

The Design and Implementation of Motor Control System Monitoring Software Based on WPF

Chonghan Hu¹, Hua Ye², Tingting Shi¹

¹Key Laboratory of Measurement and Control of CSE of Ministry of Education, Southeast University, Nanjing Jiangsu

²School of Automation, Southeast University, Nanjing Jiangsu

Email: erleyhu@foxmail.com, zhineng@seu.edu.cn, 977764362@qq.com

Received: Jan. 25th, 2016; accepted: Feb. 11th, 2016; published: Feb. 18th, 2016

Copyright © 2016 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Motor control system is the key to determine the performance of electric vehicles. In order to observe and analyze the effects of control more easily and to realize the modification of control parameters more conveniently and quickly so that the optimal control effect can be reached, the paper has designed and realized the motor control system monitoring software based on WPF. The software uses C# programming and makes full use of the multithreading technology and data binding technology of WPF, so that it provides a friendly man-machine interface, and realizes the function of high-speed real-time sampling, real-time display and real-time storage of the parameters of the motor control system and ensures the real-time monitoring of the running state of the motor control system. Firstly, the paper gives a modular design framework, and on this basis, the detailed design scheme of each module is introduced. Then the two key technologies containing multithreading and data binding are elaborated in detail. Finally, the running effect of the software is tested and the performance of the test is given. Through the test, it proves that all functions of the monitoring software meet the requirements of the design.

Keywords

WPF, Motor Control System, Monitoring Software, Multithreading, Data Binding

基于WPF的电机控制系统监控软件的设计与实现

胡崇晗¹, 叶 桦², 石婷婷¹

¹东南大学复杂工程系统测量与控制教育部重点实验室, 江苏 南京

²东南大学自动化学院, 江苏 南京

Email: erleyhu@foxmail.com, zhineng@seu.edu.cn, 977764362@qq.com

收稿日期: 2016年1月25日; 录用日期: 2016年2月11日; 发布日期: 2016年2月18日

摘 要

电机控制系统是决定电动汽车性能优劣的核心, 为了更加直观方便的观察和分析控制效果的优劣, 并方便快捷的实现控制参数的修改, 使控制效果达到最佳, 本文设计并实现了基于WPF的电机控制系统监控软件。软件采用C#编程, 充分利用了WPF的多线程技术和数据绑定技术, 为用户提供了友好的人机界面, 实现了对电机控制系统参数的高速实时采样、实时显示、实时保存等功能, 保证了对电机控制系统运行状态的实时监控。本文首先给出模块化的设计框架, 并在此基础上对各个模块的设计方案进行了具体的介绍, 然后详细阐述了软件中用到的多线程和数据绑定这两个关键技术, 最后对软件运行效果进行了测试, 给出了软件的运行效果图, 通过测试, 监控软件各部分功能满足设计的要求。

关键词

WPF, 电机控制系统, 监控软件, 多线程, 数据绑定

1. 引言

当今社会, 环境污染和能源危机日益加深, 电动汽车的兴起和飞速发展有着十分重要的意义, 而具有大功率、高效率和宽调速范围的电机是电动汽车的驱动心脏, 特别是在电池容量等技术还没有解决的前提下, 电机驱动系统是保证汽车续行里程并使之实际运用化的关键, 其控制效果直接影响电动汽车性能的优劣。依靠电机控制系统监控软件可以更加直观方便的观察和分析控制效果的优劣, 并方便快捷的实现控制参数的修改, 使控制效果达到最佳。因此, 设计一款可靠且稳定的电机控制系统监控软件就显得尤为重要, 它有助于提高技术人员的调试效率和故障排查能力, 提升电机控制系统的管理水平。

