

<<单片机与计算机串口通信实践>>

图书基本信息

书名：<<单片机与计算机串口通信实践>>

13位ISBN编号：9787121184840

10位ISBN编号：7121184842

出版时间：2013-1

出版时间：电子工业出版社

作者：张秀关

页数：315

字数：532000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<单片机与计算机串口通信实践>>

### 前言

采用Cygna!集成产品公司的C8051F020DK-U型单片机开发工具，仅需自制一条串行通信线，即可实现与一台计算机进行串口通信。

在单片机程序设计中，开发工具采用C51编程语言；在计算机程序设计中，开发工具采用Visual C++ 6.0集成开发环境。

为了易于代码的理解，本书首先对C8051F020型单片机、Windows API串口通信函数及多线程等相关知识做简单而必要的介绍。

本书中介绍的第一种串口通信模式，也是在同一时刻下，串行通信线路上仅有一个方向的数据传输；当串行通信流量较大、计算机的CPU使用率较高时，可能就会发生单片机部分测量值未被传送给计算机的情况。

在第二种串口通信模式下，可以保证单片机的测量值均被发送给计算机，但未考虑数据通信错误时的重新传输。

在第三种串口通信模式下，重新考虑了数据通信错误时的重新传输。

在本书中，编者分析并尝试了单片机接收两个连续字符的串口通信模式，且串口通信均正常。

为使计算机程序具有一定的通用性，创建了一个串口通信参数的设置对话框，相关地，创建并使用了一个ini配置文件。

采用C8051F020型单片机，测量了二路差分电压量、四路单端电压量，及其自身的片内温度；在ADC中断服务程序中，通过计数自身的执行次数，还精确记录了单片机的上电时间、串行通信时间。

单片机将各电压量的四次ADC平均值、所记录的两个时间值等，串行发送到计算机中，由计算机计算出实际的电压值、温度值，并存储到以当前时刻命名的Excel文件中。

Visual C++ 6.0向Excel 2003文件写入数据，采用了两种方法：ODBC、自动化。

本书由简入繁，详细阐述了三种串口通信模式。

第一种串口通信模式，见第2、3部分；第二种串口通信模式，见第4部分；第三种串口通信模式，见第5、6部分。

编者花费了大量的时间及精力，以尽最大可能对文中所涉内容进行注解。

三种串口通信模式的程序代码，后面一种都是以前面一种为基础进行修改的，可加深读者对代码的理解，以及查看三种模式下代码的差别之处。

编程期间所经历的错误，也列出了一二。

愿本书能够给您带来满意的收获。

在本书的出版过程中，得到了郭立总经理、符隆美编辑的有力支持，在此表示衷心的感谢。

由于编者能力、精力之所限，书中不妥之处在所难免，诚挚接受读者的批评及指正。

编者 2012年8月 于中国石油集团济柴动力总厂

## <<单片机与计算机串口通信实践>>

### 内容概要

采用Cygnal公司的C8051F020DK-U型单片机开发工具，即可实现与计算机的串口通信。本书由简入繁，详细阐述了三种串口通信模式，依据其各自的串行数据传输能力，可适用于不同的实际项目。

单片机测量了六路电压量、片内温度，记录了两个时间值，并将它们串行发送到计算机中；计算机将计算得到的物理电压值等，采用ODBC、自动化两种方法，存储到以当前时刻命名的Excel文件中。

书籍目录

第1篇 HTML

第1章 概述

1.1 应用程序的分类及特点

1.1.1 C/S编程模型

1.1.2 B/S编程模型

1.1.3 富互联网应用程序模型

1.2 W3C和ECMA标准

1.2.1 HTML

1.2.2 CSS

1.2.3 JavaScript和jQuery

1.2.4 DOM

1.2.5 C#语言

1.3 VS2010集成开发环境

1.3.1 安装集成开发环境

1.3.2 安装扩展开发工具

1.3.3 安装客户端浏览器

1.3.4 安装扩展管理工具

1.4 网页分类与基本代码交互技术

1.4.1 静态网页与动态网页

1.4.2 HTML页与Web窗体

习题

第2章 HTML基础

2.1 HTML入门

2.1.1 基本概念

2.1.2 HTML和XHTML

2.1.3 HTML5的基本结构

2.1.4 HTML5新增标记和不再支持的标记

2.2 字体和排版

2.2.1 h1标记到h6标记

2.2.2 p标记和br标记

2.2.3 hr标记

2.2.4 pre标记

2.2.5 span标记和div标记

2.2.6 粗体斜体和上下标控制标记(b、i、sup、sub)

