作是软件部分的开发，本文具体阐述一种以 LabVIEW 编程语言设计实现的数据采集处理软件的设计方法，该软件的新颖之处在于既能满足多个项目通用的数据采集功能，又能针对不同项目提供定制化的数据处理手段及针对性的终端显示界面。

1 总体设计

对于不同的数据采集项目，有着形态各异的前端传感器，一般情况下，针对不同项目的需求，需要定制化设计数据采集系统[5]。如下图所示，图 1 中标示 3 个独立的项目的开发内容，在软件开发部分，存在大量的重复工作，对于任意的一个数据采集软件，均需要开发用户界面、采集硬件控制、数据存储、数据处理等功能，其中数据处理功

收稿日期:2018-05-09； **修回日期**:2018-05-31。
基金项目:“十三五”国家科技重大专项－大型油气田及煤层气开发重大专项(2016ZX05028005－005)。
作者简介:纳杰斯(1988-)，男，云南玉溪人，硕士研究生，工程师，主要从事软件工程方向的研究。

能需要针对项目定制开发[6]，例如温度采集、声信号采集、磁信号采集，软件界面均需要显示实时采集波形、频谱等基础数据，而数据处理功能模块就不能共用，温、声、磁有各自截然不同的数据处理方法。

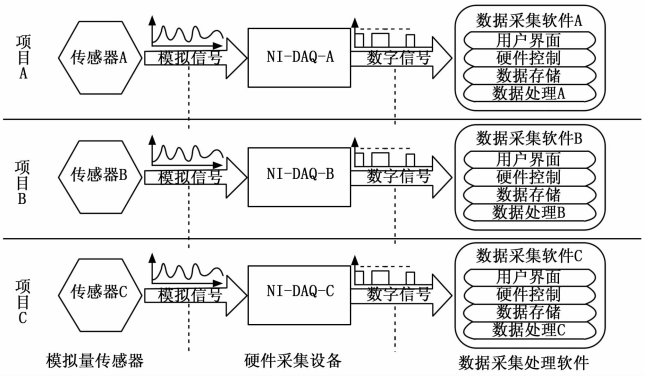

图 1　不同项目的数据采集系统

为了提高软件开发效率，整合各数据采集软件可共用的通用功能（如图 1 中的用户界面、硬件控制、数据存储），设计一个适用于大部分前端传感器的“通用采集平台”，负责通用的数据采集功能。对于不能通用的定制化代码（例如图 1 中针对“项目 C”定制开发的“数据处理 C”模块），采用一种灵活的“插件”编程风格，将定制化代码设计成一种必须依托于“通用采集平台”才能运行的“插件”，“插件”可以按需求动态加载到通用采集平台中，这样即实现了通用的数据采集功能，又满足了针对特定项目的定制化开发，这种设计方法如图 2 所示。