此软件属于虚拟仪器范畴, 虚拟仪器是当今仪器技术的发展热点, 许多研究者都提出了各自的设计方案。与一般的虚拟仪器相比, 用于电机控制系统实时监控的软件, 在功能上既有类似的地方, 也有不同之处, 它对数据采集和波形显示的实时性和稳定性要求更高。但是目前的电机控制系统监控软件, 有些并不支持示波器功能[1], 缺乏图形化的显示方式, 使用起来并不直观。就目前现有的虚拟仪器而言, 大都是利用 LabVIEW 配以高速数据采集卡来实现图形方式显示曲线[2]-[4], 功能比较完备, 可以进行波形显示、存储、频谱分析等, 但实时性较低, 可移植性较差。对于电机控制系统而言这种方案并不适用, 主要原因有: (1) 电机控制系统底层控制器大都已具备数据采集功能; (2) 能够同时显示的信号数较少, 不利于对多路信号的频率、相位等特性对比分析。

CANBUS 采用了许多新型技术及独特设计, 具有显著的可靠性、实时性和灵活性[5]。本文采用高实时性的 CANBUS 与电机控制系统进行通信, 使用高效可靠的基于 C#语言的 WPF 框架进行编程, 具有实时数据采集、实时波形显示、参数显示、参数配置以及事件和历史故障记录等功能。具有高实时性, 允许同时观测多路信号, 实现了对电机控制系统状态的实时监控, 为电机控制系统提供了实用的软件工具。

2. 软件总体设计方案

电机控制系统监控软件作为电机控制系统的附属部分，供内部人员生产测试及调试使用，主要用于电机控制系统产品维护，通过接口获取信息用于现场故障排查。该软件主要实现对控制系统多路参数的实时数据采集，并予以图形显示和文件存档，以及通过可视化界面，对控制系统的参数进行设置和状态进行监控分析。

本文涉及到的系统包括运行在 PC 上的监控软件、由控制器和电机构成的电机控制系统以及 IXXAT 公司的 USB-to-CAN 构成，如图 1 所示。通过用户与监控软件的交互操作，系统可借助 USB-to-CAN 接口卡与 CAN 总线实现 PC 机与电机控制系统的各种数据交互，从而进行对电机运行情况的监控。

2.1. 软件结构

为了提高系统的通用性和可扩展性，方便程序设计和升级，软件采用模块化的设计思想，按层次分为底层 CAN 总线通信模块处理、中层数据处理和上层界面三大模块。底层 CAN 通信层负责接收控制系统上传的数据和向控制系统下发命令。数据处理层负责解析控制系统上传的数据、封装下发的命令和存储数据到文件等。界面层通过图形、列表、编辑框控件等显示方式把获取的变流器信息呈现给用户，并予以保存，同时接受用户的操作，对数据进行安全检查等。使用人员通过操作菜单，查看各窗口的参数信息，并对控制系统参数进行设置。电机控制系统监控软件数据流图如图 2 所示。

2.2. 软件开发环境

该软件采用 Visual Studio 2013 的 WPF 应用程序的开发环境，WPF (Windows Presentation Foundation) 是微软推出的基于 Windows Vista 的用户界面框架[6]，属于 .NET Framework 3.0 的一部分。它提供了统一的编程模型、语言和框架，真正做到了分离界面设计人员与开发人员的工作；同时它提供了全新的多媒体交互用户图形界面。

