

全日制普通本科生毕业设计

基于 MATLAB 实时串口数据采集与曲线显示

REAL-TIME SERIAL DATA ACQUISITION AND FIGURE SHOW

BASED ON MATLAB

学生姓名：

学 号：

年级专业及班级：

指导老师及职称：

学 院：

提交日期：2011 年 5 月

全日制普通本科生毕业论文（设计）

诚信声明

本人郑重声明：所呈交的本科毕业论文（设计）是本人在指导老师的指导下，进行研究工作所取得的成果，成果不存在知识产权争议。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体在文中均作了明确的说明并表示了谢意。本人完全意识到本声明的法律结果由本人承担。

毕业论文（设计）作者签名：

年 月 日

目 录

摘要.....	1
关键词...	1
1 前言.....	2
1.1 Matlab 实时串口数据采集研究现状及发展趋势.....	2
1.2 研究的目的和意义.....	4
1.3 论文的组织结构.....	5
2 Matlab 下实时串口数据采集概要.....	5
2.1 Matlab 的 Serial 类.....	5
2.2 数据采集...	6
2.3 曲线显示	7
3 实时串口数据采集与曲线显示的实现.....	8
3.1 实时串口通信的实现.....	8
3.2 数据采集的实现.....	9
3.3 曲线显示 GUI 的实现.....	10
4 基于 MATLAB 的实时串口数据采集与曲线显示的具体做法.....	12
4.1 数据采集的一般流程	12
4.1.1 创建接口对象并设置属性.....	12
4.1.2 打开串口设备对象.....	12
4.1.3 读写串口操作.....	13
4.1.4 关闭并清除设备对象.....	13
4.2 基于 Matlab 中断方式的实时串行通信编程.....	13
4.3 绘制采集数据的曲线波形和数据显示.....	14
4.3.1 绘制曲线波形.....	14

4.3.2	数据显示.....	15
4.3.3	采集图像.....	... 15
4.4	扩展功能——发送数据.....	17
5	结论.....	17
	参考文献	18
	致 谢	19
	附 录	19
	附 录 1	20

基于 MATLAB 实时串口数据采集与曲线显示

摘 要：数据采集是获取信息的基本手段，数据采集技术作为信息科学的一个重要分支，它研究信息数据的采集、存储、处理及控制等作业，具有很强的实用性，与传感器、信号测量与处理、微型计算机等技术为基础而形成的一门综合应用技术。本设计是在 Matlab R2008a 版本中以串口通信实时接收目标系统数据，将采集的数据进行时间同步和字对齐处理，并在 MATLAB 的数据采集工具箱支持下，利用图形界面 GUI 的设计，直观的实时显示数据曲线。设计得重点在于在 Matlab 上实现串口数据同步采集与处理，接收数据的时间同步和字对齐，同时扩展了发送数据的功能。

关键词： MATLAB ； GUI ； 实时串口 ； 数据采集 ； 曲线显示 ；

Real-time Serial Data Acquisition and Figure Show Based on Matlab

Abstract: Data acquisition is the basic means of access to information, this technology as an important branch of information science, research information data collection, storage, treatment and control operations, and highly practical. Data acquisition technology with sensor data acquisition, signal measurement and processing, micro-computers and other technology formed the basis of a comprehensive application technology. This design version use Matlab R2008a serial communication to the target system to receive real-time data, the data will be collected for time synchronization and word alignments, and Matlab's Data Acquisition Toolbox in support of using the graphical user interface GUI design, intuitive display real-time data curve. Focus designed to achieve the Matlab serial data synchronization acquisition and processing, receive data synchronization and word alignment, expanding the function of sending data.

.Key words: MATLAB; GUI; Real-time serial; Data acquisition; Figure show;

1 前言

1.1 MATLAB 实时串口数据采集研究现状及发展趋势

随着科学技术的发展，数据采集系统得到了越来越广泛得应用，同时人们对数据采集系统的各项技术指标，如：采样率、线性度、精度、输入范围、控制方法以及抗干扰能力等提出了越来越高的要求，特别是精度和采样率更是使用者和设计者所共同

关注的重要问题，于是，高速及超高速数据采集系统应运而生并且得到了快速发展。今天，数据采集技术已经在雷达、水声、振动工程、无损监测、智能仪器、自动控制以及生物工程等众多领域得到广泛的应用并且收到了良好的效果。高速数据采集系统在国防、航天、边缘科学研究中及国民经济的各个领域的成功的应用，进一步引起了各方的关注，推动了它的研制和发展。随着科学技术的发展，数据采集系统得到了越来越广泛的应用。目前，国外很多公司与厂商都投入巨资进行数据采集系统的研制开发与生产销售，其中比较著名的有 NEFF，NI、HP，TEK 等。

从数据采集系统产品来看，各大公司提供的系列产品都包括了完成数据采集的诸如信号放大、滤波、多路开关、模数转换和接口等各种模块。现有的高速数据采集器件和开发的产品中，目前还没有完全实现高速、高分辨率。在雷达、通信、谱分析、瞬态分析、等应用领域，为满足实时检测和高速采集的日益更新的需要，实现数据采集的高速、高分辨率已成为数据采集系统的一个发展方向。现有的高速 ADC 器件和产品价格都比较昂贵，有些高速、高分辨率的器件本身还存在着不稳定性，因此，在数据采集系统向高速、高分辨率发展的同时，开发和研制的器件和产品应不断地提高可靠性，降低，提高性价比，以便使之得到更广泛的应用。

数据采集与分析一直是生产实践研究与应用领域的一个热点和难点。随着微制造工艺水平的飞速提高及数据分析理论的进一步完善与成熟，目前国内外对数据采集系统的高性能方面的研究上取得了很大的成就。就 A/D 转换的精度、速度和通道数来说，采样通道从单通道发展到双通道、多通道，采样频率、分辨率、精度逐步提高，为分析功能的加强提供了前提条件。而在数据分析的微处理器上，最初的数据采集系统以 8 位单片机为核心，随着微电子技术的不断发展，新兴单片机的不断问世，十六位、三十二位单片机也为数据采集系统研制厂家所采用，近年来采用具有 DSP 功能的数据采集系统也已投入市场。同时，通用 PC 机的 CPU 用于数据处理也较为常见。总之，伴随着高性能微处理器的采用和用户技术要求的不断提高，数据采集系统的功能也越来越完善。

数据采集系统的发展主要体现在以下几个趋势：

首先，在专业测控方面，基于 PC 计算机的数据采集系统越来越成熟和智能化。在过去的二十年中，开放式架构 PC 机的处理能力平均每十八个月就增强一倍。为了充分利用处理器速度的发展，现代开放式测量平台结合了高速总线接口，如 PCI 和 PXI/Compact PCI，以便获得性能的进一步提升。计算机的性能提升和由此引起的基于计算机的测量技术的创新，正在持续不断地模糊传统仪器和基于计算机的测量仪器之

间的界线。

其次，在通用测控方面，采用嵌入式微处理器的方案也由早期的采用 A/D 器件和标准单片机组成应用系统发展到在单芯片上实现完整的数据采集与分析，即目前极为热门的 SOC (System On Chip)。通常在一块芯片上会集成一个，可以采样多路模拟信号的 A/D 转换子系统和一个硬 CPU 核(比如增强型 80_52 内核)，而且其 CPU 的运算处理速度和性能也较早期的标准 CPU 内核提高了数倍，而且有着极低的功耗。这种单芯片解决方案降低了系统的成本和设计的复杂性。

