



- 程序源代码
- 程序运行录屏
- 系统测试录像
- 软硬件资源

MATLAB

数据通信与测控 应用编程实践

邓红涛 李 伟
李传峰 李江全 等编著

- 选用数据通信与测控领域的**8种常用硬件**
- 深入讲解数据通信与测控领域的**4类典型应用**
- 提供**超值光盘**



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

MATLAB 数据通信与 测控应用编程实践

邓红涛 李 伟 李传峰 李江全 等编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书从工程应用的角度出发,通过 8 种典型的数据采集与控制系统,包括基于单片机、PLC、PCI 数据采集卡、远程 I/O 模块、USB 数据采集板、无线数据传输模块、GSM 短信模块和智能仪器等,使用 MATLAB 编程语言的 DAQ 工具箱函数、ActiveX 控件、串口 I/O 流和 MSComm 控件,对数据采集与通信控制系统中的 4 类典型应用:模拟量输入(AI)、模拟量输出(AO)、数字量输入(DI)和数字量输出(DO)的程序设计方法进行了讲解。

本书内容丰富,提供的实例有具体的设计任务、完整的程序代码,注重解决工程实际问题,可供自动化、计算机应用、机电一体化等各类专业的大学生、研究生学习 MATLAB 数据通信与测控技术时使用,也可供计算机控制系统研发的工程技术人员参考。

为方便读者学习,本书提供超值配套光盘,内容包括所有实例的源程序、程序运行录屏、系统测试录像、软硬件资源等。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

MATLAB 数据通信与测控应用编程实践 / 邓红涛等编著. —北京:电子工业出版社, 2011.9
ISBN 978-7-121-14456-1

I. ①M… II. ①邓… III. ①可编程序控制器—数据通信—计算机辅助计算—软件包, MATLAB
IV. ①TM571.6②TN919

中国版本图书馆 CIP 数据核字 (2011) 第 172708 号

责任编辑:田宏峰 特约编辑:牛雪峰

印 刷: 北京市李史山胶印厂

装 订:

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 26.25 字数: 705 千字

印 次: 2011 年 9 月第 1 次印刷

册 数: 4 000 册 定价: 59.00 元 (含 DVD 光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

MATLAB 是当今世界上使用最为广泛的数学软件之一,具有相当强大的数值计算、数据处理、系统分析、图形显示甚至符号运算等功能,是一个完整的数学平台。同时,它又是面向工程与科学计算的高级语言,可以设计出功能强大、界面优美、稳定可靠的高质量程序,编程效率和计算效率极高。

MATLAB 在以下的领域里解决各种问题是一个十分有效的工具:工业研究与开发;数学教学,特别是线性代数,所有的基本概念都能涉及;在数值分析和科学计算方面的教学与研究,能够详细地研究和比较各种算法;在电子学、控制理论和物理学等工程和科学学科方面的教学与研究;在经济学、化学和生物学等有计算问题的所有其他领域中的教学与研究。

MATLAB 不仅作为科学数值计算软件而被使用,事实上,**MATLAB** 还提供了一系列的面向各个工程领域的功能极其强大的工具箱,使得 **MATLAB** 已经成为科研界的一个通用软件,有极其庞大的用户群和开发群。

作为一个科研工作者,应该学会利用 **MATLAB** 进行数据采集。**MATLAB** 环境下的数据采集工具箱可以简化和加快数据的采集工作。使用这种工具箱更容易将实验测量、数据分析和可视化的应用集合在一起。数据采集工具箱提供了一整套的命令和函数,通过调用这些命令和函数,可以直接控制各种与 PC 兼容的硬件设备的数据采集和通信。

本书从工程应用的角度出发,通过 8 种典型的数据采集与控制系统,包括基于单片机、PLC、PCI 数据采集卡、远程 I/O 模块、USB 数据采集板、无线数传模块、GSM 短信模块和智能仪器等,使用 **MATLAB** 编程语言的 DAQ 工具箱函数、ActiveX 控件、串口 I/O 流和 MSComm 控件,对数据采集与通信控制系统中的 4 类典型应用:模拟量输入(AI)、模拟量输出(AO)、数字量输入(DI)和数字量输出(DO)的程序设计方法进行了讲解。