3. 软件详细设计

通信协议是实现监控软件和控制系统进行信息交互的基础。软件采用模块化的设计方法，将底层通



Figure 1. System connection diagram

图 1. 系统连接示意图

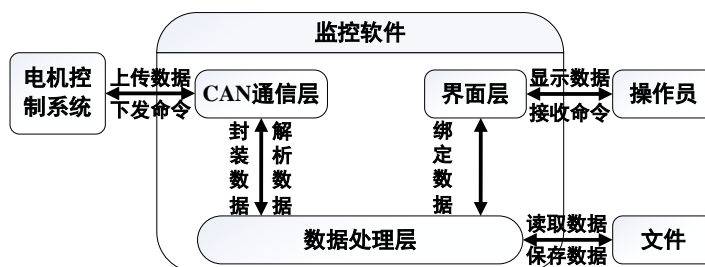


Figure 2. Software design schematic diagram

图 2. 软件设计原理图

信模块和功能模块分别进行封装。底层通信模块用于通过 USB-to-CAN 与控制系统节点进行信息交互，功能模块用于软件与用户进行信息交互。下面分别进行介绍。

3.1. 通讯协议的设计

3.1.1. 通信报文的设计

监控软件和电机控制系统之间的通信方式分为两种：主从问答方式和自动上传方式。主从问答方式是指主控设备(监控软件)向受控设备(控制系统)发送命令帧，受控设备(控制系统)接收到命令帧后，向主控设备(监控软件)发送响应帧，从而实现数据交换；自动上传方式是指受控设备(控制系统)定时上传数据给主控设备(监控软件)，主控设备(监控软件)只需要接收，无需发送响应帧。

监控软件和电机控制系统之间的通信报文有三种，控制报文、设置报文与数据报文。其中，控制报文用于对电机控制系统运行状态进行控制，如控制电机控制系统运行于操作模式或预操作模式；设置报文用于设置电机控制系统的各项参数；数据报文规定了控制系统定时上传数据的格式。从报文功能来看，控制报文和设置报文需要控制系统的响应，因此，这两种报文采用主从问答方式，而数据传输时通信速率较高，问答式效率低，而且如果要保证毫秒级别的精确定时，监控软件 CPU 占有率会很高，因此控制系统采用主动定时 1 ms 的方法上传数据。

3.1.2. 通信报文格式的制定

监控软件与电机控制系统通信报文格式采用 CAN2.0A 标准数据帧格式进行设计[7]，在设计时需要指明 Identifier 及数据字段各个字节所代表的具体含义。具体内容根据通信报文分别制定，具体格式见表 1。其中控制报文 Identifier 以 0x00 开头，X 根据不同控制命令设置为不同值，数据部分长度为 1，数据值也于控制命令相关；同理设置报文和数据报文的 Identifier 以 0x40 和 0x80 开头，数据部分长度为 1，数据部分为具体参数值。

3.1.3. 浮点型数据处理方法

设置报文和数据报文中的参数有整形和浮点型之分。对于整形参数，无需特殊处理，即可实现数据的无精度损失的发送和接收；对于浮点型参数，若经过特殊数据处理，也可实现无精度损失的发送和接收。

float 类型变量 dataFloat 在内存中占 32 位，若把 32 位 float 类型数据以 32 位 int 类型数据 dataInt 读取，即可实现无精度损失的发送，具体方法见公式(1)；同理把 32 位 int 类型变量 dataInt 以 32 位 float 类型数据 dataFloat 读取，即可实现无精度损失的接收，具体方法见公式(2)。

$$dataInt = *(int*)&dataFloat \quad (1)$$

$$dataFloat = *(int*)&dataInt \quad (2)$$

3.2. 底层通讯模块的设计

底层通讯模块作为监控软件与电机控制系统信息交互的通道，在整个软件设计中起着至关重要的作

Table 1. Frame format of communication message
表 1. 通信报文帧格式

	Identifier	DataLength
控制报文	0x00X	1
设置报文	0x40X	8
数据报文	0x80X	8

用。根据通讯协议，通讯方式有两种：主从问答方式和主动上传方式。对于主动上传方式，底层通讯接口相关函数只负责接收数据；对于主从问答方式，底层通讯接口相关函数先发送数据给控制系统节点，然后采用查询方式接收控制系统上传的数据。底层通讯模块功能如图 3 所示。