此外，为了解决 SOC 方案中数据处理性能的不足，采用 DSP 作为数据采集系统的 CPU 的研究与应用目前也逐渐引起业内重视。但是这类产品目前仅仅处于发展的初级阶段，在精度、速度或其它性能指标上并不能很好的满足要求。因此，国内外以 DSP 作为数据采集系统的采样控制和分析运算的研究与应用正在展开。

近年来随着芯片技术、计算机技术和网络技术的发展，数据采集技术取得了许多新的技术成果，市场上推出了繁多的新产品。高速数据采集技术的发展一方面是提高采集速率，另一方面不断向两端延伸。一端是输入的信号调理，另一端是采集后的数字化信号的实时处理与事后处理。20 世纪 90 年代末，随着数字技术快速发展，数据采集技术已向着并行、高速、大量存储、实时分析处理、集成化等方向发展。

Matlab 是 Mathworks 公司推出的一套高性能数值计算和可视化软件，是目前控制系统数据处理较为实用有效的工具。它不仅能解决测试与控制系统中存在的大量的数值计算和矩阵运算，而且将图像与图形、显示及处理、图形界面设计集于一身。同时，它还提供了强有力的工具箱支持，极大地方便了研究人员的学习与开发。软件开发采用 Matlab 语言编程，利用 Matlab 中控制工具箱及仿真工具混合开发的方法，使其编写的程序更精练，软件功能更强大，开发周期更短，软件形式灵活、易于扩展，用户使用起来更容易、更方便。因此，许多工程技术人员把 Matlab 软件作为数据离线处理的工具。

业内领先的工具箱极大的扩展了 Matlab 的应用领域，所以 Matlab 自推出以来就受到广泛的关注，数据采集工具箱就是其中之一。它是为简化和加快数据采集工作而设计的，使用该工具箱更容易将实验测量得到的数据进行分析 and 可视化操作。利用该数据采集工具箱可以方便地建立数据采集系统。通过建立一个串口接口对象，可以使用 Matlab 命令直接和外部设备进行通信。

1.2 研究的目的是和意义

在近几十年来 IC 技术和技术的高速发展，为数据采集与分析提供了非常良好与

可靠的科学技术基础，也提出了更高的要求 and 强有力的推动。随着现代工业技术的迅猛发展，生产规模的不断壮大，生产过程和制作工艺的日趋复杂，对自动测试和各种信息集成的要求也就越来越高。在当今社会各个领域，包括科研和实验研究，数据采集系统有着不可替代的作用，数据采集和处理进行得越及时，工作效率就越高，取得的经济效益就越大。数据采集系统性能的好坏主要取决于它的精度和速度，在保证精度的条件下，还要尽可能地提高采样速度，以满足实时采集、实时处理和实时控制的要求。

数据采集系统的好坏将直接影响自动测试系统的可靠性和稳定性，为了满足不同的测试需求，以及减少对资源的浪费，在系统的设计上应该尽量满足通用性和可扩展性。在高度发展的当今中，科学技术的突飞猛进和生产过程的高度自动化已成为人所共知的必然趋势，而它们的共同要求是必须建立在有着不断发展与提高的信息工业基础上。人们只有从外界获取大量准确、可靠的信息经过一系列的科学分析、处理、加工与判断，进而认识和掌握自然界与科学技术中的各种现象与其相关的变化规律，并通过相应的系统和方法实现科学实验研究与生产过程的高度自动化。换言之，生产过程的自动化面临的第一个问题就是必须根据从各种传感器得到的数据来检测、监视现场，以保证现场设备的正常工作。所以对现场进行数据采集是重要的前期基础工作，然后再对现场数据进行传输和相应的处理工作，以满足不同的需要。

如果要对大量的实验数据利用 Matlab 的数据分析处理功能进行处理，则首先要将实验数据转换成 Matlab 的数据格式，这无疑是一件十分繁杂的任务。若能直接从 Matlab 环境下采集实验数据，无疑对与实验分析和数据处理都是有益的。Matlab (Version 7.0) 的数据采集箱 DAQ (Data Acquisition Toolbox) 为此提供一个实现直接数据采集的平台。利用该工具箱配以适当的数据采集卡不仅可以进行实时数据采集，而且还可以进行实时显示和控制，由此还可以组建成为自动测试分析系统。利用该数据采集工具箱可以方便地建立数据采集系统。

基于 MATLAB 实时串口数据采集，将采集的数据进行时间同步和字对齐处理，并在 Matlab 的工具箱支持下实时显示数据曲线，具有重要的意义。

1.3 论文的组织结构

摘要。阐述了此次设计的总思路，利用 MATLAB 接收目标系统串口通信实时数据，将采集的数据进行时间同步和字对齐处理，并在 MATLAB 的工具箱支持下实时显示数据曲线。

1. 前言。介绍了 MATLAB 实时串口数据采集的研究现状及发展趋势、研究的目的

和意义，说明了此设计的实用价值。

2. MATLAB 实时串口数据采集概要。对此设计中的理论知识进行概述，是设计过程中所需要的理论和条件支撑，以便设计的顺利进行。

3. 实时串口数据的采集与曲线显示的实现。大致描述了该设计的设计与实现方法，是设计过程中所需要的实践环节。

4. 基于 MATLAB 的实时串口数据采集与曲线显示的具体做法。详细的给出了完成此次设计的思路和具体做法。

有了正确的过程，便会得出结论，结论部分即是对此次设计结果的一个说明。本文总的来说是按照从抽象的概念描述到具体实现方法步骤的写作思路来完成的。

2 Matlab 实时串口数据采集概要

2.1 Matlab 的 serial 类

串行接口 (Serial Interface) 简称串口，也称串行通信接口 (通常指 COM 接口)，是采用串行通信方式的扩展接口。串口的出现是在 1980 年前后，数据传输率是 115kbps ~ 230kbps。串口出现的初期是为了实现连接计算机外设的目的，初期串口一般用来连接鼠标和外置 Modem 及老式摄像头和写字板等设备。串口也可以应用于由于两台计算机 (或设备) 之间的互联及数据传输。

串行接口是指数据一位一位地顺序传送，其特点是通信线路简单，只要一对传输线就可以实现双向通信，并可以利用电话线，从而大大降低了成本，特别适用于远距离通信，但传送速度较慢。由于串口 (COM) 不支持热插拔及传输速率较低目前部分新主板和大部分便携电脑已开始取消该接口，目前串口多用于工控和测量设备以及部分通信设备中。

自 Matlab6.0 版本起，Mathworks 公司在软件中增加了 Instrument Control Toolbox，提供了对串口通信的正式支持。利用该工具箱的 serial 及其相关函数，能可靠地进行串行通信。

仪器控制工具箱 serial 类的主要特点如下：

- (1)支持基于 GPIB 总线 (IEEE-488、HPIB 标准)、VISA 总线、串行接口 (RS-232、RS-422、RS-485)的通信标准；
- (2)支持二进制和文本 (ASCII) 两种数据的通信，文本方式支持 (standard commands for programmable instruments)语言；
- (3)支持异步通信和同步通信；
- (4)支持基于事件驱动的通信。

2.2 数据采集

数据 (Data) 也称观测值 , 是实验、测量、观察、等的结果 , 常以数量的形式给出。数据采集 (Data Acquisition), 又称数据获取 , 就是将被测对象 (外部世界、现场) 的各种参量 (可以是量 , 也可以是化学量、生物量等) 通过各种传感元件作适当转换后 , 再经信号调理、采样、量化、编码、传输等步骤 , 最后把系统需要的所有对象的原始数据收集、归类、整理、录入到系统当中去 , 是管理系统使用前的一个数据初始化过程。数据采集技术广泛应用在信号检测、信号处理、仪器仪表等领域 , 比如摄像头 , 麦克风 , 都是数据采集工具 , 随着数字化技术的不断发展 , 数据采集技术也呈现出速度更高、通道更多、数据量更大的发展态势。

