

# **NCD-CIP51F020-B 实验机使用说明书**

# 目 录

目 录 .....	1
前 言 .....	4
第一章：实验设备简介 .....	5
1.1 系统实验设备的组成 .....	5
1.2 C8051Fxxx 单片机开发工具简介 .....	5
1.2.1 开发工具概述 .....	5
1.2.2 开发工具主要技术指标 .....	5
1.2.3 IDE 软件运行环境 .....	5
1.2.4 开发工具与 PC 机硬件连接 .....	5
1.3 C8051F020 CPU 板简介 .....	6
1.3.1 C8051F020 CPU 板概述 .....	6
1.3.2 C8051F020 片上系统单片机片内资源 .....	6
1.4 系统实验板 .....	7
第二章 C8051Fxxx 单片机开发工具集成开发环境 IDE 使用说明 .....	9
2.1 C8051Fxxx 集成开发环境软件简介 .....	9
2.2 C8051Fxxx IDE 界面 .....	10
2.2.1 窗口 .....	10
2.2.2 菜单 .....	11
2.2.3 工具栏 .....	14
2.2.4 状态栏 .....	16
2.3 软件的基本操作 .....	16
2.3.1 项目管理 .....	16
2.3.2 源程序的编辑 .....	17
2.3.3 源程序的编译和链接 .....	17
2.3.4 集成开发环境与您的目标系统连接 .....	18
2.3.5 下载代码到 FLASH .....	18
2.3.6 设置断点和观察点 .....	19
2.3.7 运行和停止 .....	20
2.3.8 查看和修改存储器、寄存器和变量 .....	20
第三章 C8051F 单片机实验 .....	23
3.1 C8051F 单片机 I/O 口交叉开关设置 .....	23
3.1.1 优先权交叉开关译码器 .....	23
3.1.2 交叉开关引脚分配 .....	24
3.2 在 C8051F020 单片机上的初级软件实验(汇编语言) .....	25
3.2.1 无符号数、十进制加法实验 .....	25
3.2.2 数据传送实验 .....	26
3.2.3 数据排序实验 .....	26

3.2.4 整数二翻十实验 .....	28
3.3 I/O 口实验 ( C 语言 ) .....	32
3.3.1 LED 驱动实验 .....	32
3.3.2 光耦/继电器驱动实验 .....	33
3.4 串行总线通信实验 ( C 语言 ) .....	错误！未定义书签。
3.4.1 RS-232 串口通信实验 .....	39
3.4.2 RS-485 串口通信实验 .....	40
3.5 并行总线扩展实验 .....	40
3.5.1 6 位数码管显示实验	
3.5.2 键盘实验	
3.5.3 实时时钟芯片 DS12887 应用实验	
3.5.4 外部扩展 SRAM 实验	
3.5.5 1602 字符型 LCD 显示实验	
3.6 电机驱动实验	
3.6.1 步进电机驱动实验	
3.6.2 直流电机驱动实验	
3.7 可编程计数器阵列(PCA)实验	
3.7.1 PCA 频率输出功能实验	
3.7.2 PCA 捕捉功能实验	
3.7.3 PCA 输出 16 位 PWM 实验	
3.8 A/D 转换实验	
3.9 比较器功能实验	
3.10 红外收发实验	
3.11 电子影响实验	
3.12 频率计实验	
3.13 外部中断实验	
附录 1 C8051F 系列单片机开发工具 JTAG 口线定义 .....	58
附录 2 系统实验板原理图 .....	58
附录 3 系统实验板元件排列图 .....	58

## 前 言

MCS-51 单片机及其衍生产品在我国乃至世界范围获得了非常广泛的应用，尤其是在我国，大部分大专院校都采用 MCS-51 单片机作为教学机型，大部分单片机系统工程师都熟悉 MCS-51 单片机。随着一些高集成度、高性能的 8 位和 16 位 RISC 单片机的推出，基于标准 8051 内核的单片机正面临着退出市场的境地。因此一些半导体公司开始对传统 8051 内核进行大的改造，主要是提高速度和增加片内模拟和数字外设，以期大幅提高单片机的整体性能。Silicon Labs 集成产品公司推出的 C8051F 单片机的出现令业界人士耳目一新，使广大单片机系统设计人员看到了 51 单片机新的曙光。