本书内容丰富,提供的实例有具体的设计任务、完整的程序代码,注重解决工程中的实际问题,可供自动化、计算机应用、机电一体化等各类专业的大学生、研究生学习 **MATLAB** 数据通信与测控技术,也可供计算机控制系统研发的工程技术人员参考。

为方便读者学习,本书提供超值配套光盘,内容包括所有案例的源程序、软硬件资源、程序运行录屏、系统测试录像等。

本书由塔里木大学李传峰编写第 1 章,万畅编写第 2 章,李平编写第 3 章,李向阳编写第 4 章,石河子大学葛云编写第 5 章,邓红涛编写第 6、7 章,李伟编写第 8、9、10 章,李江全编写第 11 章,全书由邓红涛担任主编,李传峰、李伟担任副主编,由李江全教授统稿。参与本书编写工作的人员还有田敏、郑瑶、刘恩博、李宏伟、郑重、任玲、王洪坤、汤智辉等老师。北京研华科技、西安达泰电子、电子开发网、石河子大学电气工程实验中心等单位或公司为本书的编写提供了宝贵的技术支持和帮助,编者借此机会对他们致以深深的谢意。

由于编者水平有限,书中难免存在不妥或错误之处,恳请广大读者批评指正。

目 录

第 1 章	MATLAB 数据采集与串口通信基础	1
1.1	MATLAB 的数据采集工具箱	1
1.1.1	数据采集工具箱简介	1
1.1.2	数据采集工具箱函数	4
1.1.3	数据采集输入设备的对象属性	7
1.1.4	MATLAB 数据采集步骤	8
1.2	数据采集卡 MATLAB 编程	11
1.2.1	概述	11
1.2.2	模拟量输入 (AI 或 A/D)	13
1.2.3	模拟量输出 (AO 或 D/A)	17
1.2.4	使用触发器功能	19
1.2.5	使用事件和回调	20
1.3	MATLAB 串口通信	21
1.3.1	MATLAB 串口简介	21
1.3.2	开始使用串口 I/O 流	22
1.3.3	串口对象的建立	26
1.3.4	数据的读写	29
1.3.5	事件和函数回调	37
1.3.6	使用控制针	41
1.3.7	记录数据到磁盘	44
1.3.8	保存和装载数据	48
1.3.9	串口对象属性参考	49
第 2 章	基于研华数据采集卡的测控应用	51
2.1	PCI-1710HG 数据采集卡简介	51
2.1.1	数据采集系统概述	51
2.1.2	基于 PC 的 DAQ 系统组成	54
2.1.3	用 PCI-1710HG 数据采集卡组成的测控系统	57
2.1.4	PCI-1710HG 数据采集卡的安装与测试	59
2.2	系统设计说明	64
2.2.1	设计任务	64
2.2.2	线路连接	65
2.3	数据采集与控制程序设计	66
2.3.1	模拟量输入	66
2.3.2	模拟量输出	69
2.3.3	数字量输入	72
2.3.4	数字量输出	74