3.3. 功能模块的设计

为了实现监控软件的各项功能，本软件设计了包括参数显示、参数配置、示波器以及事件和历史故障记录等功能模块。

示波器功能模块主要用于监控控制系统的控制效果，通过通信接口实时采集数据并将接收到的数据以图形化方式连续动态的显示在示波器窗口中。

参数显示功能模块支持控制系统运行参数及状态的动态显示，实现对控制系统进行实时动态监控分析，状态周期刷新时间用户可自行设定，默认为 0.33 s。

参数配置功能模块实现对控制系统运行模式的控制功能，使得控制系统运行于操作模式或预操作模式，并在预操作模式下实现对重要参数的配置功能。

事件和历史故障记录功能模块实现控制系统事件和历史故障记录功能，可手动实时更新事件和历史故障记录，可以导出为 CSV 文件，支持默认文件名方式保存。

4. 软件的关键技术

4.1. 多线程技术

电机控制系统监控软件是对电机控制系统的状态进行监控，具有多任务的特点。采用单线程运行时，数据传输和采集占用的资源过多导致速度慢、延迟等缺陷，而采用多线程技术和消息映射机制可有效地同步完成不同的任务，使应用程序的并行处理得以实现[8]，并且可以满足实时数据传输、及时响应用户操作的需要，因此将多线程编程技术应用于此监控软件的多任务中。

4.1.1. 多线程技术简介

在 WPF 应用程序中，线程有两种类型：用户界面线程和工作者线程两大类。用户界面线程可以产生自己的窗口并负责处理相关的窗口消息，拥有单独的消息队列，能够对事件和用户输入做出响应。工作者线程适用于处理后台任务，而不影响用户对应用程序的使用，工作者线程仅仅由一个函数体实现，其实现简单，便于编程者控制，适用于一些费时的后台任务处理在多线程应用程序中。

两个或更多的线程同时访问同一数据会导致不可预知的结果，因此保持线程间的同步是一个不可或缺的环节。使隶属于同一进程的各线程协调一致的工作称为线程的同步。Windows 系统提供了多种同步方法：临界区、信号灯、互斥量和事件等[9]，在 C#语言中均有相应的实现。

消息映射就是消息与消息处理函数一对一的联系，一旦有消息产生，程序就会调用相应的消息响应函数来进行处理。

4.1.2. 多线程技术在监控软件中的应用

在设计过程中，高速实时数据采集与实时数据处理共同进行是软件面临的一大难题。应数据采集

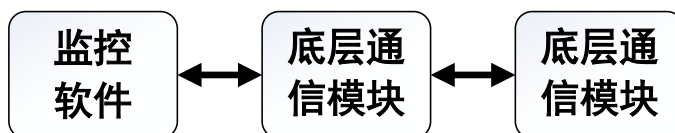


Figure 3. Communication module diagram
图 3. 底层通讯模块功能示意图

与处理的实时性、可靠性、完整性的要求，本文采用了多线程技术将数据接收和数据处理放在不同的线程中，实现程序的并行处理。数据处理流程如下：

(1) 在数据采集线程中，定时调用(本文中调用周期为 1 ms)底层通信接口函数，读取电机控制系统(下位机)发送的数据，根据需要存入不同的数据缓冲区(如参数显示数据缓冲区、示波器数据缓冲区等)；

(2) 各个数据处理线程，根据需要定时读取(本文中对实时性要求较低的参数显示模块定时周期为 500 ms，而对实时性要求比较高的示波器模块定时周期为 100 ms)相应数据缓冲区中的数据，并进行不同的处理；

(3) 由于数据采集线程和数据处理线程会同时访问数据缓冲区，因此保持线程间的同步是软件设计过程中一个不可或缺的环节。C#中的 lock 关键字封装了 Windows 中临界区的功能[10]，本文中使用 lock 关键字，使得任意时刻只有一个线程可以拥有临界区对象，拥有临界区的线程可以访问被保护起来的资源，其它希望进入临界区的线程将被挂起等待，直到拥有临界区的线程放弃临界区时为止，这样就保证了不会在同一时刻出现多个线程访问共享资源。

数据采集线程和数据处理线程流程图如图 4 所示。

4.2. 数据绑定技术

数据绑定是在应用程序 UI 与业务逻辑之间建立连接的过程[11]。如果绑定具有正确设置并且数据提供正确通知，则当数据更改其值时，绑定到数据的元素会自动反映更改。数据绑定可能还意味着如果元素中数据的外部表现形式发生改变，基础数据可以自动更新以反映更改。例如，如果用户编辑 Textbox 元素中的值，基础数据值会自动更新以反映该更改。