MCS-51 单片机引入教学已达 10 年之久，大家一直期盼着有更先进的单片机引入教学，C8051F 已成为事实上 51 的升级换代产品，那么必然会成为经典单片机教学的方向，做为 Silicon Labs 产品中国代理的新华龙电子有限公司适时的为各教学单位设计了 C8051F 单片机系统实验设备，该设备可为大专院校教学实验课提供多项实验，使得学生和单片机的初学者建立单片机系统概念，并可完成基础应用的培训，使学生掌握 C8051F 的基本应用。

本实验设备比常规的教学设备增加了以太网测控实验，可做为研究生教学选修课。也可起到为本科生提供了解当前测控领域前沿技术的引导作用。

欢迎教学单位选用 Silicon Labs C8051F 单片机教学实验设备。

新华龙电子有限公司

[Http://www.xhl.com.cn](http://www.xhl.com.cn)

Email:[support@xhl.com.cn](mailto:support@xhl.com.cn)

2006 年 4 月

# 第一章：实验设备简介

## 1.1 系统实验设备的组成

NCD-CIP51F020 综合实验设备由 C8051F020 CPU 板和系统实验板二部分组成,应用该设备可进行片上系统单片机较典型应用的实验,请参见以下介绍。

## 1.2 Silicon Labs C8051F 单片机开发工具简介

### 1.2.1 开发工具概述

Silicon Labs 的开发工具实质上就是计算机 IDE 调试环境软件及计算机 USB 到 C8051F 单片机 JTAG 口的协议转换器(UEC5)的组合(此部分电路已经集成到系统实验板上)。Silicon Labs C8051F 系列所有的单片机片内均设计有调试电路,该调试电路通过边界扫描方式获取单片机片内信息,通过 4 线的 JTAG 接口与开发工具连接以便于进行对单片机在片编程调试。NCD-CIP51F020 系统实验设备中的 C8051F020 CPU 板上的单片机为 C8051F 系列中的 F020。

适配器(UEC5)一端与计算机相连,另一端与 C8051F 单片机 JTAG 口相连,应用 Silicon Labs 提供的 IDE 调试环境或 Keil 的 uVision2 调试环境就可以进行非侵入式、全速的在系统编程(ISP)和调试。

Silicon Labs 开发工具支持观察和修改存储器和寄存器支持断点、观察点、堆栈指示器、单步、运行和停止命令。调试时不需要额外的目标 RAM、程序存储器、定时器或通信通道,并且所有的模拟和数字外设都正常工作。

### 1.2.2 开发工具主要技术指标

- ◆ 支持的目标系统：所有 C8051Fxxx 系列单片机。
- ◆ 系统时钟：最大可达 25Mhz。
- ◆ 通过 USB 接口与 PC 机连接。
- ◆ 支持汇编语言和 C51 源代码级调试。
- ◆ 第三方工具支持 ( Keil C )。