2.3.5	温度测控	75
第 3 章	基于 NI 数据采集卡的测控应用	83
3.1	PCI-6023E 数据采集卡简介	83
3.1.1	PCI-6023E 数据采集卡的功能	83
3.1.2	安装 DAQ 设备驱动程序	84
3.1.3	数据采集卡的参数设置与测试	85
3.2	系统设计说明	89
3.2.1	设计任务	89
3.2.2	线路连接	89
3.3	数据采集与控制程序设计	90
3.3.1	模拟量输入	90
3.3.2	数字量输入	94
3.3.3	数字量输出	95
3.3.4	温度测控	97
第 4 章	基于西门子 PLC 的测控应用	105
4.1	西门子 PLC 模拟量扩展模块与通信协议	105
4.1.1	西门子 PLC 模拟量输入模块	105
4.1.2	西门子 PLC PPI 通信协议	108
4.2	系统设计说明	112
4.2.1	设计任务	112
4.2.2	线路连接	113
4.3	数据采集与控制程序设计	114
4.3.1	模拟量输入	114
4.3.2	模拟量输出	123
4.3.3	数字量输入	126
4.3.4	数字量输出	138
4.3.5	温度测控	149
第 5 章	基于三菱 PLC 的测控应用	157
5.1	三菱 PLC 特殊功能模块与通信协议	157
5.1.1	FX _{2N} 系列 PLC 的特殊功能模块	157
5.1.2	三菱 PLC 编程口通信协议	163
5.2	系统设计说明	170
5.2.1	设计任务	170
5.2.2	线路连接	170
5.3	数据采集与控制程序设计	172
5.3.1	模拟量输入	172
5.3.2	模拟量输出	177
5.3.3	数字量输入	181
5.3.4	数字量输出	196
5.3.5	温度测控	209
第 6 章	基于单片机的测控应用	216

6.1	典型单片机开发板简介	216
6.1.1	单片机控制系统的组成	216
6.1.2	单片机开发板 B 的功能	218
6.1.3	单片机开发板 B 的主要电路	219
6.2	系统设计说明	222
6.2.1	设计任务	222
6.2.2	线路连接	222
6.3	数据采集与控制程序设计	226
6.3.1	模拟量输入	226
6.3.2	模拟量输出	240
6.3.3	数字量输入	253
6.3.4	数字量输出	261
6.3.5	温度测控	266
第 7 章	基于分布式 I/O 模块的测控应用	284
7.1	典型分布式 I/O 模块简介	284
7.1.1	集散控制系统的结构与特点	284
7.1.2	ADAM4000 远程数据采集控制系统	286
7.1.3	ADAM4000 系列模块简介	288
7.2	系统设计说明	297
7.2.1	设计任务	297
7.2.2	线路连接	297
7.3	数据采集与控制程序设计	298
7.3.1	模拟量输入	298
7.3.2	模拟量输出	300
7.3.3	数字量输入	302
7.3.4	数字量输出	305
7.3.5	温度测控	309
第 8 章	基于 USB 数据采集模块的测控应用	314
8.1	USB 总线在数据采集系统中的应用	314
8.1.1	USB 总线及其数据采集系统的特点	314
8.1.2	采用 USB 传输的数据采集系统	316
8.1.3	典型 USB 数据采集模块简介	318
8.2	系统设计说明	320
8.2.1	设计任务	320
8.2.2	线路连接	320
8.3	数据采集与控制程序设计	321
8.3.1	模拟量输入	321
8.3.2	模拟量输出	324
8.3.3	数字量输入	327
8.3.4	数字量输出	329
8.3.5	温度测控	330

第 9 章 基于 GSM 短信模块的测控应用	338
9.1 GSM 网络短信测控技术	338
9.1.1 GSM 短信测控系统的特点与组成	338
9.1.2 AT 指令介绍	342
9.2 系统设计说明	346
9.2.1 设计任务	346
9.2.2 线路连接	346
9.3 数据采集与控制程序设计	348
9.3.1 利用 C51 语言实现单片机温度检测及短信发送	348
9.3.2 利用 C51 语言实现单片机短信接收及继电器控制	354
9.3.3 利用 MATLAB 实现 PC 短信接收与发送	360
第 10 章 基于无线数据传输模块的测控应用	367
10.1 典型无线数据传输模块简介	367
10.1.1 无线数据传输技术概述	367
10.1.2 DTD46X 系列无线数据传输模块	369
10.2 系统设计说明	374
10.2.1 设计任务	374
10.2.2 线路连接	374
10.3 温度测量与控制程序设计	375
10.3.1 利用 C51 语言实现基于 DS18B20 的单片机温度测控	375
10.3.2 利用汇编语言实现基于 DS18B20 的单片机温度测控	382
10.3.3 利用 MATLAB 实现无线数据传输模块温度测控	389
第 11 章 基于智能仪器的测控应用	394
11.1 典型智能仪器简介	394
11.1.1 智能仪器的结构与特点	394
11.1.2 XMT-3000A 型智能仪器的通信协议	395
11.1.3 PC 与 XMT-3000A 型智能仪器串口通信调试	398
11.2 系统设计说明	399
11.2.1 设计任务	399
11.2.2 线路连接	399
11.3 温度测量与控制程序设计	402
11.3.1 利用 MATLAB 实现 PC 与单台智能仪器温度测控	402
11.3.2 利用 MATLAB 实现 PC 与多台智能仪器温度测控	405
参考文献	412