2.2.7 短语标记(strong、em、code、cite、blockquote)

2.3 列表和图像显示

2.3.1 有序列表(ol、li)

2.3.2 无序列表(ul、li)

2.3.3 自定义列表(dl、dt、dd)

2.3.4 图像显示标记(img)

2.4 超链接和页内框架

2.4.1 绝对URL和相对URL

2.4.2 超链接标记(a)

2.4.3 页内框架标记(iframe)

2.4.4 利用a标记和iframe标记实现导航

## <<单片机与计算机串口通信实践>>

2.4.5 利用a标记和iframe标记链接到锚点

2.5 表格

2.5.1 表格标记(table)

2.5.2 规则表格

2.5.3 非规则表格

2.6 音频和视频

2.6.1 音频播放标记(audio)

2.6.2 视频播放标记(video)

2.7 语义化标记

习题

第3章 CSS样式控制

3.1 基本概念

3.1.1 CSS的级联控制

3.1.2 CSS的单位表示形式

3.1.3 设计和引用CSS

3.2 CSS选择器

3.2.1 基本选择器

3.2.2 关系选择器

3.2.3 特性选择器

3.2.4 伪元素选择器

3.2.5 伪类选择器

3.3 CSS的盒模型

3.3.1 盒模型简介

3.3.2 外边距(margin)

3.3.3 内边距(padding)

3.3.4 盒大小(box-sizeing)

3.3.5 盒阴影效果(box-shadow)

3.4 定位与布局控制

3.4.1 宽度和高度控制

3.4.2 定位控制

3.4.3 布局控制

3.5 边框控制

3.5.1 复合属性(border)

3.5.2 边框宽度和样式(border-width、border-style)

3.5.3 边框颜色(border-color)

3.5.4 边框圆角(border-radius)

3.6 字体和文本控制

3.6.1 字体控制

3.6.2 文本控制

3.7 背景控制

3.7.1 复合属性(background)

3.7.2 背景色与背景图像

3.7.3 背景图像的控制

习题

第4章 JavaScript和jQuery

4.1 在网页中使用JavaScript和jQuery

4.1.1 基本概念

## <<单片机与计算机串口通信实践>>

- 4.1.2 在网页中直接编写JavaScript和jQuery代码
- 4.1.3 在js文件中编写JavaScript和jQuery代码
- 4.1.4 提高JavaScript和jQuery性能的技巧
- 4.2 JavaScript和jQuery基本用法
  - 4.2.1 变量和运算符
  - 4.2.2 流程控制语句
- 4.3 JavaScript内置的函数和对象
  - 4.3.1 定义函数和对象
  - 4.3.2 JavaScript的内置函数
  - 4.3.3 数据类型转换
- 4.4 浏览器对象和客户端事件
  - 4.4.1 DOM
  - 4.4.2 window对象
  - 4.4.3 screen对象
  - 4.4.4 document对象
  - 4.4.5 客户端事件
- 4.5 jQuery选择器
  - 4.5.1 基本格式
  - 4.5.2 基本选择器
  - 4.5.3 层次选择器
  - 4.5.4 特性选择器
  - 4.5.5 基本过滤选择器
  - 4.5.6 内容过滤选择器
  - 4.5.7 可见性过滤选择器
  - 4.5.8 子元素过滤选择器
  - 4.5.9 表单对象选择器
  - 4.5.10 表单对象特性选择器
- 4.6 jQuery方法
  - 4.6.1 CSS操作
  - 4.6.2 HTML操作
  - 4.6.3 元素大小和位置操作
  - 4.6.4 元素特性和属性操作
  - 4.6.5 元素自定义数据操作
  - 4.6.6 文档处理
  - 4.6.7 事件处理
- 4.7 jQuery动画
  - 4.7.1 参数(duration、easing、callback)
  - 4.7.2 显示隐藏和切换(hide、show、toggle)
  - 4.7.3 滑动效果(slideDown、slideUp、slideToggle)
  - 4.7.4 淡入淡出效果(fadeIn、fadeOut)
  - 4.7.5 透明度渐变(fadeTo)
  - 4.7.6 自定义动画(animate)
  - 4.7.7 jQuery动画常见问题
- 习题
- 第5章 Canvas
  - 5.1 Canvas基础
    - 5.1.1 HTML5的canvas标记