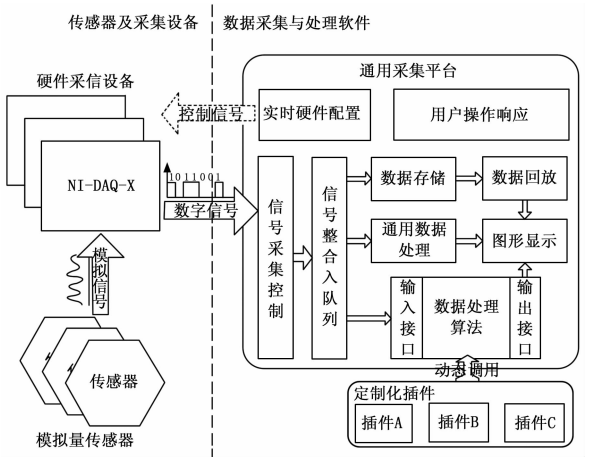

图 2　通用数据采集平台＋插件的设计方式

图 2 中，传感器为任意的模拟量传感器，NI－DAQ－X 为 NI 公司的各型 DAQ 硬件采集设备。数据采集与处理软件分为“通用采集平台”及“定制化插件”两部分，其中“通用采集平台”一次开发成型后，对于不同的项目，只需进行极少的改动、甚至无需改动就可以使用，负责通用的数据采集功能；“定制化插件”需要根据项目定制化开发，满足特定项目的软件功能需求。例如“定制化插件”内的“插件 C”就是针对“项目 C”定制化开发的。简单来说“通用采集平台”实现了“数据采集与处理软件”的“采集”功能，而“定制化插件”则负责实现“处理”功能。

“通用采集平台”实现了适用于大部分模拟信号采集项目的基本数据采集功能，包括采集硬件配置、用户操作响应、信号采集、数据存储、数据回放、图形显示界面等功能，具体实现上，本项目采用基于消息队列的生产者消费者编程框架[7]，该框架的多线程编程方式使得数据采集、数据存储、图形显示等模块化功能处于不同的线程并行运行，保证数据采集的实时性要求，限于篇幅，“通用采集平台”的实现方法不作为本文重点，不再赘述。

2　软件定制化设计

2.1　总体方案

“通用采集平台”实现了各项目的通用采集需求，对于不同项目的特殊需求，开发针对性的定制化“插件”，同时在“通用采集平台”上搭建一个通用接口，可调用所有的定制化“插件”，而且这种调用必须要能“动态”实现，“动态”的意义在于针对不同项目调用加载不同“插件”，其他无关“插件”并不加载入内存[8]，更不会调用，避免随着“插件”数量的增加软件性能降低及内存溢出。如图 3 所示为接口设计的原理框图，包括项目信息检索、UI 菜单初始化、用户操作响应、“插件”动态调用、消息队列初始化等几部分逻辑代码功能块。

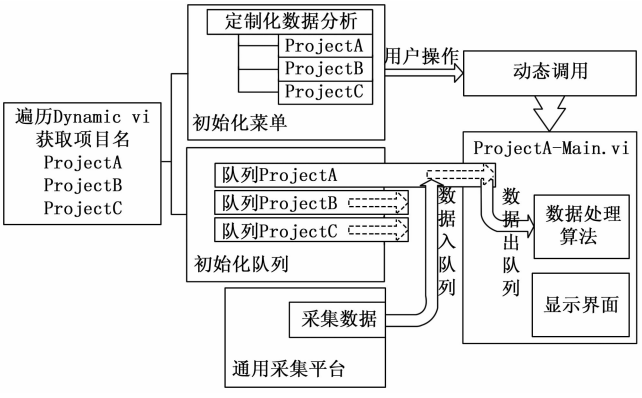

图 3　接口实现原理框图

总体上，“通用采集平台”负责数据的采集，采集后的数据通过队列的方式，传输到与项目对应的“插件”中，具体数据传输到哪个项目、哪个“插件”，由用户操作菜单选项指定，“插件”的工作方式由具体代码决定，既可以是只在后台运行的某种数据处理方法，也可以是在前台运行可与用户交互的窗口。

2.2　项目信息检索

软件启动时，首先检索项目信息，包括项目名与对应“插件”资源位置等。具体实现方法上，将各项目的定制化代码存放在特定文件夹（本项目使用“Dynamic vi”文件夹），文件夹下以项目名（例如“ProjectA”、“ProjectB”……）创建子文件夹存储项目相关定制化代码 vi 文件（LabVIEW 的源代码文件的后缀名为 .vi，指代源代码文