### 1.2.3 IDE 软件运行环境

要求 PC 机能够运行开发工具软件并能与串行适配器通信。对 PC 机有如下系统要求：

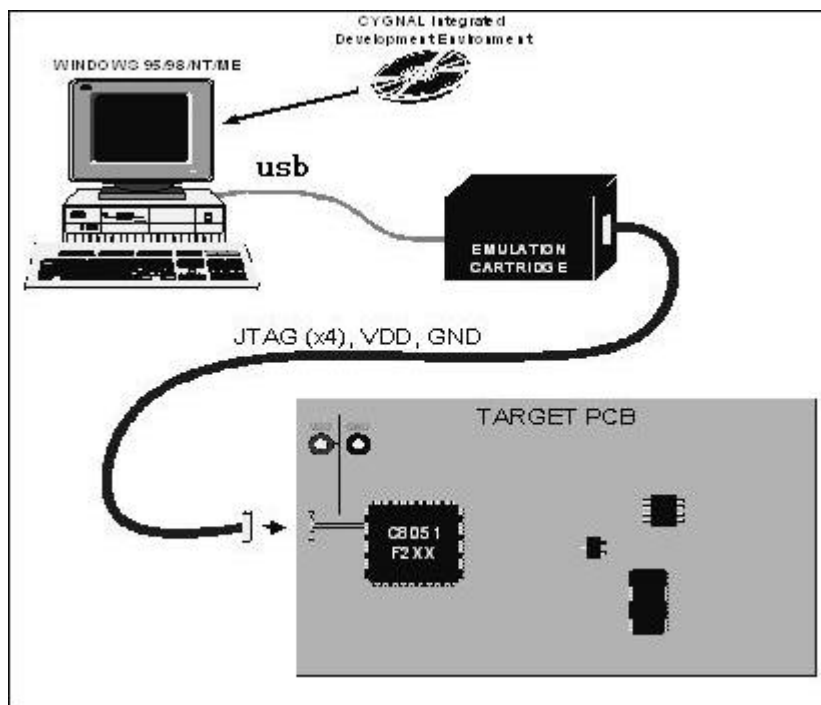
- ◆ Windows 95/98/Me/NT/2000/XP 操作系统
- ◆ 32Mb RAM
- ◆ 40Mb 自由硬盘空间
- ◆ 空闲的 USB 口

### 1.2.4 开发工具与 PC 机硬件连接

在系统编程和调试环境如图下所示。

硬件连接及软件安装：

- ◆ 将 USB 串行电缆的一端与教学机的 USB 连接；
- ◆ 连接 USB 串行电缆的另一端到 PC；
- ◆ 给目标系统上电；
- ◆ 插入 CD 并运行“SETUP.EXE”，将 IDE 软件安装到您的 PC 机；
- ◆ 在 PC 机的开始菜单的“程序”项中选择 Silicon Labs IDE 点击 Silicon Labs 图标，运行 IDE 软件。



### 1.3 C8051F020 CPU 板简介

#### 1.3.1 C8051F020 CPU 板概述

C8051F020 CPU 板是为了便于安装而设计的;C8051F020 CPU 板是将 C8051F020 的所有引脚(100 个引脚)引到四个单排针;该四组单排针可与不同用户设计的应用系统连接,如在该教学系统上将和系统实验板连接。

#### 1.3.2 C8051F020 片上系统单片机片内资源

##### 、模拟外设

- (1) 逐次逼近型 8 路 12 位 ADC0  
转换速率最大 100ksps  
可编程增益放大器 PGA  
温度传感器
- (2) 8 路 8 位 ADC1 输入与 P1 口复用  
转换速率 500ksps  
可编程增益放大器 PGA

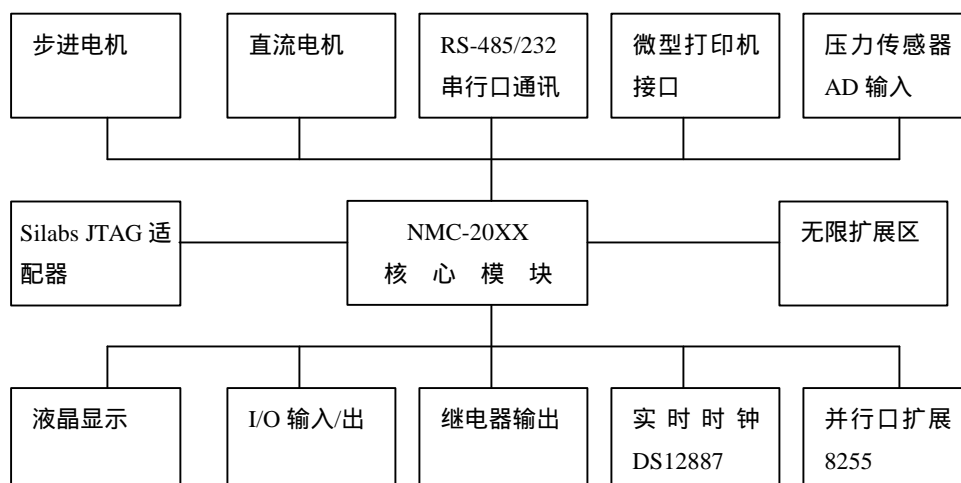
- (3) 两个 12 位 DAC
- (4) 两个模拟电压比较器
- (5) 电压基准 内部提供 2.43V  
外部基准可输入
- (6) 精确的 VDD 监视器
- 、 高速 8051 微控制器内核
- 流水线式指令结构速度可达 25MIPS
- 22 个矢量中断源
- 、 存储器
- 片内 4352 字节数据 RAM
- 64KB Flash 程序存储器可作非易失性存储
- 外部可扩展的 64KB 数据存储器接口
- 、 数字外设
- 8 个 8 位的端口 I/O
- I<sup>2</sup>C、SPI、2 个增强型 UART 串口
- 可编程的 16 位计数器/定时器阵列(PCA)
- 5 个通用 16 位计数器/定时器
- 专用的看门狗 WDT
- 更详细资料可参见 Silicon Labs C8051F02x datasheet