第 1 章 MATLAB 数据采集 与串口通信基础

MATLAB 是一种功能十分强大、运算效率很高的数学工具软件，MATLAB 不仅可作为科学数值计算软件而被使用，事实上，MATLAB 提供了一系列的面向各个工程领域的功能极其强大的工具箱，使得 MATLAB 已经成为科研界的一个通用软件，有极其庞大的用户群和开发群。作为一名科研工程工作者，应该学会利用 MATLAB 进行数据采集。

1.1 MATLAB 的数据采集工具箱

MATLAB 环境下的数据采集工具箱可以简化和加快数据的采集工作，使用这种工具箱更容易将实验测量、数据分析和可视化的应用集合在一起。数据采集工具箱提供了一整套的命令和函数，通过调用这些命令和函数，可以直接控制各种与 PC 兼容的硬件设备的数据采集和通信。

1.1.1 数据采集工具箱简介

MATLAB 环境下的数据采集工具箱可以简化和加快数据的采集工作，使用这种工具箱更容易将实验测量、数据分析和可视化的应用集合在一起。数据采集工具箱提供了一整套的命令和函数，通过调用这些命令和函数，可以直接控制各种与 PC 兼容的硬件设备的数据采集和通信。

每一种厂商的硬件数据采集设备的内部特性对 MATLAB 接口来说都是透明的。无论使用一个或几个硬件设备，数据采集工具箱会向所有的硬件设备提供单一的和统一的接口。使用一整套熟悉的 MATLAB 命令可以对单个接口板进行访问并对它的属性进行完全的控制。

因为数据采集工具箱集成在 MATLAB 之中，所以在进行数据采集的同时，可以对采集的数据进行实时分析，或者存储后进行处理，或者针对数据分析的需要对测试条件的设立进行不断更新，还可以利用 MathWorks 的特殊应用工具箱和 Simulink 工具箱的功能提供额外的分析和建模能力。应用数据采集工具箱提供的函数和命令可以控制任何类型的数据采集。例如，在硬件设备运行时，可以获取事件信息，评估采集状态，定义触发器和回访状态，预览数据以及进行实时分析，可以设置和显示所有的硬件特性以满足用户的技术指标。

1. 数据采集工具箱组件

数据采集工具箱包含有三种组件：M 文件函数（M-File Functions）、数据采集引擎（Data Acquisition Engine）和硬件驱动接口（Hardware Driver Adaptors）。MATLAB 程序通过这三种组件实现与数据采集硬件（Data Acquisition Hardware）的互连与信息传递，它们之间的关系如图 1-1 所示。

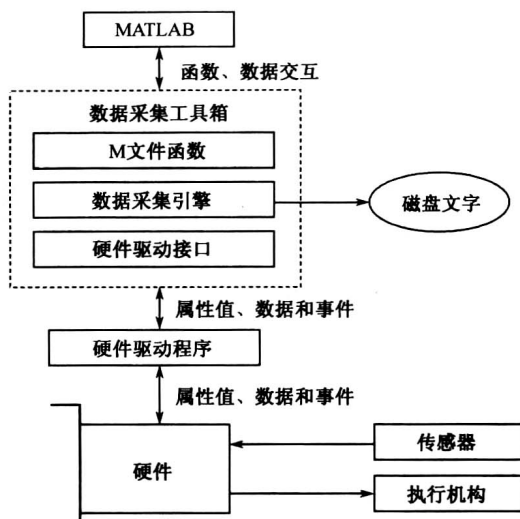


图 1-1 MATLAB 数据采集工具箱提供的三种组件与硬件之间的关系图

在 MATLAB 工具箱、硬件驱动程序和硬件之间传递的信息内容包括属性值 (Property Values)、数据 (Data) 和事件 (Event)。通过属性值可以配置数据采集程序的设置；数据是我们采集的信号 (或者要输出的信号)；事件由数据采集系统底层应用产生，可以用来触发指定的回调函数。