件），项目名文件夹下的所有 vi 文件的集合既是对应该项目的“插件”。“插件”的入口 vi 文件以“项目名－Main. vi”命名，文档结构如图 4 所示。

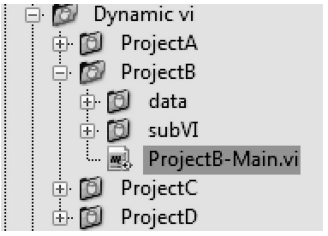

图 4 项目文件结构图

程序初始化时，遍历整个“Dynamic vi”文件夹，得到所有项目名及“插件”资源位置，遍历算法的具体实现如图 5 所示。值得一提的是，图 5 中为了查找“Dynamic vi”的路径，基路径使用的是函数——“本 vi 路径”提供的相对路径，而没有使用绝对路径，这样的好处在于代码移植到其他计算机或文件夹下无需更改[9]。另外，在得到遍历的项目名之后，对应每一个项目名均创建了一个“通知器”，“通知器”可以简单的理解为容量只有 1 的队列，且数据为损耗式入队（Lossy Enqueue），队列满时，不管队列已有数据是否被读取，强制将已有数据清除并将新数据入队，这种数据入队的方式好处是不会发生数据阻塞，保证线程之间的独立性，本文中“通知器”与“队列”意义相同，该“通知器”的使用在下文详述。

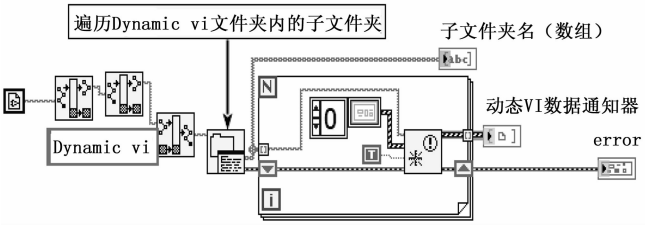

图 5 遍历算法及队列初始化

2.3 UI 菜单初始化

软件启动时，默认运行在通用采集状态下，需要针对项目定制化分析时，需要用户指定用于哪个项目，也就是告知程序调用哪个“插件”。本项目通过 UI（User Interface，用户界面）菜单选项的方式提供给用户操作接口[10]，菜单的初始化是在程序启动时动态完成的，菜单选项对应项目名，项目名的来源由上文所述的遍历检索 Dynamic vi 文件夹得到。如图 6 所示为软件运行时用户点击菜单选择项目的效果截图，当用户点击“定制化分析”菜单选项下面的“ProjectA”时，后台程序就会调用项目对应的“插件A”，准确来说是调用入口 vi 文件“ProjectA－Main. vi”，之后弹出该 vi 的前面板，也就实现了“ProjectA”的定制化分析及显示功能。

2.4 动态调用

软件主 UI 使用菜单选项提供给了用户操作接口，真正的设计重点在于后台接收到用户点击菜单选项后如何实现

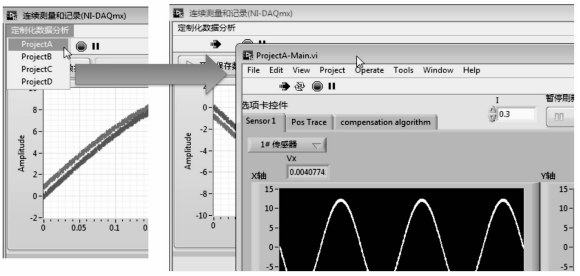

图 6 点击 UI 界面菜单选项效果截图

“插件”的动态调用，具体实现方法上，使用“vi 服务器”的动态加载技术。“vi 服务器”可实现前面板对象、vi 和 LabVIEW 环境的动态控制，vi 服务器的本质是一套 LabVIEW 运行环境的底层管理函数，如图 7 所示为动态调用 vi 的关键代码。