## 1.4 系统实验板

系统实验板除与 C8051F020 CPU 板接口外还配置了丰富的硬件资源，各教学单位可根据自己的需要安排实验内容。

系统实验板硬件逻辑结构如下图，电路图及器件分布图参见附录

系统实验板框图



系统实验板可按教学需要与 C8051F020 CPU 板组成目标系统，然后由与 C8051F020 CPU 板中的 C8051F020 单片机 JTAG 口相连接的 Silicon Labs 开发工具在系统实现开发调试。

该机除了能完成全部单片机基础实验教学外，还采用了扩展总线外引、标准插件组成无限开展实验区的办法。更方便的扩展完成当前各种前沿的、智能化的、单片机与新型器件的应用实验。如可扩展：各种 IC 器件、SPI 器件、1-WIRE、2-WIRE、3-WIRE、的实验，接触卡、非接触卡、USB、CAN、以太网的实验等。高校教师也可自行扩展自己所需的专用实验板。

实验者可参照第三章后的实验例程下装各应用程序，也可独立设计程序(C 或汇编)，然后就可开始在系统调试了。



## 第二章 C8051F 单片机开发工具集成开发环境 IDE 使用说明

### 2.1 Silicon Labs 集成开发环境软件简介

Silicon Labs 集成开发环境软件提供了开发和测试您的项目所必须的工具。具有如下特点：

- ◆ 源代码编辑器
- ◆ 项目管理器
- ◆ 集成 8051 宏汇编器
- ◆ FLASH 编程器
- ◆ 支持 Silicon Labs 的全速、非侵入、在线调试逻辑
- ◆ 实时断点
- ◆ 比使用 ICE 芯片、目标仿真头、电缆与仿真插座的仿真系统有更优越的性能
- ◆ 源程序级调试
- ◆ 有条件的存储器观察点
- ◆ 存储器与寄存器检查与修改
- ◆ 单步与连续单步执行方式
- ◆ 支持第三方开发工具
- ◆ MCU 程序代码初始化配置向导

#### 源代码编辑器

编辑器包括所有标准的 Windows 编辑功能,包括剪切、粘贴、复制、取消/重复、查找/替换及书签等。并为 8051 汇编语言和 C 语言提供了彩色句法加亮功能。你可以扩充加亮的关键字的目录,也可以定义所使用的颜色,用户可配置字体、文本颜色与 Tab 键设置。

#### 项目管理器

一个项目由源文件、目标与库文件、工具配置和 IDE 查看等组成。项目管理保存了查看与工具设置,及在编译中所使用的多卷文件,包括要通过第三方汇编器、编译器和连接器处理的文件。

#### 集成 8051 宏汇编程序

8051 宏汇编程序与 IDE 结合成一体。此汇编程序接受 Intel MCS-51 可兼容源文件并且创立可下载的 Intel 十六进制文件。它也产生所有的必要调试信息,提供汇编语言源程序级调试。