被采集数据是已被转换为电讯号的各种物理量 , 如温度、水位、风速、压力等 , 可以是模拟量 , 也可以是数字量。采集一般是采样方式 , 即隔一定的时间 (称采样周期) 对同一点数据重复采集。采集的数据大多是瞬时值 , 也可是某段时间内的一个特征值。准确的数据测量是数据采集的基础。数据测量方法有接触式和非接触式 , 检测元件多种多样。不论哪种方法和元件 , 都以不影响被测对象状态和测量为前提 , 以保证数据的正确性。数据采集含义很广 , 包括对连续物理量的采集。在计算机辅助制图、测图、设计中 , 对图形或图像数字化过程也可称为数据采集 , 此时被采集的是几何量数据。

数据采集系统是一种应用极为广泛的模拟量测量设备 , 其基本任务是把信号送入计算机或相应的信号处理系统 , 根据不同的需要进行相应的计算和处理。它将模拟量采集、转换成数字量后 , 再经过计算机处理得出所需的数据。同时 , 还可以用计算机将得到的数据进行储存、显示和打印 , 以实现某些物理量的监视 , 其中一部分数据还将被用作生产过程中的反馈控制量。

在数据采集中 , 一般将待测对象的信号通过传感器和 A / D 转换器或其它专业测试仪器 (如 TDS3000 系列数字式示波器) , 把数据传到计算机。编制计算机与测试仪器的接口程序可使用 C++ Builder、VB 等编程语言。数据采集到计算机后要进行各种处理。如 FFT、各种方法的滤波、系统辨识、小波变换等 , 由于 Matlab 提供功能强大的各种计算 , 使用几个简单语句就能实现 , 因此采用 Matlab 编程语言非常适合。

Matlab 自带的数据采集工具箱 DAQ 能更容易地将实验测得的数据进行分析和可视化操作 , 包括了三个模块 , 其中 M-File 函数是 Matlab 中可直接调用的数据采集命令函数 , 通过 M-file 函数传输数据、参数和命令给数据采集引擎 , 并可从数据采集引擎中查看数据采集设备对象的状态、参数和抽取数据 , 所有的实时数据采集任务都通

过这些函数的调用来实现。

数据采集引擎 (Data Acquisition Engine) 是 Matlab 环境下进行实时数据采集的核心。它执行 M-file 函数所规定的功能，并监视和控制数据采集设备对象的状态，以及存储和管理所采集的数据。一旦实时数据采集任务启动，它就类似于 DOS 环境下后台运行的监控程序，所有的数据采集任务就置于它的监控和管理之中。

硬件转接驱动程序 (Hardware Driver Adaptors) 建立采集卡驱动软件 (driver) 与数据采集引擎之间的数据、命令和参数的传输通道，对于不同的板卡，需要不同的驱动程序，可以是动态链接库文件 (*.DLL)，也可以是 Matlab 下的共享库文件 (*.MEX)。

它提供了以下主要的功能：

1. 以相同的指令操作模拟输入 (AI)、模拟输出 (AO)、数字输入输出 (DIO) 和同步模拟输入输出转换各式功能，不会因不同硬件而产生不同的指令。
2. 一个及时数据采集环境，被测量的数据不需经由转换可直接进入 MATLAB 直接进行分析。
3. 支持 PC 声卡和主要的数据采集卡的厂商，如：凌华，Agilent, ComputerBoards 和 National Instruments。
4. 支持事件驱动 (Event-driven) 的数据采集。

2.3 曲线显示

Matlab 具有强大的绘图功能，在 Matlab 中将采集到的数据实时显示出来，可以使用二维曲线绘图的方法。

Plot 函数是 Matlab 中最常见的绘图函数，可以用来绘制单条或多条曲线，它是针对向量或矩阵的列来绘制曲线的。也就是说使用 plot 函数之前，必须首先定义好曲线上每一点的 X 及 Y 坐标，常用的格式有：

`plot(x,y)`

以向量 x、y 为轴，绘制曲线。其中 x 和 y 为坐标向量。

`plot(x1,y1,x2,y2...)`

绘制多条曲线 (也可以用矩阵变量 `plot(x,y)`)

`plot(x,y1,x,y2,x,y3...)`

其功能是以公共向量 x 为 X 轴，分别以 y1,y2,y3...为 Y 轴，在同一幅图内绘制出多条曲线。

`plot(x1,y1,LineStyle1,x2,y2,LineStyle2...)`

将按顺序分别画出由三参数 Xi,Yi,LineStylei 定义的线条。其中参数 LineSpeci 指明

了线条的类型，标记符号，和画线用的颜色。

```
xlabel ( ' frequency (Hz); ' )           %X 轴名称为 frequency(Hz)
ylabel ( ' magnitude(dB) ' ;)           %Y 轴名称为 magnitude(dB)
```

3 实时串口数据采集与曲线显示的设计方法

3.1 实时串口通信的实现

Matlab7.0 支持面向对象技术，用一个对象把计算机串口封装起来。创建串口对象后，对其的操作就是对串口操作，从而实现了计算机串口操作的简化。因此，使用 Matlab7.0 平台，通过计算机串口实现对专业测试仪器的控制以及测试数据的实时采集、传输、处理和结果显示。是十分有效的实时数据采集与处理方法之一。

Matlab7.0 封装的串口对象支持对串口的异步读写操作，计算机在读写串口时能同时进行其他处理工作，因而使计算机具有极高的执行效率。Matlab7.0 用多线程技术实现这种异步操作，通过异步读写设置，计算机在执行读写串口函数时能立即退回，不必等待串口把数据传输完毕。当指定的数据传输结束时就触发事件，执行事件回调函数。可以在事件回调函数中编程，进行数据处理，这样就不会造成因等待串口传输数据引起的机时浪费。

Matlab 串行口对象属性。使用 serial 函数创建串口对象后，通过对象句柄操作串口，非常方便。串口对象有很多属性，使用如下的命令可以查看串口所有属性

```
set(s) %s 是 serial 创建的 COM1 串口对象的标识符
```

通过定义串口对象的属性，就能定义串口的通信模式，也可以了解串口的状态。

几个比较重要的属性如下：

1. 串口名属性

Port 定义系统的串口名称（一般为 ' COM1 到 COM 7 '）

2. 串口通信属性

BaudRate 定义传输波特率（缺省 9600bps，可设为 1200，2400，4800，19200bps 等等）

DataBits 定义数据帧的比特位数（一般为 7 或 8 位）

Parity 定义校验方式（可取值：none、odd、even、mark、space）

StopBits 定义数据帧停止位位数（一般为 1、1.5 或 2）

Terminator 定义结束字符（一般为 ' LF '）

3. 串口读数据方式

ReadAsyncMode 定义异步读操作为连续或手工方式（continuous-连续方式（缺省），或定义为 manual）

3.2 数据采集的实现

数据采集工具箱集成于 MATLAB 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，或者存储后再进行处理，或者针对数据分析的需要对测试条件的设立进行不断的更新。应用数据采集工具箱提供的命令和函数可以控制任何类型的数据采集。DAQ 是由一组 M-File 文件格式的函数和 MEX-File 等动态链接库所组成的，专门使用在数据采集方面的一组函数库。如图 1 所示，这些组成部分使得 Matlab 与数据采集硬件之间的信息传递成为可能。