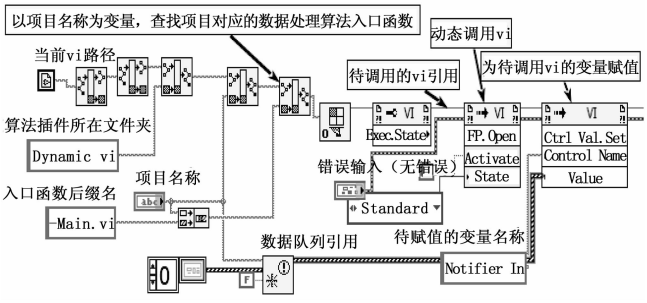

图 7 动态调用代码实现

首先以用户操作传入的“项目名”为变量查找项目对应的插件资源位置，之后使用系统自带的“动态调用函数”调用该“插件”的入口 vi 文件。调用入口 vi 文件时，需要明确指定该 vi 的输入变量“Notifier IN”，该变量是一个“通知器”的引用变量，表明了该 vi 从哪一个“通知器”读取数据。程序初始化时会为每一个项目一一对应创建与之相对应的“通知器”，项目名即为“通知器”名，此处调用时，使用用户操作传入的“项目名”获取对应“通知器”的引用，再将该引用赋值给“Notifier IN”变量，也就指明了被动态调用的 vi 从哪一个“通知器”读取数据。

“Notifier IN”变量的使用，根本上解决了“通用采集平台”与“插件”之间数据交互的统一接口的问题，在“插件”代码编写时，预留“Notifier IN”输入变量，此时该“插件”只知道从该变量对应的“通知器”中去读取数据，而不知道具体是哪个，用户操作选择特定项目时，“Notifier IN”变量被赋值为与项目对应的“通知器”引用，“插件”这时才确切知道从哪里读数据。在“通用采集平台”中，默认工作模式下，“通知器”并不工作，用户操作选择特定项目时，“通用采集平台”才启用项目对应的“通知器”并向其中写数据，损耗性入队（Lossy Enqueue）的写数据方式保证了“通用采集平台”与“插件”的独立性，就算“插件”异常也不影响“通用采集平台”的正常工作，否则可能发生消息堵塞。可见，“插件”与“通用采集平台”之间是相互独立的，“通用采集平台”可独立运行在默认的通用采集模式下，而

“插件”必须依托于“通用采集平台”才能正常工作，他们之间在用户操作时才建立起联系。

2.5　硬件动态配置

另一个需要考虑的问题是对不同项目使用硬件采集设备的实时动态配置。由于 NI 公司提供了功能强大的硬件驱动，在 LabVIEW 软件端实现硬件的控制较为简便，驱动提供了 LabVIEW 自动识别系统中已连接设备的功能，还提供了详尽的硬件操作接口，例如“打开硬件资源”、“读取数据”、“重启设备”等。具体软件设计上，设计独立的模态窗口用于用户操作配置硬件，采集开始前用户必须自行配置硬件参数，例如“采样率”、“采样电压范围”、“存盘路径”等必要参数，配置信息具有自动存储功能，下次启动软件时默认参数为上一次配置参数，如图 8 为硬件配置窗口，包括数据输入配置选项及数据输出配置选项，当系统没有连接实际硬件设备时，可配置为虚拟硬件，用于软硬件调试。

图 8　硬件配置窗口

3　效果评估

本文所述的软件设计方法，主要是为了解决代码的复用性问题，提高项目开发效率。对于文本编程序言，可以通过评估复用代码的行数在总代码行数中的占比来衡量代码的复用性。

对于 LabVIEW 这种图形化编程语言，很难严格的量化代码量，一种方法是将 vi 数量理解为代码量，不过，每个 vi 的功能不尽相同，规模也各不相同，单个用户 vi 的规模并没有定性规定，完全随工程师的个人编程习惯，就算对于同一个工程师编写的程序，这种方法也只是一种大概的估算；另外还可用 vi 的 nodes 来评估代码规模，nodes 可以简单理解为 LabVIEW 代码中的输入输出节点的总数量，节点的概念可简单理解为输入输出接口，例如简单的加法运算包括 2 个输入节点，1 个输出节点，LabVIEW 中自带的底层 vi 的节点数是固定的，而大部分用户 vi 可分解为各种底层 vi，这种方法相对更客观。