#### FLASH 编程器

编译之后,在 IDE 界面集成的 FLASH 编程器允许代码立即下载至 MCU 在片闪存,将源代码修改与在系统调试之间的时间最小化。

#### 非侵入调试

连接到 MCU 片上调试电路的 IDE,使用最终应用中安装的 MCU 进行全速、非侵入式、在系统编程调试。片上调试逻辑比使用 ICE 芯片、目标仿真头与有噪声的电缆的仿真系统性能更优越,为评估你的混合信号设计的实际模拟性能,提供了必要的信号完整性。

## 源程序级调试

源程序窗口也是工作调试窗口。当监控寄存器与存储器内容时，你可以在源程序中观察当前的程序计数器位置，设置并且清除断点，执行单步运行。

### 断点

断点可以设置在源程序行中，在执行指定源程序行的第一指令之前，立即停止执行。断点由 MCU 的片上调试电路支持，并且不影响程序的实时执行。

### 存储器观察点

当一个或者多个数据存储器位置或者寄存器与指定的值符合或者改变时，可以有条件的定义存储器观察点，停止程序执行。

### 第三方工具支持

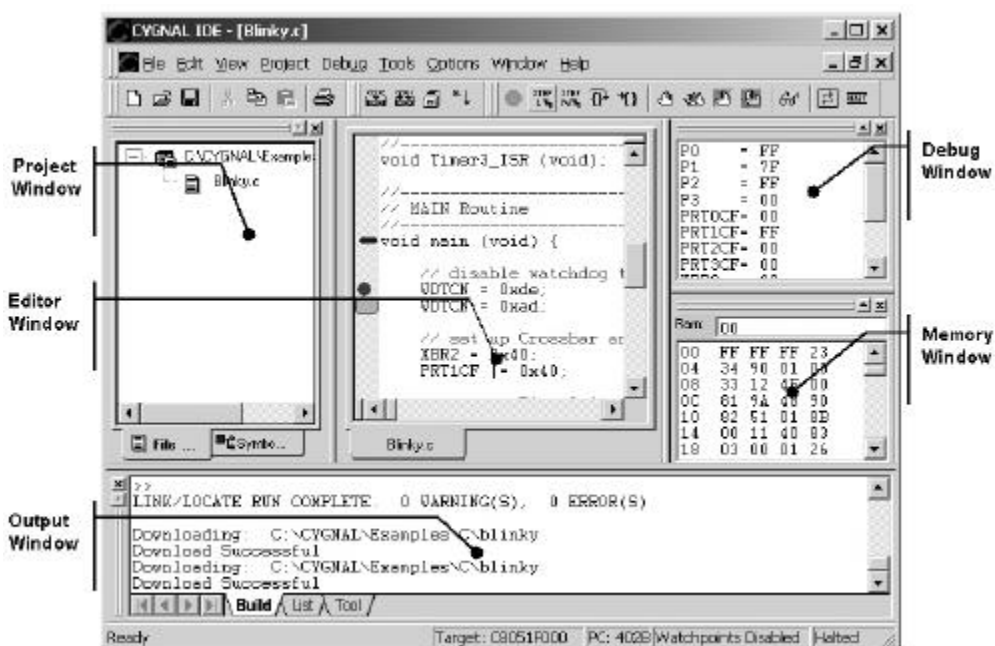
完全支持 FLASH 编程和 Intel OMF-51 绝对目标文件的源程序级调试，允许在软件开发时使用第三方链接工具。

### 配置向导

配置向导自动地产生 MCU 和片上外设初始化代码。单击检验栏，并且在对话框中输入数值产生所需的带注释的汇编语言代码，使能和配置外部设备，设定输入/输出端口功能，并指定 MCU 等操作。