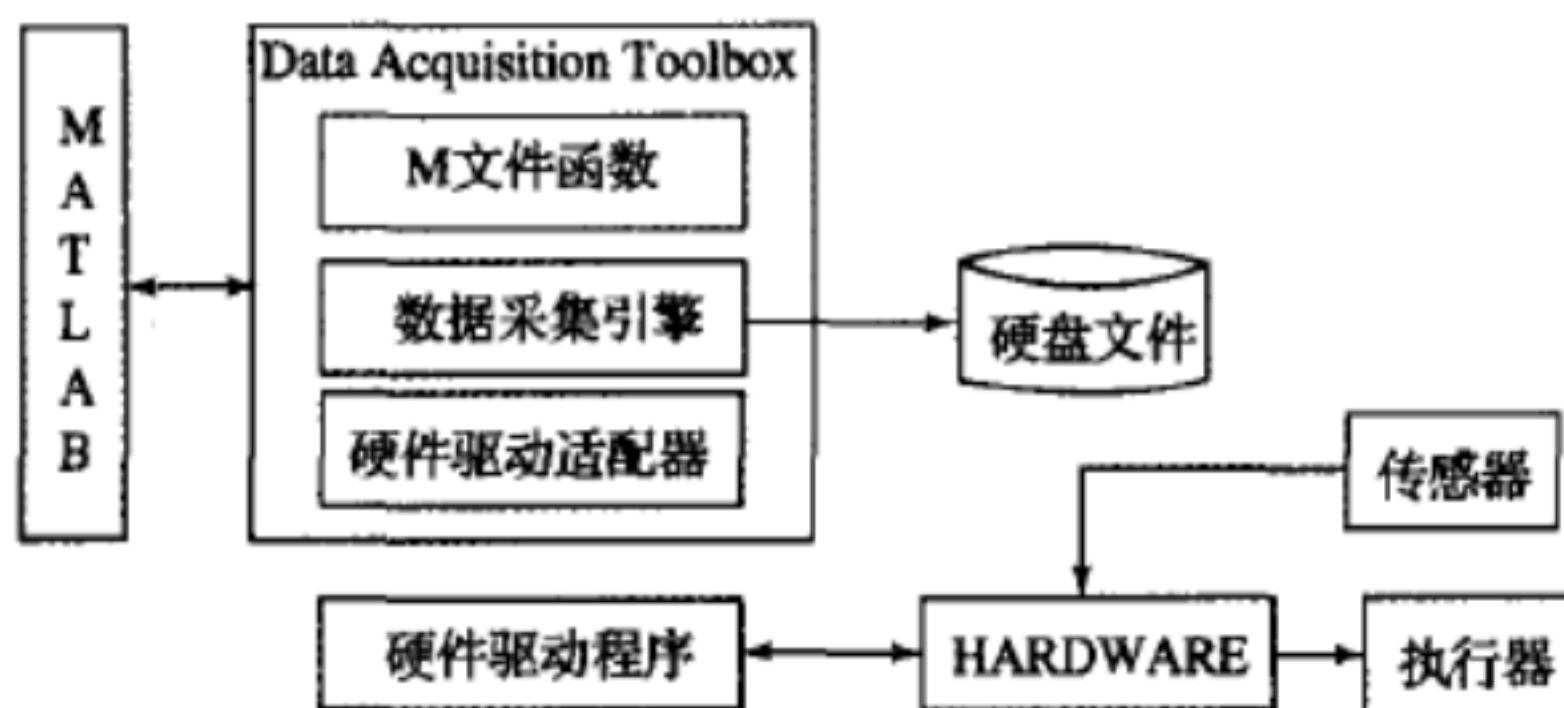


图 1: 数据采集引擎与硬件驱动

Fig 1 : Data Acquisition Engine and Hardware Driver

将被测物理量通过传感器连接到数据采集卡，通过采集函数命令启动数据采集任务，在数据采集引擎管理下，数据将从采集卡流入数据引擎，临时存储在内存中，以便于数据的重写，数据重写的速度依赖于内存的大小、数据采样的速率和采集通道数。存储在内存中被数据采集引擎管理的数据并不能自动进入 Matlab 工作空间，为了在 Matlab 工作空间使用这些数据，必须使用 get-data 函数从引擎中抽取。被抽取到 Matlab 空间的数据，可以用 Matlab 中其他工具箱的函数进行进一步的分析和处理，也可以直接写入磁盘文件。

利用数据采集工具箱，在 Matlab 环境下实时数据采集的步骤如图 2 所示。首先利用命令函数 analoginput 创建设备对象，并将该设备对象赋予一个 Matlab 变量，以后对该设备对象的操作就转换为对该变量操作。当设备对象建立后，即可用 ad-dchannel 命

令给该设备按照需要添加数据采集通道。一旦采集通道添加完成，就可以用 `set` 命令为设备对象设置采集参数，不同的采集卡，具有不同的采集特性参数。当采集参数设定后，即可用 `start` 命令启动数据采集，这时采集卡的状态就置于数据采集引擎的管理中，数据采集引擎会自动按照设定的参数监管和响应采集卡，并将其所采集的数据输入数据引擎管理的内存区，通过相应的命令可以从 Matlab 中查看数据采集任务的状态和抽取数据到 Matlab 工作空间，并可对抽取到 Matlab 工作空间的数据进行处理分析。当采集到数据满足设定参数时，采集任务即告结束。这时必须利用 `delete` 和 `clear` 命令清除设备对象，否则，该设备将会一直处于采集引擎的监管之中，再次创建该设备对象时，系统将会报告一个错误。