1) M 文件函数 (M-File Functions)

任何一个 MATLAB 的应用程序都必须使用 M 文件。进行数据采集之前，我们必须在 MATLAB 环境下调用一个或一些由 M 文件所构成的函数，以完成特定的数据采集任务。

在数据采集应用系统中，使用 M 文件可以实现以下功能：

- 创建设备对象，为硬件提供功能性的通路和控制数据采集过程；
- 通过将信息传递给硬件或从硬件获取信息来控制采集过程；
- 通过来自硬件的特定信息获得数据采集过程中的一些状态。

2) 数据采集引擎 (Data Acquisition Engine)

数据采集引擎是 MEX 文档格式的动态链接库 (DLL) 文件。数据采集引擎的功能如下：

- 存储设备对象及其属性值，用于控制数据采集；
- 控制事件同步；
- 负责采集数据的存储以及数据的输出。

当 DAQ 引擎执行数据采集任务时，MATLAB 能够用于分析所获得的数据等其他工作，这就是说，DAQ 引擎或 MATLAB 是异步工作的。

3) 硬件驱动接口 (Hardware Driver Adaptors)

硬件驱动接口负责在数据采集引擎和硬件驱动程序之间传递信息。在使用 MATLAB 的数据采集工具箱进行数据采集之前必须在操作系统里装上数据采集硬件的驱动程序。目前，MATLAB 的数据采集工具箱提供的数据采集硬件接口有以下 7 种。

- windows: 支持系统声卡；
- parallel: 支持系统并口；



- nidaq: 支持 NI (National Instruments) 的产品;
- mcc: 支持 Measurement Computing 产品;
- keithley: 支持 Keithley 产品;
- hpel432: 支持 Agilent Technologies 产品;
- advantech: 支持 Advantech 产品。

2. 检查数据采集工具箱信息

1) 查看一般信息

可以使用 `daqhwinfo` 命令获得数据采集工具箱的信息。

```
>>out=daqhwinfo
out =
    ToolboxName: 'Data Acquisition Toolbox'
    ToolboxVersion: '2.7 (R14SP3)'
    MATLABVersion: '7.1 (R14SP3)'
    InstalledAdaptors: {4x1 cell}
```

从中可以看出, `InstalledAdaptors` 里包含的是系统已经安装了的数据采集硬件, 可用如下命令进行详细查看。

```
>> out.InstalledAdaptors
ans =
    'advantech'
    'nidaq'
    'parallel'
    'winsound'
```

由上可以看到计算机里除了安装了声卡, 以外还安装了其他数据采集设备。

2) 查看硬件信息

要查看某一硬件的具体信息, 使用带参数的 `daqhwinfo` 即可, 可用下面的命令查看声卡的具体信息。

```
>> out=daqhwinfo('winsound')
out =
    AdaptorDllName: [1x64 char]
    AdaptorDllVersion: '2.7 (R14SP3)'
    AdaptorName: 'winsound'
    BoardNames: {'Realtek AC97 Audio'}
    InstalledBoardIds: {'0'}
    ObjectConstructorName: {1x3 cell}
```

由上可以看出, `ObjectConstructorName` 里面包含有声卡支持的设备对象创建函数名。用下面命令可以查看具体的创建函数。

```
>> out.ObjectConstructorName(:)
ans =
```



```
'analoginput('winsound',0)'  
'analogoutput('winsound',0)'  
,
```

从显示的信息中可知,这款声卡支持模拟输入和模拟输出,并且还提示了应该如下创建对象:

- 创建输入对象 `ai=analoginput('winsound');`
- 创建输出对象 `ao=analogoutput('winsound');`

3) 查看设备对象信息

要查看某一特定设备对象的信息,使用已经创建的设备对象作为参数调用函数 `daqhwinfo` 即可,运行如下命令。

```
>>ai=analoginput('winsound');  
>> out=daqhwinfo(ai)  
out =  
  
    AdaptorName: 'winsound'  
        Bits: 16  
    Coupling: {'AC Coupled'}  
    DeviceName: 'Realtek AC97 Audio'  
DifferentialIDs: []  
        Gains: []  
        ID: '0'  
    InputRanges: [-1 1]  
    MaxSampleRate: 44100  
    MinSampleRate: 8000  
NativeDataType: 'int16'  
    Polarity: {'Bipolar'}  
    SampleType: 'SimultaneousSample'  
SingleEndedIDs: [1 2]  
    SubsystemType: 'AnalogInput'  
    TotalChannels: 2  
VendorDriverDescription: 'Windows Multimedia Driver'  
VendorDriverVersion: '5.10'
```

由上面信息可以看出,使用的这款声卡(c-Medlia Wave Device)的输入设备对象所支持的最低采样频率是 8 kHz,最高采样频率是 44.1 kHz,转换位数是 16 位,输入范围为-1~1V,双极性,两个通道。

1.1.2 数据采集工具箱函数

本节按照功能归类列出数据采集工具箱的所有函数,以便读者参考。设备对象分为以下三种类型。

- AI: analog input, 模拟输入;
- AO: analog output, 模拟输出;
- DIO: digital I/O, 数字量输入/输出。



1) 创建设备对象的函数

创建设备对象的函数见表 1-1。

表 1-1 创建设备对象的函数

函数名	函数功能描述	AI	AO	DIO
analoginput	创建一个模拟量输入设备对象	√		
analogoutput	创建一个模拟量输出设备对象		√	
digitalio	创建一个数字量输入/输出对象			√

2) 增加通道或线路的函数

增加通道或线路的函数见表 1-2。

表 1-2 增加通道或线路的函数

函数名	函数功能描述	AI	AO	DIO
addchannel	增加通道到模拟量输入/输出设备对象	√	√	
addline	增加线路到数字量输入/输出设备对象			√
addmuxchannel	增加通道到多路复用通道（仅 NI 的采集板使用）	√		

3) 读取或设置属性值的函数

读取或设置属性值的函数见表 1-3。

表 1-3 读取或设置属性值的函数

函数名	函数功能描述	AI	AO	DIO
get	读取设备对象属性	√	√	√
set	设置或显示设备对象属性	√	√	√
setverify	设置并返回设备对象属性	√	√	√

4) 执行函数

执行函数见表 1-4。

表 1-4 执行函数

函数名	函数功能描述	AI	AO	DIO
start	启动设备对象	√	√	√
stop	停止设备对象	√	√	√
trigger	人为地产生一个触发信号	√	√	
waittilstop	等待，直到设备对象停止运行	√	√	

5) 数据函数

数据函数见表 1-5。

表 1-5 数据函数

函数名	函数功能描述	AI	AO	DIO
flushdata	清除数据采集引擎中的数据	√		
getdata	从数据引擎中读取数据、时间和事件信息	√		
getsample	立即采样一次	√		
getvalue	从线路上读取数据		√	



续表

函 数 名	函数功能描述	AI	AO	DIO
peekdata	读取最近采集的数据	√		
putdata	输出数据		√	
putsample	立即输出一个数据		√	
putvalue	写数据到线路上			√

6) 帮助函数

帮助函数见表 1-6。

表 1-6 帮助函数

函 数 名	函数功能描述	AI	AO	DIO
daqhelp	显示帮助信息（包括设备对象、函数、接口和各种属性）	√	√	√
daqhwinfo	显示数据采集硬件信息	√	√	√
daqpropedit	调用图形交互界面的属性编辑器	√	√	√
propinfo	读取设备对象、通道和线路参数	√	√	√