数据绑定的一种典型用法是将服务器或本地配置数据放置到窗体或其他 UI 控件中。在 WPF 中，此概念得到扩展，包括了大量属性与各种数据源的绑定。在 WPF 中，元素的依赖项属性可以绑定到 CLR

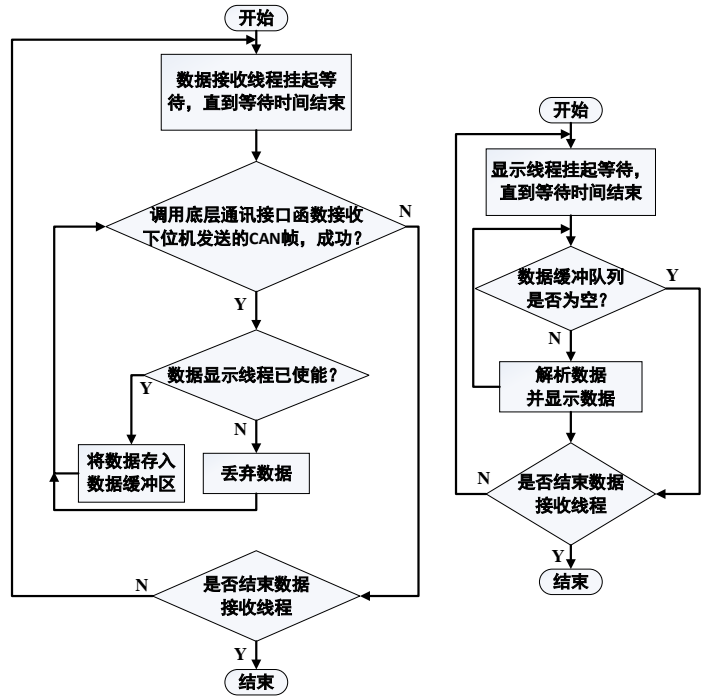


Figure 4. Flow chart of data acquisition thread and data processing thread
图 4. 数据采集线程和数据处理线程流程图

对象(包括 ADO.NET 对象或与 Web 服务和 Web 属性相关联的对象)和 XML 数据。

4.2.1. 数据绑定技术简介

数据绑定实质上是绑定目标与绑定源之间的桥梁。不论要绑定什么元素,不论数据源的特性是什么,每个绑定都始终遵循图 5 所示的模型。

该图演示了以下基本的 WPF 数据绑定概念:

(1) 通常,每个绑定都具有四个组件:绑定目标对象、目标属性、绑定源,以及要使用的绑定源中的值的路径。

(2) 目标属性必须为依赖项属性。

(3) 绑定源对象并不限于自定义 CLR 对象。

数据绑定分为单向绑定、双向绑定、单向源绑定和单次绑定。其中,单向绑定将导致对源属性的更改会自动更新目标属性,但是对目标属性的更改不会传播回源属性;双向绑定将导致对源属性的更改会自动更新目标属性,而对目标属性的更改也会自动更新源属性;单向源绑定与单向绑定相反;它在目标属性更改时更新源属性;单次绑定会导致源属性初始化目标属性,但不传播后续更改,这意味着如果数据上下文发生了更改或者数据上下文中的对象发生了更改,则更改不会反映在目标属性中。

触发源更新的原因有多种,由绑定目标的 UpdateSourceTrigger 属性决定。如果 UpdateSourceTrigger 值为 PropertyChanged,则双向绑定或单向源绑定的右箭头指向的值会在目标属性更改时立刻进行更新;如果 UpdateSourceTrigger 值为 LostFocus,则仅当目标属性失去焦点时,该值才会使用新值进行更新。

4.2.2. 数据绑定技术在监控软件中的应用

数据绑定技术在本文监控软件中的应用几乎无处不在,软件中用于数据展示的控件,几乎都有数据源或是其他控件与其绑定。例如,参数配置功能模块的 DataGrid 控件与 ObservableCollection 类型的集合类绑定;参数显示功能模块的 ProgressBar 控件与自定义数据类型 VehDetectParameters 的数据绑定,同时 TextBlock 控件又与 ProgressBar 控件进行数据绑定等。