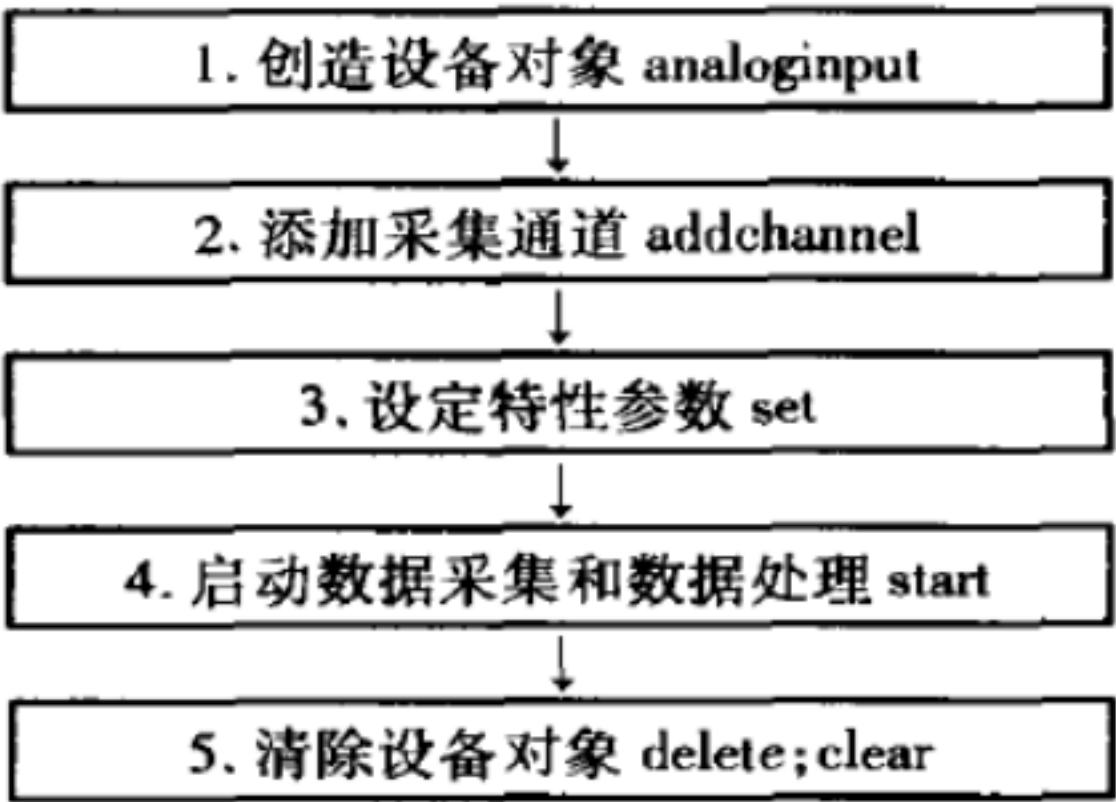


图 2：数据采集命令流程图

Fig 2: Command Flow Chart of Data Acquisition

3.3 曲线显示 GUI的实现

利用 MATLAB 软件中 GUI 模块可以方便地设计出基于对话框的图形用户界面，它提供了诸如编辑框、按钮、滚动条等图形对象，通过对这些图形对象的有机组合，再对相应的图形对象编写程序，就可以设计出界面友好、操作方便的系统软件。在 Matlab 的命令窗口输入：`guide`，就可以进入图形用户界面的设计向导，在向导控制面板中，鼠标单击左边用户界面控制对象面板中所需的按钮对象，分别把它们放在图形窗口中，并且可以通过拖动来改变其位置和大小。然后双击各个按钮对象，就可以打开其属性编辑器，拖动属性编辑器右边的滑块，改变各个按钮对象属性，这样就可以设计出对话框的外观。设计效果图如图 3 所示。

MALTAB 的 GUI 主要有两个文件，当设置好 GUIDE 时点击保存，这时将在当前路径自动产生两个名字相同的文件，但扩展标识符不同，一个为 `.fig` 文件，一个为 `.m`

文件，.fig 主要是 GUI 图形文件，和 GUIDE 设计的图像相同，.m 文件为程序文件，所有事件响应和计算功能都在这里实现。

MATLAB 在创建每一个图形对象时，都为该对象分配唯一的一个值，称其为图形对象句柄（Handle）。句柄是图形对象的唯一标识符。在进行对话框设计时，要让对话框实现输入输出功能，就要通过其句柄值调用一些函数文件，而当句柄未知时，可以用标识标签‘Tag’属性和函数 findobj 确定。其方法是通过给期望的图形对象提供唯一的‘Tag’属性字符串，用 findobj 函数来查找带了期望的标识标签的句柄。

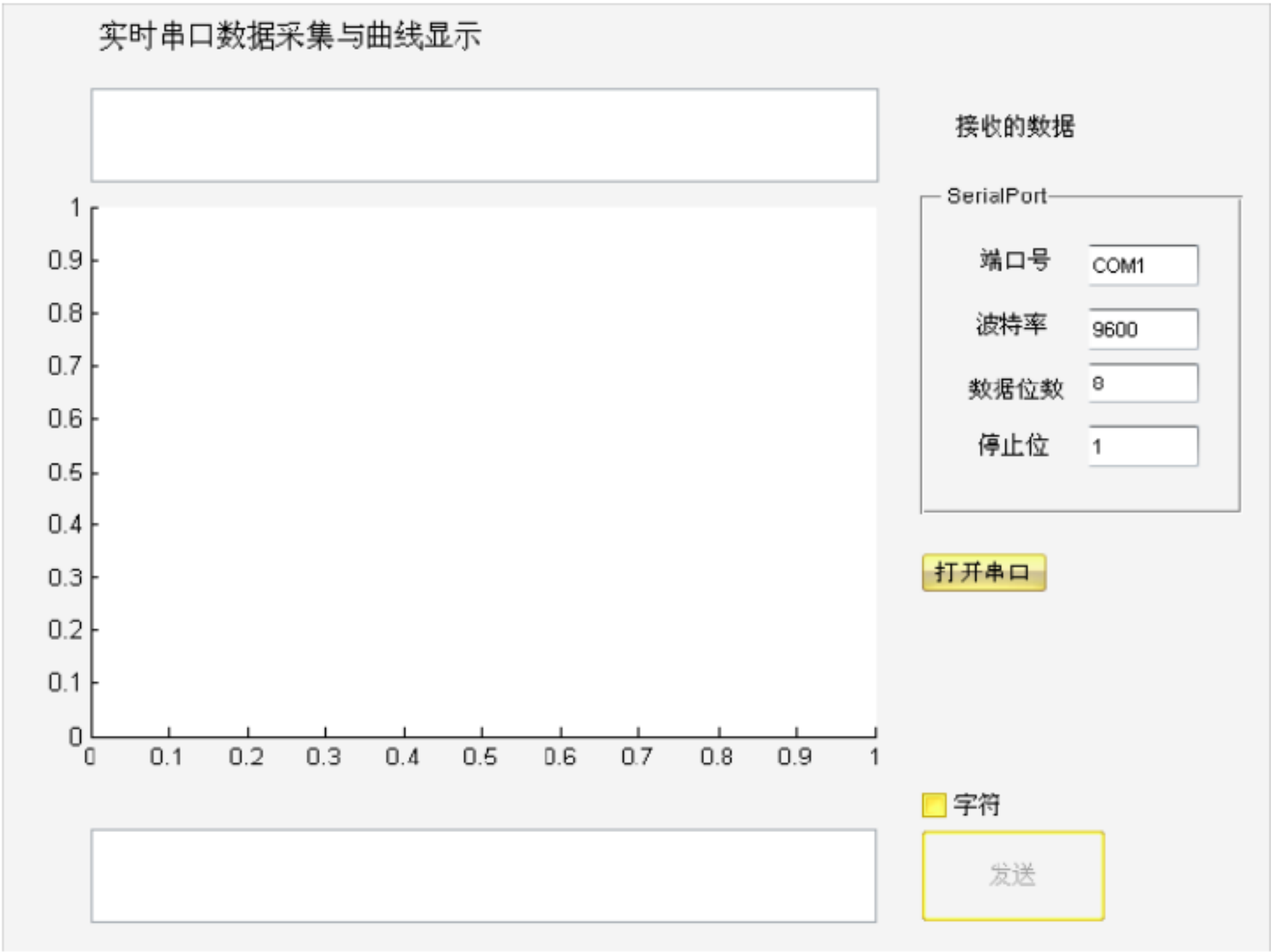


图 3：GUI 界面

Fig 3:GUI Interface

MATLAB 的按钮以及各种控件事件响应是依靠一种叫做回调函数 (Callback Function)来实现的。GUI 可以包括两个方面：控件和回调函数 (use interface components & callbacks)；回调函数间数据的传递，也就是程序结构的问题。 Matlab 提供了两种方法构造控件：用 uicontrol 系列的函数，或者用 guide 直接添加。控件有回调函数 callback 属性，这实际上是一个字符串，包括函数名和参数，它应该是可以在 workspace 里面执行的表达式或者 M 文件名。