7) 通用函数

通用函数见表 1-7。

表 1-7 通用函数

函 数 名	函数功能描述	AI	AO	DIO
binvec2dec	将二进制向量转换为十进制数			√
clear	从 MATLAB 工作空间中清除设备对象	√	√	√
daqcallback	显示指定事件的回调函数	√	√	√
daqfind	从数据采集引擎读取设备对象、通道和线路	√	√	√
daqmem	开辟或显示内存资源	√	√	
daqread	读取一个数据采集工具箱文件 (*.daq)	√		
daqregister	注册或取消一个硬件驱动接口	√	√	√
daqreset	从内存中清除数据采集设备对象和动态链接库	√	√	√
dec2binvec	将十进制数转换为二进制向量			√
deletd	从数据采集引擎中清除设备对象、通道或线路	√	√	√
disp	显示对象、通道或线路的摘要信息	√	√	√
Ischannel	检查是否为通道	√	√	√
Isdioline	检查是否为线路	√	√	√
IsValid	检查对象、通道或线路是否有效	√	√	√
length	读取对象通道组或线路组的长度	√	√	√
load	加载对象、通道和线路到 MATLAB 工作空间	√	√	√
makenames	生成通道或线路的描述名称	√	√	√
muxchanidx	读取多路复用通道的扫描指数（仅 NI 的采集板使用）	√		
obj2mfile	将对象通道或线路转换为 MATLAB 代码	√	√	√
save	保存设备对象为 MAT 文件	√	√	√
showdaqevents	显示事件日志信息	√	√	
size	读取设备对象、通道组或线路组的大小	√	√	√



1.1.3 数据采集输入设备的对象属性

数据采集输入设备的对象属性可以分为公共属性和通道属性两类，公共属性作用于所有通道，而通道属性只作用于当前通道。

1. 公共属性

根据用途不同，公共属性又可分为以下几类。

1) 基本配置属性

- **SamplesPerTrigger**: 每次触发的采样数目;
- **SampleRate**: 采样频率;
- **TriggerType**: 触发方式。

2) 存放属性

- **LogFileName**: 存放信息的文件名;
- **Logging**: 数据是否保存内存或文件;
- **LoggingMode**: 采集数据的保存模式;
- **LogToDiskMode**: 数据、事件和硬件信息的保存模式。

3) 触发属性

- **InitialTriggerTime**: 首次触发的绝对时间;
- **ManualTriggerHwon**: 手动触发启动设备;
- **TriggerFcn**: 触发时的回调函数;
- **TriggerChannel**: 触发通道;
- **TriggerConditlon**: 触发条件;
- **TriggerConditionValue**: 触发条件值;
- **TriggerDelay**: 触发到保存数据间的延时;
- **TriggerDelay Units**: 触发延时单位;
- **TriggerRepeat**: 触发的重复次数;
- **TriggersExecuted**: 已经执行了的触发次数;
- **TriggerType**: 触发方式。

4) 状态属性

- **Logging**: 正在存放数据;
- **Running**: 设备正在运行;
- **SamplesAcquired**: 需要的采样点数;
- **SamplesAvailable**: 引擎中的有效采样点数。

5) 回调属性

- **DataMissedFcn**: 丢失数据时回调函数;
- **InputOverRangeFcn**: 输入超过设定范围时回调函数;



- **RuntimeErrorFcn**: 运行时错误函数;
- **StartFcn**: 启动时回调函数;
- **StopFcn**: 停止时回调函数;
- **TimerFcn**: 定时器回调函数;
- **TimerPeriod**: 定时器周期;
- **TriggerFcn**: 触发时回调函数。

6) 通用属性

- **BufferingConfig**: 为每个通道开辟的缓冲区（内存）大小;
- **BufferingMode**: 开辟缓冲区的方式;
- **Channel**: 对象的通道;
- **EventLog**: 对象事件日志;
- **Name**: 对象的描述名称;
- **Tag**: 对象的身份标签 (ID);
- **Timeout**: 设置读取数据的超时;
- **Type**: 对象类型;
- **UserData**: 与对象关联的数据。