## <<单片机与计算机串口通信实践>>

- 5.1.2 路径(beginPath、closePath、stroke、fill)
  - 5.1.3 移动坐标(moveTo)
  - 5.2 基本形状的绘制
    - 5.2.1 直线(lineTo)
    - 5.2.2 矩形(fillRect、strokeRect、clearRect)
    - 5.2.3 圆形和弧形(arc)
    - 5.2.4 用贝塞尔曲线绘制图形(quadraticCurveTo)
  - 5.3 颜色和样式
    - 5.3.1 填充样式和轮廓样式(fillStyle、strokeStyle)
    - 5.3.2 透明度(Transparency)
    - 5.3.3 渐变(Gradients)
    - 5.3.4 图案(Patterns)
  - 5.4 阴影和文字绘制
    - 5.4.1 阴影(Shadow)
    - 5.4.2 字符文本的绘制
  - 5.5 像素级绘图
    - 5.5.1 导入和绘制图像
    - 5.5.2 切片(Slicing)
    - 5.5.3 剪切(clip)和蒙版(mask)
  - 5.6 绘图状态与矩阵变换
    - 5.6.1 绘图状态
    - 5.6.2 变换(Transformations)
- 习题

### 第2篇 ASP.NET Web应用程序

#### 第6章 Web窗体和基本控件

- 6.1 基于事件的编程模型
  - 6.1.1 Web窗体页
  - 6.1.2 母版页和使用母版页的Web窗体页
  - 6.1.3 主题和外观
- 6.2 界面交互控件的分类
  - 6.2.1 表单(form)
  - 6.2.2 HTML控件
  - 6.2.3 HTML服务器控件
  - 6.2.4 Web服务器控件
- 6.3 HTML控件的基本用法
  - 6.3.1 label标记和button标记
  - 6.3.2 input标记
- 6.4 基本Web服务器控件
  - 6.4.1 标签和按钮
  - 6.4.2 文本框及密码框(TextBox)
  - 6.4.3 列表和下拉列表(ListBox、DropDownList)
  - 6.4.4 单选按钮(RadioButton、RadioButtonList)
  - 6.4.5 复选框(CheckBox、CheckBoxList)
  - 6.4.6 容器控件(Panel、Literal、View、MultiView)
  - 6.4.7 图像显示和区域控制(Image、ImageMap)
- 6.5 ASP.NET内置对象与状态管理
  - 6.5.1 ASP.NET提供的API

## <<单片机与计算机串口通信实践>>

6.5.2 Page类及其扩充类

6.5.3 ASP.NET状态管理

6.5.4 ASP.NET配置

6.6 页面切换与页间数据传递

6.6.1 从一个网页转到另一个网页

6.6.2 网页之间的数据传递

6.7 用户控件

习题

第7章 ASP.NET AJAX

7.1 基本概念

7.1.1 ajax

7.1.2 ASP.NET AJAX

7.2 内置的ASP.NET AJAX扩展控件

7.2.1 ScriptManager控件

7.2.2 UpdatePanel控件

7.2.3 UpdateProgress控件

7.2.4 Timer控件

7.3 AJAX控件工具包

7.3.1 弹出扩展

7.3.2 动画扩展

7.3.3 显示扩展

7.3.4 数据输入扩展

7.3.5 数据选择扩展

7.3.6 其他扩展

7.3.7 AJAX控件工具包中的独立控件

习题

第8章 在ASP.NET中使用jQuery

8.1 jQuery UI及其基本操作

8.1.1 jQuery UI的选项和方法

8.1.2 拖动(draggable)

8.1.3 拖放(dropable)

8.1.4 改变元素大小(resizable)

8.1.5 选择(selectable)

8.1.6 排序(sortable)

8.2 联合使用ASP.NET AJAX和jQuery UI

8.2.1 PageRequestManager类

8.2.2 拉帘式折叠面板(accordion)