考虑 M 文件形式的回调函数，因为它需要能够在 workspace 中执行，那么它的参

数就必须在 workspace 中存在，它执行的结果也在 workspace 中。这样所有需要的数据都必须是全局的，甚至比全局还要糟糕，它可能会与 workspace 中其他程序的数据冲突。matlab 提供了一个巧妙的途径：把数据跟图形 (.fig 文件) 联系起来。fig 上的控件，是以句柄(handles)的形式存在的，handles=guihandles(fig)可以得到 fig 的所有句柄。matlab 是把图形的数据也作为 fig 的句柄保存起来。

4 基于 MATLAB 的实时串口数据采集与曲线显示的具体做法

4.1 数据采集的一般流程

4.1.1 创建接口对象并设置属性

```
handles.S=serial(com); %创建串口对象  
set(handles.S, BaudRate, 9600, StopBits, 1); %设置波特率缺省 9600bit/s  
和停止位
```

```
set(handles.S, Parity, none); %设置无奇偶校验  
set(handles.S, FlowControl, none); %无流控制  
set(handles.S, DataBits, 8); %设置 8 位数据位  
set(handles.S, InputBufferSize, receivelen*10); %设置缓冲区大小  
设置串口设置对象的属性也可以用一条指令完成，如：
```

```
set(handles.S,'COM3','BaudRate',3600,'Parity','none','DataBits',8,'StopBits',1)
```

创建了对象后可以在 Matlab 命令窗口直接敲对象名并回车，看到其基本属性和当前状态，若需要知道其全部的属性，可以用 get(s)命令。

有些属性只有在对象没有被打开时才能改变其值，如 InputBufferSize、OutputBufferSize 属性等。对于一个 RS-232/RS-422/RS-485串口设备对象，其属性的缺省值为波特率 9600b/s，异步方式，通信数据格式为 8 位数据位，无奇偶校验位，1 位停止位。如果要设置的串口设置对象的属性值与缺省值的属性值相同，可以不用另行设置。

4.1.2 打开串口设备对象

```
fopen(handles.s);
```

4.1.3 读写串口操作

初始化并打开串口调协对象之后，现在可以对串口设备对象进行读写操作，串口的读写操作支持二进制和文本 (ASCII) 两种方式。当 Matlab 通信数据采用西方 (ASCII) 方式时，读写串口设备的命令分别是 fscanf、fprintf；当 Matlab 通信数据采用二进制方式时，读写串口设备的命令分别是 fread、fwrite。下面以文本方式读写串口为例：

读串口。 `A=fscanf(s,'%d',[10,100]);` %从串口设备对象 `s` 中读入 10×100 个数据填充到数组 `A[10,100]` 中，并以整型的数据格式存放。

写串口。 `Fprintf(s,'%s','RS232','async');` %将字符串 ‘RS232’ 以字符的数据格式写入到串口设备 `s`，写操作以异步的方式进行。

4.1.4 关闭并清除设备对象

`fclose(handles.s);` %关闭串口设备对象

`delete(handles.s);` %删除内存中的串口设备对象

`clear handles.s;` %清除工作空间中的串口设备对象

当不再使用该串口设备对象时，顺序使用以上 3 条命令，可以将所创建的串口对象清除，以免占用系统资源。

4.2 基于 Matlab 中断方式的实时串行通信编程

在 Matlab 环境下，读取串口数据的方式可以分为两种——查询和中断。以查询的方式进行串行通信时，如下位单片机有大量的数据分时分批传送给 PC 机，就需要不停查询串口的缓冲区，有数据就读取；虽然编程容易，但这样做不能对数据进行实时处理，系统实时性不高，而且会极大地占用系统的资源。以中断的方式对串口进行控制实现串行通信，就可以实时处理下位机传送的数据；但编程相对复杂一些，需要采用 Matlab 的事件和回调函数机制。

在 Matlab 环境下以中断的方式进行串行通信，实际上是采用事件驱动的方法实现的。Matlab 提供了 `instrcallback(obj, event)` 回调函数，用户根据需要可以自行设置具体的串行通信事件。Matlab 常用的串行口通信中断事件有：缓冲区有指定字节数目的数据可用事件（`bytes-available event`）、串口接收到的数据长时间处于非激活状态事件（`break-interrupt event`）、串行口引脚状态改变事件（`pin-status event`）、输出缓冲区为空事件（`output empty event`）等。当串口上有监视的事件发生时，Matlab 会自动调用回调函数进行通信事件的处理。因此，事件驱动实质上是一种中断机制，而回调函数实质上相当于一个中断服务子程序。Matlab 端实时串行通信的程序流程如图 4 所示。以下是具体的编程步骤。

1. 建立一个串行通信主程序：`SerialPortCommunication.m` 文件，在主程序中进行串口设备初始化操作，并指定回调函数中串行通信的事件。

程序主要源代码如下（创建串口设备对象、设备串口设备属性及打开串口等初始化操作的代码与前述的查询方式下的初始化代码相同）：

%设置串口事件回调

```
o_SerialPort.BytesAvailableFcnMode='byte' %中断触发事件为 'byteAvailable  
Event '
```

```
o_SerialPort.BytesAvailableFcnCount=10 %接收缓冲区每收到 10 个字节时，触发  
回调函数
```

```
o_SerialPort.BytesAvailableFcn=@EveBytesAvailableFcn(handles) %得到回调函  
数句柄
```

2.修改 instrcallback(obj,event)回调函数，对所发生的串口通信事件进行处理。

Matlab 缺省的回调函数 instrcallback (obj, event)存在于 instrcallback.m 文件中。该文件实际上是一个有待于用户修改的程序模块。其中只有一些最基本的程序代码，能够显示导致串口中断发生的是哪一类事件，中断事件所发生的时间以及导致事件发生的对象名等信息（修改回调函数文件时，注意要取消文件中相应信息后的分号，才能够在 Matlab 的命令窗口（command window）中将这些信息显示出来）。中断发生后的通信事件处理以及通信数据的分析处理任务，需要用户自行添加相应的服务程序代码。

4.3 绘制采集数据的曲线波形与数据显示

在 MATLAB 的二维绘图指令中，最重要、最基本的指令是 PLOT，其他许多特殊的绘图指令都是以它为基础而形成的。作为绘制线性坐标平面图形的函数 PLOT，对于不同的输入参数，该函数用不同的形式可以实现不同的功能。

4.3.1 绘制曲线波形

在 GUI 设计中，拖拽出 Axes 控件即可在界面上显示曲线坐标，再使用命令 plot(x, y)将采集到的数据曲线波形显示出来。其中 x, y 是同维向量，那么 plot 将绘制出以 x, y 元素为横、纵坐标的曲线。

```
plot(handles.axes_main,1:length(number),number) %确立该图在界面上的位置并  
按照比例缩放显示
```

为了更好地观察曲线波形，可以使用 grid on 命令，该命令主要用于在坐标平面上加上网格，从而更有利于观察和分析。

```
grid on; %加坐标网格
```

4.3.2 数据显示

在 GUI 界面设计中拖拽出 TXT 控件，然后修改其 Tag 属性为 edit_ReceiveData，用以显示接收到的串口所发送过来的数据，String 留空，并把 MAX 改为 2.0，以便可以多行显示数据。

4.3.3 采集图像

实现串口通信需要借助 COMTOOLS 工具向做好的 SerialPortCommunication 程序发送数据，此工具可以将发送的数据记录在指定的文件目录。其界面如图 4 所示，此时已打开 COM4，输入了要发送的数据。