2. 通道属性

- **ChannelName**: 通道的描述名称;
- **HwChannel**: 对应的硬件通道号;
- **Index**: MATLAB 中的指数;
- **InputRange**: 输入范围;
- **NativeOffset**: 原始数据与双精度型 (double) 数据互换时的偏移量;
- **NativeScaling**: 原始数据与双精度型 (double) 数据互换时的比例因子;
- **Parent**: 父对象 (设备对象);
- **SensorRange**: 传感器的输入范围;
- **Type**: 通道类型;
- **Units**: 工程单位;
- **UnitsRange**: 工程单位的范围。

1.1.4 MATLAB 数据采集步骤

在 MATLAB 中提供了数据采集工具箱, 调用该工具箱使 MATLAB 可以控制 NI 的 DAQ 采集卡, 完成采集与分析任务。MATLAB 通过 NI-DAQ 控制 NI 的采集卡, 在编程、调试与使用前, 应正确安装采集卡的驱动程序, 并在 NI-MAX 中对采集卡做好设置, 使其能够正常工作。

1. 通过 MATLAB 编程实现 MATLAB 与 NI 采集卡的连接

MATLAB 将 NI 采集卡等设备都作为对象处理, 其后的一切操作与硬件不直接相关, 而是通过对该对象的操作来作用于硬件设备的。而对于该系统来说, 用 NI 采集卡采集数据, 实际



上输入/输出的都是模拟信号，因此在 MATLAB 中，只要使用相应的语句就可以很容易地实现 MATLAB 与 NI 采集卡的连接。

MATLAB 中有 `analoginput` 语句，专门用于建立模拟信号输入，其语法为：

```
AI=analoginput('adaptor',ID)
```

其中，`adaptor` 是硬件设备的对应名称，MATLAB 所支持的硬件设备包括 `advantech`、`hpel432`、`keithley`、`mcc`、`nidaq` 和 `winsound`。ID 则是硬件设备的标识，如果硬件设备是声卡，则 ID 标识可以不要。对于该系统，由于是用 NI 采集卡作为采集硬件设备，则用以下语句：

```
AI=analoginput('nidaqr',1)
```

就可以使 MATLAB 在众多的 PC 硬件中找到 NI 采集卡，并创建一个模拟数据的采集目标。此处 ID 在不同配置的机器下可能不同，可自行设定。

当一个输入目标被创建之后，它不可能占用所有的硬件信号通道。为了操控硬件设备，必须创建一条专用的通道用于模拟信号数据的传输。这就要用到 MATLAB 中的 `addchannel` 语句。其语法一般为：

```
chan=addchannel(obj,hwch)
```

其中，`obj` 是模拟输入或输出的目标，在这里是上述的“AI”；`hwch` 是指定设备的 ID 数量。对于该系统，指定 `hwch` 为 1。所以创建数据传输通道的语句如下：

```
chan=addchannel(AI,1)
```

2. 设置采样频率和采样时间以及采样点数

对于设定采样时间、频率和采样点数，可以由以下语句完成：

```
duration=2;  
Fs=8000;  
Blocksize=duration*Fs;
```

考虑到以后在设计图形用户界面的时候可以在界面上对它们进行设置，所以可利用 MATLAB 中的 `set` 语句对它们进行赋值。其语法格式为：

```
set(object,'property',value);
```

在该系统中，`object` 为 AI，`property` 为 `SampleRate`，而 `value` 取为 8000，即

```
set(AI,'SampleRate',8000);
```

当要设置采样点数时，`object` 为 AI，`property` 为 `SamplesPer Trigger`，而 `value` 为 `duration*ActualRate`，即

```
set(AI,'SamplesPer Trigger',duration*ActualRate);
```

而在此之后则可以使用 `get` 语句将设定好的值传送给变量，其基本语法格式为：

```
get(ob,'PropertyName')
```

其中，`ob` 为目标名称，在这里为 AI，`PropertyName` 为数值来源名称，在这里为 `SampleRate` 和 `SamplesPer Trigger`，即

```
ActualRate=get(AI,'SampleRate');  
blocksize=get(AI,'SamplesPer TfiggeI');
```

所以该部分的全部语句为：