8.2.3 自动完成(autocomplete)

8.2.4 按钮(button)

8.2.5 日期选择(datepicker)

8.2.6 对话框(dialog)

8.2.7 进度条(progressbar)

8.2.8 滑动条(slider)

8.2.9 页签(tabs)

8.3 jQuery代码和C#代码之间的数据交互

8.3.1 通过HiddenField控件传递数据

8.3.2 通过HTML5的data特性传递数据



## <<单片机与计算机串口通信实践>>

8.3.3 通过RegisterDataItem方法传递数据

8.3.4 通过jQuery ajax和Web服务实现jQuery和C#的交互

8.4 利用jQuery Validate实现客户端数据验证

8.4.1 调用不带参数的Validate方法实现客户端验证

8.4.2 在Validate方法的参数中定义验证选项

8.4.3 jQuery Validate提供的验证规则

习题

第9章 ASP.NET数据访问

9.1 LINQ基本技术

9.1.1 语言集成查询(LINQ)

9.1.2 查询表达式

9.1.3 Lambda表达式

9.2 ASP.NET数据绑定

9.2.1 数据绑定表达式

9.2.2 将数据源绑定到控件

9.3 利用实体框架访问数据库

9.3.1 实体框架基本概念

9.3.2 创建数据库和实体数据模型

9.3.3 GridView控件

9.3.4 EntityDataSource控件

9.3.5 实体框架完整示例

习题

第3篇 ASP.NET MVC Web应用程序

第10章 ASP.NET MVC基础

10.1 基本概念

10.1.1 什么是MVC

10.1.2 Web窗体和MVC的兼容性

10.2 ASP.NET MVC 3 Web应用程序模板

10.2.1 创建项目

10.2.2 查看MVC3应用程序的结构

10.3 ASP.NET路由

10.3.1 路由

10.3.2 URL路由

10.3.3 MVC应用程序中的典型URL模式

10.4 控制器和操作方法

10.4.1 控制器

10.4.2 操作方法

10.4.3 创建控制器和操作方法

10.5 视图

10.5.1 使用视图呈现UI

10.5.2 布局页和视图页

10.5.3 分部视图

10.5.4 创建视图页

10.5.5 将参数从控制器传递到视图

10.6 创建简单的模型

习题

第11章 Razor视图引擎

## <<单片机与计算机串口通信实践>>

### 11.1 基本知识

#### 11.1.1 Razor视图引擎的优点

#### 11.1.2 Razor基本用法

#### 11.1.3 在Razor视图中引用.NET命名空间

### 11.2 Razor引擎提供的类和对象

#### 11.2.1 视图操作类

#### 11.2.2 数据类型转换类

#### 11.2.3 HttpRequestBase对象和HttpResponseBase对象

#### 11.2.4 HttpServerUtilityBase对象

#### 11.2.5 其他对象

#### 11.2.6 Razor引擎增加的数据库操作方法

### 11.3 常用的HTML帮助器

#### 11.3.1 @helper标记

#### 11.3.2 form和input帮助器

#### 11.3.3 在控制器中处理同一个视图中的多个按钮

### 11.4 具有特殊功能的HTML帮助器

#### 11.4.1 URL帮助器

#### 11.4.2 WebGrid帮助器

#### 11.4.3 WebImage帮助器

#### 11.4.4 Chart帮助器

#### 11.4.5 Video帮助器

#### 11.4.6 WebMail帮助器

#### 11.4.7 其他HTML帮助器

### 习题

## 第12章 MVC中的数据访问和客户端验证

### 12.1 实体框架开发模式和技术

#### 12.1.1 实体框架开发模式

#### 12.1.2 采用数据库优先开发模式创建实体数据模型

#### 12.1.3 采用代码优先开发模式创建和访问数据库

### 12.2 数据验证与强类型视图

#### 12.2.1 使用特性控制显示格式和验证规则

#### 12.2.2 动态类型视图和强类型视图

### 12.3 MVC中的数据访问和验证示例

#### 12.3.1 修改与学生管理有关的控制器和视图