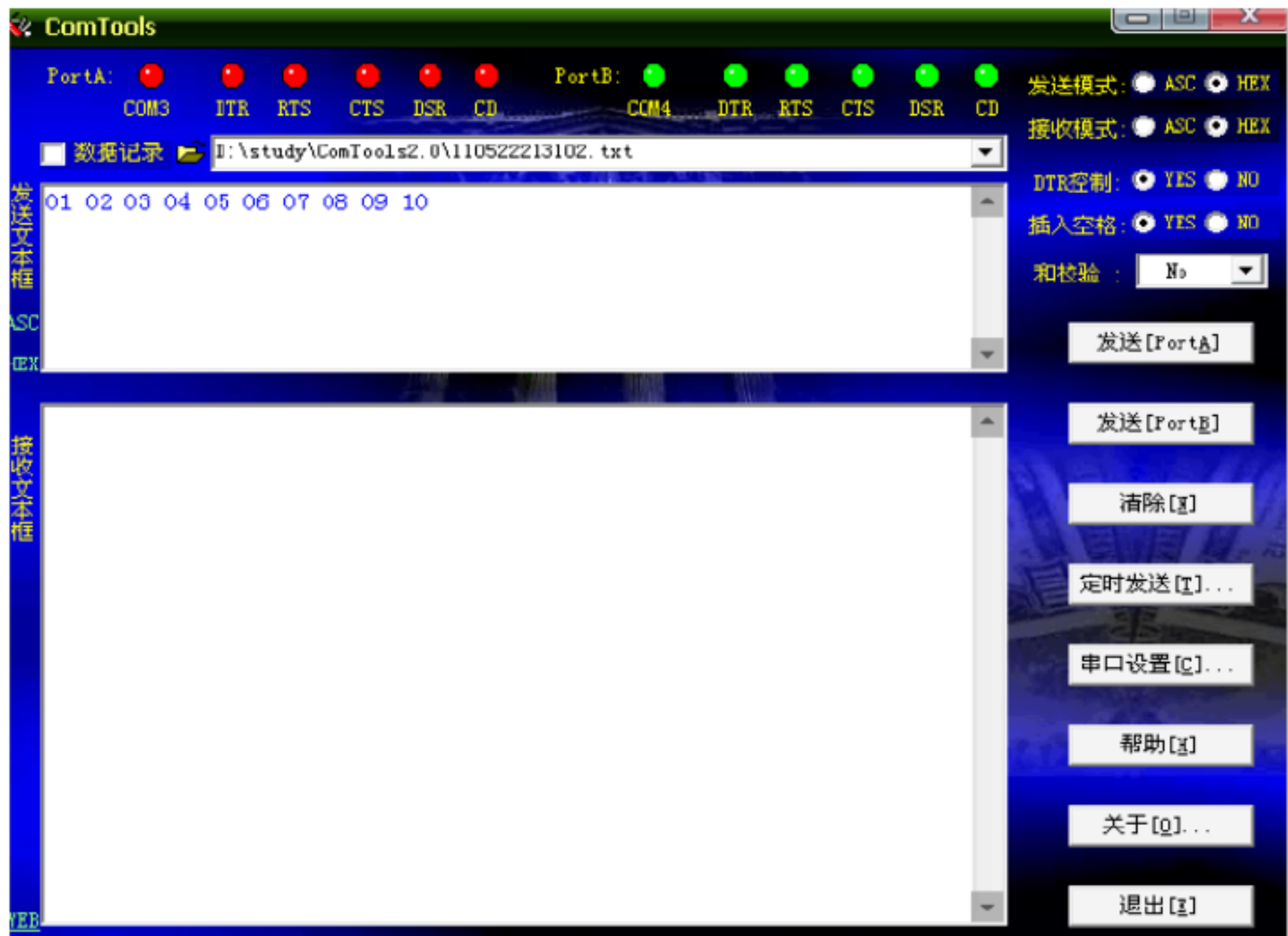


图 4：串口助手
Fig 4: Comtools

打开 SerialPortCommunication 的 COM3 串口，即可显示 COMTOOLS 发送过来的数据的曲线波形，如图 5。

如果要实现实时串口数据采集，则使用 COMTOOLS 工具里面的“定时发送”-“循环发送”功能即可实现。效果图如图 6 所示。

4.4 扩展功能——发送数据

在 Matlab 中实现 SerialPortCommunication 的发送数据功能其实也很简单，同接收数据一样，在 GUI 界面设计中拖拽出 TXT 控件，然后修改其 Tag 属性为 edit_SendData，用以显示发送到的串口的数据，String 留空，并把 MAX 改为 2.0，以便可以多行显示数据。发送 8 次后的应用效果如图 7。