## 2.2 Silicon Labs IDE 界面

Silicon Labs IDE 的主界面如下图所示。



### 2.2.1 窗口

Silicon Labs IDE 的主界面由项目窗口、编辑/调试窗口和输出窗口组成。

#### 项目窗口

- ◆ 文件察看，用于察看和管理与项目相关的文件。
- ◆ 符号察看，用于察看项目中使用符号的地址。

**编辑/调试窗口**

- ◆ 编辑窗口 用于项目中所选文件的编写或编辑。
- ◆ 调试窗口 代码下载后，在调试期间此窗口用于观察存储器、寄存器和变量等。

**输出窗口**

输出窗口是由三个复选窗口组成，这些复选窗口用于显示调试过程中的信息。

- ◆ Build 选项窗口 显示由集成的汇编/编译/链接工具产生的输出信息。如果在汇编/编译过程中出错，用户可以双击窗口中的一条错误信息，则在编辑窗口中就会显示发生错误的代码行。
- ◆ List 选项窗口 用来显示最新编译或汇编所产生的列表文件。
- ◆ Tool 选项窗口，如果工具输出被重定向到"tool.out"文件名，此窗口将显示自定义工具所产生的输出。

**2.2.2 菜单****File 菜单**

菜单项	描述
New File ( 新文件 )	创建新文件。
Open File ( 打开文件 )	打开文件对话框，打开所选文件。
Close File ( 关闭文件 )	关闭当前打开的文件。如果打开的文件已被编辑，则将弹出对话框询问是否存盘。
Save ( 保存 )	保存当前激活的文件。
Save As ( 另存为 )	允许当前打开的文件换名存盘。
Save All ( 保存所有 )	IDE 将保存所有打开的文件。
Print Setup ( 打印设置 )	打开打印机对话框，选择打印机参数。
Print ( 打印 )	打印当前文件。
Recent Files ( 最近文件 )	此菜单区将列出 IDE 最近编辑的文件。
Recent Projects ( 最近项目 )	此区域提供一种快捷方式用以打开 IDE 的最近打开的项目。
Exit ( 退出 )	推出 IDE。

**Edit 菜单**

菜单项	描述
Undo ( 撤消 )	此命令使编辑器退回到最近的编辑命令。
Redo ( 重做 )	此命令使编辑器退回到最近的 undo 命令。
Cut ( 剪切 )	此命令使选定的文字 ( 高亮 ) 被删除，但将文字复制到剪切板。
Copy ( 复制 )	此命令将选定的文字复制到剪切板。
Paste ( 粘贴 )	此命令将剪切板的内容粘贴到当前光标位置。
Find ( 查找 )	此命令打开对话框，用户可键入查找的参数并在当前文件查找。
Replace ( 替换 )	此命令打开对话框允许用户在当前文件查找并替换字符串。

**View 菜单**

菜单项	描述
Debug Windows ( 调试窗口 )	此菜单包含有子菜单，在子菜单中列出了所有存储器和寄存器窗口（这些窗口只有在调试时才可见）。
Project Window ( 项目窗口 )	此菜单项触发显示 IDE 项目观察窗口。
Output Window ( 输出窗口 )	此菜单项触发显示 IDE 输出窗口。
Toolbars ( 工具栏 )	此项目菜单允许用户选择工具栏是否可见，也允许用户定制工具栏。
Status Bar ( 状态栏 )	此菜单允许用户触发显示 IDE 状态栏。
Workbook Mode( 工作簿模式 )	此菜单项允许用户在正常和笔记本模式之间选择。

**Project 菜单**

菜单项	描述
Add Files to Project ( 加文件到项目 )	此菜单命令将添加文件到当前项目。
Assemble/Compile Current File and Stop Assemble/Compile Current File( 汇编/编译当前文件和停止汇编/编译当前文件 )	此菜单将汇编/编译当前文件。汇编器/编译器输出将显示在输出窗口中的 build 窗。如果编译器/汇编器报告错误，输出窗口中将显示错误概要。用鼠标双击错误，IDE 将显示相应发生错误的源代码行。
Build/Make Project ( 生成项目 )	此菜单命令将生成目标代码。
Open Project ( 打开项目 )	调用浏览对话框浏览项目文件并打开。
Save Project ( 保存项目 )	保存当前打开的项目。
Save Project As ( 另存项目为 )	换名保存项目。
Close Project ( 关闭项目 )	关闭当前打开的文件和窗口。
Tool Chain Integration ( 工具链接集成 )	调用集成链接工具对话框，来定义外部汇编器、编译器和链接器。
Target Build Configuration ( 目标生成配置 )	调用目标生成配置对话框，用来定义生成过程。