此处以参数配置功能模块的 DataGrid 控件与 ObservableCollection 类型的集合类绑定为例,分析数据绑定的功能。当需要读取电机控制系统参数时,CAN 通信层读取控制系统上传的数据,更新数据处理层 ObservableCollection 类型的集合类的数据后界面层 DataGrid 控件中展示的数据也会自动改变。相反,当需要配置电机控制系统参数时,用户改变界面层 DataGrid 控件中的数据后,数据处理层的数据源的数据也会自动改变,并通过 CAN 通信层将数据传出。

总之,通过在监控软件中使用数据绑定技术,使得程序员可以专心于业务逻辑的开发,而减少控件操作带来的繁琐工作,在降低了代码维护难度的同时提高了代码的重用性,无疑增加了软件开发的效率。

5. 软件测试

本文测试平台即图 1 所示系统,在电机控制系统中,控制器对电机进行单闭环控制,监控软件和电



Figure 5. Data binding model chart

图 5. 数据绑定模型图

机控制系统通过 CAN 总线进行通信。需要测试监控软件对电机控制系统各项参数进行显示和配置、对控制效果的波形进行显示以及对事件和历史故障进行记录等功能。

将监控软件运行于 Windows 操作系统平台上，并对以上的功能进行了测试，结果表明该软件各功能模块运行正常稳定。运行效果如图 6 所示。

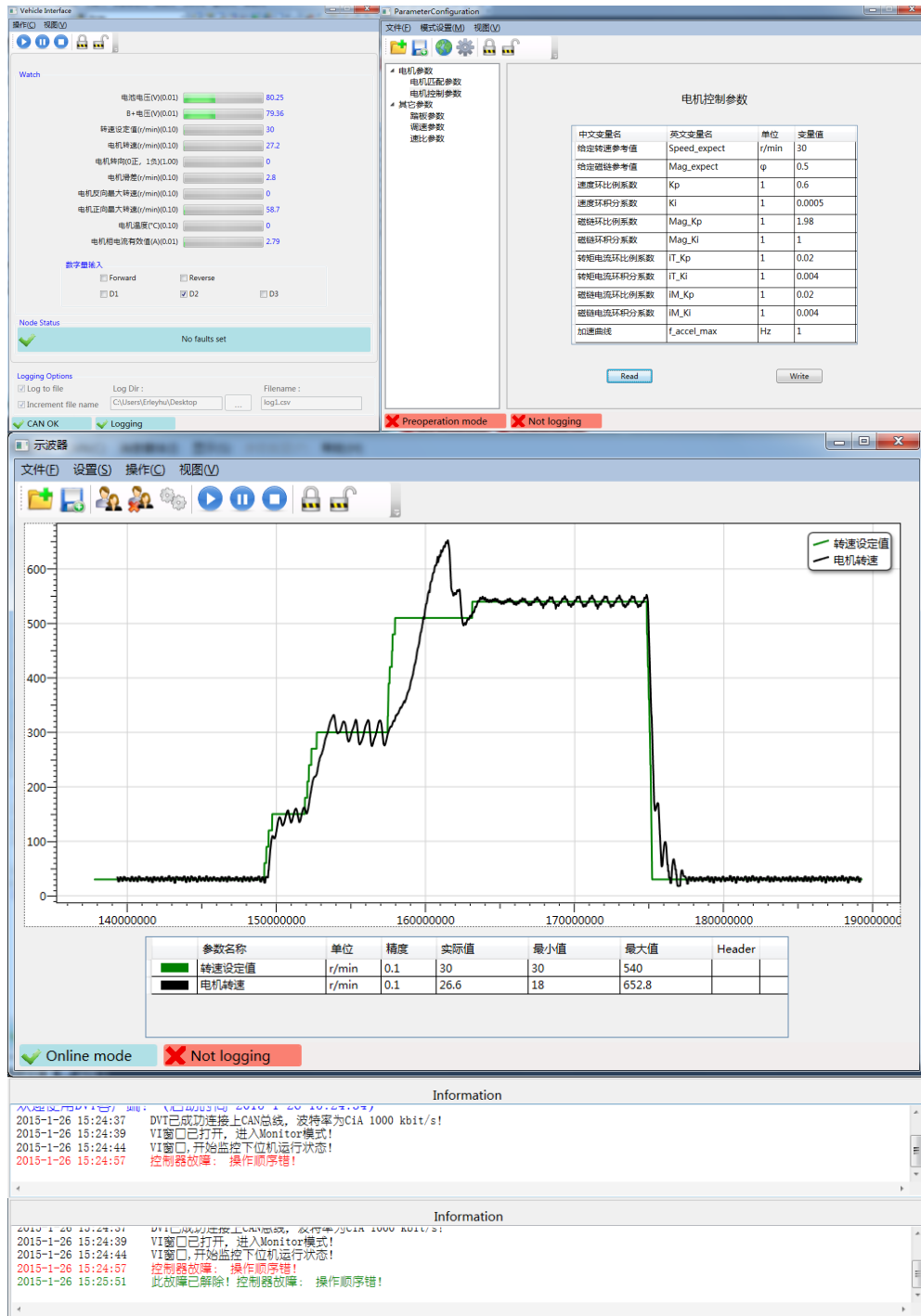


Figure 6. Software running effect

图 6. 软件运行效果

6. 结束语

本文设计并实现了一款基于 WPF 的电机控制系统监控软件,电机控制系统和监控软件利用 CANBUS 总线进行通信,软件设计中运用 C#语言进行编程,并运用了多线程技术和数据绑定技术等关键技术,实现了 CAN 通信层、数据处理层和界面层的设计,具有参数显示、参数配置、示波器以及事件和历史故障记录等功能。由于电机控制系统监控软件界面美观、成本低、功能强、扩展性好、可移植性高,因此具有较强的实用价值和广阔的市场前景。

参考文献 (References)

- [1] 董海燕. 基于 CAN 通信的风机变流器监控技术的研究[D]: [硕士学位论文]. 淮南: 安徽理工大学, 2009.