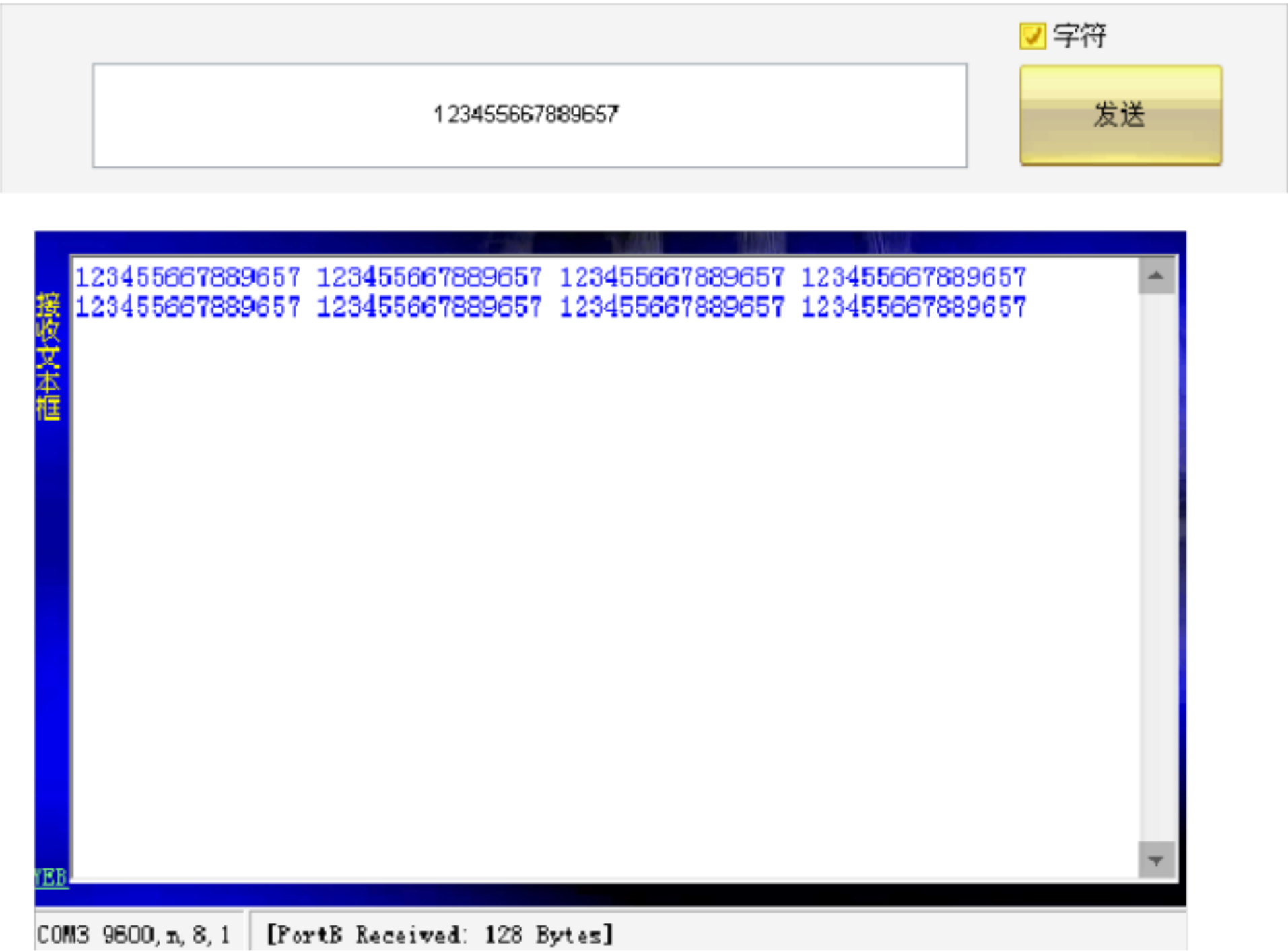


图 7：发送数据

Fig 7:Send Data

5 结论

可以看出，在 Matlab 中进行串行通信是十分方便的，编程较为简单。而且，在 Matlab 中串行通信的失误率很低，通信较为可靠，也可以采用增加握手信号以及数据校验等方式进一步增加通信的可靠性。 建立基于 MATLAB 的实时串口数据采集与曲线显示，能够实现信号采集、设备控制以及结果显示等功能。

实践证明该功能具有精度高、实时性好、性价比高、人机界面友好、升级修改简单等优点。在进行项目研究过程中，常常需要进行多次实验，采集大量的数据，并且要求对数据能实时地进行分析处理，该功能能很好地满足这种研究需要。此外，这一功能还可以

扩展应用到其他相关的领域中 ,如在语音识别工作中可以用该系统采集语音信号并且加入语音处理的相关分析等。 因此 ,该功能不仅具有良好的实用性 ,还可为其他的相关研究提供理论和应用基础。

参考文献

- [1] 何学明,陈学东,冯仰平,王华民 . 一种基于 MATLAB 的测量数据实时处理方法 [J]. 自动化仪表, 2005, 26(4): 45-47 .
- [2] 赵景波,逢锦梅,等 . MATLAB 控制系统仿真与设计 [M]. 北京:机械工业出版社, 2010: 126-130.
- [3] 李念强.数据采集技术与系统设计 [M]. 北京:机械工业出版社, 2009: 76-79.
- [4] 李炎新.用 Matlab 实现高速数据采集自动化 [J]. 测控技术, 2002, 11: 9-13 .
- [5] 余新平,张弦 . 基于 MATLAB 的实时数据采集与分析系统的实现 [J]. 长江大学学报(自科版), 2006,12(3): 80-83.
- [6] 付俊庆,李风铃. MATLAB 环境下的实时数据采集方法 [J]. 测控技术, 2004, 23(1): 30-32.
- [7] 陈德娟.基于 Matlab 的数据采集系统设计 [J]. 应用科技, 2003,30(4): 23-24.
- [8] 张立材.基于 Matlab 环境的串行数据通信 [J]. 微计算机信息, 2004, 20(3): 100-101.
- [9] MathWorks. Instrument Control Toolbox User's Guide Ver 5[Z]. MathWorks, 2000.
- [10] 刘志俭等. MATLAB 应用程序接口用户指南 [M]. 北京:科学出版社, 2000:256-257.
- [11] 王海龙,陈珊杰,,李茜,张鹏,刘涛,徐大华. 基于 MATLAB 的数据采集系统的研究 [J]. 中国农学通报, 2007,23(1): 381-384.
- [12] 张园,邢彦梅,何巧. 基于 MATLAB 数据采集系统的设计与实现 [J]. 电声基础, 2010,34(09): 37-39.
- [13] 李成功,赵昱. Matlab 下实现的实时数据采集和处理 [J]. 风机技术, 2002, 1: 41-43.
- [14] 伍萍辉,刘国繁,唐勇奇. MATLAB / RTW 在实时数据采集中的应用 [J]. 机床与压液, 2004,7:140-141.
- [15] Texas Instruments. TL16C550C Asynchronous Communications Element With AutoFlow Control[Z]. Texas Instruments.2001.
- [16] 罗华飞. MATLAB GUI 设计学习手记 [M]. 北京:北京航空航天大学出版社, 2011: 503-512.
- [17] 赵广元. MATLAB 与控制系统仿真实践 [M]. 北京:北京航空航天大学出版社, 2009: 114-120.
- [18] David McMahon. MATLAB demystified[M]. McGraw-Hill,2007.
- [19] 吴跃华,冯昊.基于 MATLAB 图形用户界面的对话框设计的应用 [J]. 福建电脑, 2003(9):

50-51.

[20] 伯晓晨. MATLAB 工具箱应用指南 [M]. 北京: 电子工业出版社, 2003: 95-105.

致 谢

本论文是在老师的悉心指导和热情关怀下完成的, 老师渊博的学识、严谨的治学态度、精益求精的工作作风和诲人不倦的高尚师德, 都将深深地感染和激励着我, 也为我提供了良好的实验条件, 在开展实验、撰写论文等方面提供了很多专业性的指导, 在此谨向老师致以诚挚的感谢!

生我者父母。感谢生我养我, 含辛茹苦的父母。是你们, 为我的学习创造了条件; 是你们, 一如既往的站在我的身后默默的支持着我, 没有你们就不会有我的今天。

感谢学术界的前辈们, 为我在这篇论文的完成中提供了大量的参考资料, 使我学习到了很多知识。

附录

附录 1: MATLAB 实时串口数据采集与曲线显示源代码

附录 1

```
function figure1_CloseRequestFcn(hObject, eventdata, handles)

global number;

global o_SerialPort;

%-----      关闭串口      -----

if strcmp(o_SerialPort.Status,'open')

    fclose(o_SerialPort);

end

number=[];

clear number;

delete(hObject);

%-----      按钮显示切换设置      -----

set(hObject,'Visible','off');

set(handles.pb_CloseSerialPort,'Position',get(handles.pb_OpenSerialPort,'Position'));

%-----      端口初始化      -----

set(handles.pb_CloseSerialPort,'Visible','on');

int_Index_COM=get(handles.pop_SerialPort,'Value');

string_COM=get(handles.pop_SerialPort,'String');

string_Select_COM=string_COM{int_Index_COM};

o_SerialPort=serial(string_Select_COM);

%-----      设置缓冲区大小为 1M-----

set(o_SerialPort,'InputBufferSize',1024000);

%----- EveBytesAvailableFcn      事件回调函数 -----

function EveBytesAvailableFcn( t,event,handles )

global o_SerialPort;

global number;

number=[number fread(o_SerialPort,10)'];

set(handles.edit_ReceiveData,'String',num2str(number));

% -----      发送按钮回调      -----

function pb_SendData_Callback(hObject, eventdata, handles)
```



```

global o_SerialPort;
string_edit_SendData=get(handles.edit_SendData,'String');
if(get(handles.cb_CharOrHex,'Value')==get(handles.cb_CharOrHex,'Max'))
    fprintf(o_SerialPort,string_edit_SendData);
else
    double_edit_SendData=str2num(string_edit_SendData);
    fwrite(o_SerialPort,double_edit_SendData);
end

%-----          串口下拉位选择  -----

function pop_SerialPort_CreateFcn(hObject, eventdata, handles)
if      ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%-----          发送数据  -----

function edit_SendData_CreateFcn(hObject, eventdata, handles)
if      ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%-----          接收数据  -----

function edit_ReceiveData_CreateFcn(hObject, eventdata, handles)
if      ispc      &&      isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```