如图 9 所示，从左到右分别是项目主 vi（Main. vi，简称 Main）、“插件 A”的入口 vi（ProjectA－Main. vi，简称 PA）、“插件 B”的入口 vi（ProjecB－Main. vi，简称 PB）的子 vi 调用数和总 nodes。其中“插件 A”的代码较复杂，“插件 B”的代码相对简单，Main 是“通用采集平台”的主 vi，简单认为“通用采集平台”的所有代码均是可复用的。

在表 1 中，简明展示了 Main、PA、PB 调用的子 vi、

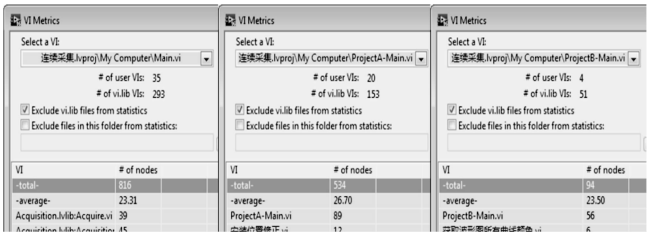

图 9　调用子 vi 及 nodes 统计

库函数 vi 的数量，及 Total Nodes（总节点数），这些统计量一定程度反应了各自 vi 的代码量，而 Main 的代码量占某个总代码量的比例，也一定程度上反应了代码的复用程度（占比越高，说明可复用的 Main 占项目代码量较多，代码复用性越好）。

表 1　Main、PA、PB 估算代码量及占比

vi 名		调用子 vi	调用库函数 vi	Total Nodes
Main		35	293	816
ProjectA		20	153	534
ProjectB		4	54	94
复用代码占总比	ProjectA	63.6%	65.6%	56.2%
	ProjectB	87.9%	84.4%	89.6%

4　结论

本文提出了一种以“通用采集平台”为基础、采用插件式编程模式实现针对项目定制化的“数据采集与处理软件”的设计方法，并给出了重要的代码的实现方法。使用该方法实现的“数据采集与处理软件”既有通用化采集软件的实用性，又可满足了特定任务的定制化需求。在项目开发上，由于“通用采集平台”的通用化设计，极大的提高了代码的复用性，使得该设计模式具有较大的实用价值。

参考文献：

[1] 耿玉玲，吕　强. 基于 TMS320VC33 的 PCI 高速数据采集系统设计 [J]. 自动化技术与应用，2005；(3)：34－37.
[2] 段彦肖. 教学车床位移数据采集与图形显示系统的研究 [D]. 天津：河北工业大学，2007.
[3] 贺正泽，何长明. 基于 LabVIEW 的井下弱磁信号检测系统 [D]. 天津：河北工业大学，2016.
[4] 李　娟. 传感器与测试技术 [M]. 北京：北京航空航天大学出版社，2007.
[5] 裴　郡. 基于 LabVIEW 的船模试验水池实时数据采集系统研究 [D]. 武汉：武汉理工大学，2015.
[6] 张素萍，李朝强. 基于 MSComm 和队列技术的 LabVIEW 数据采集系统设计 [J]. 国外电子测量技术，2016 (6)：26－31.
[7] 向力为. 基于 LabVIEW 的煤矿微震监测系统设计 [D]. 山东：山东大学，2017.
[8] 纳杰斯. LabVIEW 编程中基于 AMC 框架的多机通讯实现方法 [J]. 舰船电子工程，2016 (7)：51－55.
[9] 王林飞. 插件式地球物理软件开发平台（GeoProbe）设计、实现与应用 [D]. 北京：中国地质大学，2013.
[10] 吴　刚，李鸿君，许　娜. 面向复用的软件设计方法研究 [J]. 铁路计算机应用，2017 (3)：18－21.