- [2] Dong, S.L., Han, J.F., Pan S.H. and Deng, J.F. (2011) Design of Virtual Oscilloscope Based LabVIEW. *Communications in Computer and Information Science*, **244**, 34-41.
- [3] 宁歆. 基于 LabVIEW 的虚拟数字示波器的设计与实现[D]: [硕士学位论文]. 广州: 第一军医大学, 2006.
- [4] Wei, J. and Fang, Y. (2009) Design of Oscilloscope Based on Virtual Instrument Technique. *2nd International Conference on Power Electronics and Intelligent Transportation System*, Shenzhen, 19-20 December 2009, 284-287
- [5] 李建林, 张仲超. CANBUS 总线简介及其在电力系统中的应用[J]. 电力系统及其自动化学报, 2002, 14(5): 69-72.
- [6] 王婵娟. 浅谈 WPF 架构及其应用[J]. 科技信息, 2011(21): 114-115.
- [7] Tan, Y., Wang, J.F., Qiu, C.D. and Li, Z. (2006) The Intelligent Online Monitor System for Battery Based on CANBUS. *Proceedings of the 6th World Congress on Intelligent Control and Automation*, **1**, 5070-5074.
- [8] 王荣存, 唐翔, 蔺超文. 基于多线程的太阳能热力监控系统的设计与实现[J]. 测控技术, 2009, 28(8): 79-82.
- [9] Tanenbaum, A.S. 著. 现代操作系统[M]. 陈向群, 马洪兵, 译. 北京: 机械工业出版社, 2009.
- [10] 里伯提(Liberty, J.). Programming C# [M]. 中文版. 北京: 电子工业出版社, 2010.
- [11] 麦克唐纳(MacDonald, M.). WPF 编程宝典[M]. 北京: 清华大学出版社, 2013.