#### 12.3.2 利用WebGrid帮助器实现分页和排序

#### 12.3.3 为学生信息页添加搜索框

#### 12.3.4 使用模板化帮助器显示日历

## 附录 上机实验

### 附录A 基础型实验

#### 实验1 HTML5页面设计

#### 实验2 CSS3样式控制

#### 实验3 jQuery事件与动画设计

#### 实验4 Canvas绘图设计

#### 实验5 Web窗体界面交互

#### 实验6 AJAX和客户端验证

#### 实验7 ASP.NET数据访问

### 附录B 设计型实验

## <<单片机与计算机串口通信实践>>

B.1 需求说明

B.2 系统模块功能要求

B.3 成果要求

## &lt;&lt;单片机与计算机串口通信实践&gt;&gt;

## 章节摘录

版权页：插图：在串口通信协议中，计算机向单片机连续发送的字符个数，由一个更改为两个，其他无任何变化。

与“3.3单片机程序A—1”节相比较，单片机代码的差异之处，主要在UART0\_ISR（）中断服务程序的if（RIO）{...}模块中。

5.1单片机接收两个连续字符的分析 1.单片机的硬件条件 在8051系列单片机中，名为SBUF的串行数据缓冲寄存器，实际是两个寄存器：一个发送缓冲寄存器、一个接收缓冲寄存器。

读操作自动访问接收缓冲寄存器、写操作自动访问发送缓冲寄存器。

在C8051F020型单片机中，串行口UART0的数据缓冲寄存器是SBUF0、串行口UART1的数据缓冲寄存器是SBUF1。

在单片机串口的所有工作方式下，接收到的数据被放入进一个保持（移位）寄存器，然后才可能进入接收缓冲寄存器SBUF；这就允许在软件尚未读取前一个数据字节的情况下，开始接收第二个数据字节。

2.单片机的接收工作分析 单片机UART的接收控制电路，监测其RX信号状态：当检测到一个下降沿时，表明起始位出现了；在确定起始位之后，每接收到一个二进制数据位，就使接收（输入）移位寄存器（共九位）移动一位。

在C8051F020单片机的UART0工作于方式1、（SCON0.5 = SM20）= 1设置下，当接收到停止位的逻辑电平为1且接收中断标志位RIO = 0时，将接收到的8个数据位装入到接收（输入）数据缓冲器之中，并设置RIO = 1。

在这一个字符接收完毕之后，UART接收控制电路继续测试RX信号。

在C8051F020单片机的UART0工作于方式1设置下，当串行通信波特率为115.2kbps时，接收一个字符（共十位）所用的时间为 $10 / 115.2 = 1 / 11.52 = 0.0868\text{ms} = 86.8\mu\text{s}$ 。

在发送设备采取连续发送两个字符的策略下，当C8051F020单片机接收到第一个字符的停止位之后，只要在86.8us时间内，将这—个字符的8个数据位自移位寄存器装入到接收缓冲寄存器SBUF0，然后复位RIO = 0（其他操作无须执行完毕），则就有条件将第二个字符的8个数据位装入到接收数据缓冲器SBUF0中。

显然，这是非常简单的任务；但是，需要继续分析较之UART0\_ISR相同优先级及更高优先级的中断函数的执行时间。

在本设计中，系统时钟SYSCLK = 22.1184MHz，则对于单时钟周期的指令，每一条指令的执行时间为 $1 / 22.1184 = 0.045\mu\text{s}$ ，1us可执行22.1184条指令，19.53us可执行 $19.53 \times 22.1184 = 432$ 条指令；同样在19.53us时间内，则可执行 $432 / 2 = 216$ 条双时钟周期指令。

本设计中的UART0\_ISR中断函数，其RIO模块中的执行内容很少。

虽然ADCO\_ISR的中断优先级高于UART0\_ISR，但ADCO\_ISR函数中的代码也不算长，且已知ADCO\_ISR执行频率为19.53us，可推知，为了给UART0\_ISR中断函数留有处理时间，则ADCO\_ISR中断函数的执行时间必须小于19.53us。

## <<单片机与计算机串口通信实践>>

### 编辑推荐

《单片机与计算机串口通信实践》具有思路清晰、讲解详细的特点，从实际应用出发、程序代码经过反复调试，可直接为采用Keilc的单片机编程者、采用VC++6.0的串口编程者使用。

<<单片机与计算机串口通信实践>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介, 请支持正版图书。

更多资源请访问:<http://www.tushu007.com>