**Debug 菜单**

菜单项	描述
Connect ( 连接 )	通过 EC2 将串口目标系统连接起来。
Disconnect ( 断开 )	释放计算机串口。
Download ( 下载 )	将下载当前打开项目代码到 Flash。如果当前无文件或项目打开，将弹出对话框允许用户选择文件下载。但文件必须是 Intel Hex 或 OMF-51 格式。
Go ( 运行 )	将释放调试中断信号，允许运行用户程序代码。
Stop ( 停止 )	将发出调试中断信号使芯片停止运行程序，并开始执行调试用户程序代码。

Step ( 单步 )	单步执行用户程序代码。
Multiple Step ( 多步 )	执行 N 步用户程序代码。
Step Over ( 越过单步 )	允许用户程序代码越过当前代码行执行下面的代码。
Run to Cursor( 运行到光标 )	将允许用户程序代码运行到光标所在的代码行。
Breakpoints ( 断点 )	调用断点管理对话框，显示当前所有断点信息，断点可以加入/删除/允许/禁止。
Watchpoints ( 观察点 )	调用观察点管理对话框，显示当前所有观察点信息，观察点可以加入/删除/允许/禁止。
Refresh ( 刷新 )	当在 IDE 中修改某些值后，强制写仿真器，修改存储器或寄存器值。
Reset ( 复位 )	复位按钮迫使 IDE 和硬件返回到调试初始状态。

### Tools 菜单

菜单项	描述
Silicon Labs Configuration Wizard( Silicon Labs 配置向导 )	调用 Silicon Labs 配置向导，能快速生成带有外设详细信息的初始化配置代码。
Memory Fill ( 填充存储器 )	此菜单包含有子菜单，调用填充存储器对话框，填充 RAM、代码空间或外部 Mem。
Erase Code Space ( 擦除代码空间 )	删除和复位整个 FLASH 代码空间。
Output Memory to File ( 输出存储器到文件 )	调用输出存储器到文件对话框。
Add/Remove User Tool ( 加入/移出用户工具 )	调用对话框管理 IDE 用户工具。可以添加、移出或修改用户工具。

### Options 菜单

菜单项	描述
Multiple Step Configuration ( 多步配置 )	调用多步配置对话框。
Serial Port ( 串口 )	选择 RS232 串口。
Serial Baud Rate ( 串口波特率 )	选择串口波特率。
Toolbar Configuration ( 工具栏配置 )	调用对话框选择允许那些工具栏可见；工具栏按钮配置，还可创建新工具栏。
Toolbar Extended Styles ( 工具栏扩展类型 )	调用对话框允许选择各种工具类型。
Editor Font Selection ( 编辑器字体选择 )	调用对话框允许设定编辑器字型大小和颜色。
Editor Tab Configuration	调用对话框允许 tab 键设置。
Select Language ( 选择语言 )	强制编辑器使用特殊语言配置文件。
Debug Window Font Selection ( 调试窗口字体选择 )	调用对话框允许选择调试/编辑窗口的字体。
File Backup Settings ( 文件备份设置 )	调用对话框允许选择备份文件的数量。

**Window 菜单**

菜单项	描述
Cascade (层叠)	标准 windows 层叠格式。
Tile Horizontal (水平平铺)	标准 windows 水平平铺格式。
Tile Vertical (垂直平铺)	标准 windows 垂直平铺格式。







**Help 菜单**

菜单项	描述
SILICON LABS IDE Help	调用在线帮助程序。
Keil Assemble/Link Manual	Keil 汇编/链接手册。
Keil Compile Manual	Keil 编译手册。
About SILICON LABS IDE	显示 IDE 版本信息。

**2.2.3 工具栏**

工具栏	按钮		描述
文件/编辑		新建	创建一个新文件
		打开	打开一个文件
		保存	保存当前文件
		剪切	剪切选定文本到剪切板
		复制	复制选定文本到剪切板
		粘贴	粘贴剪切板到光标位置
		打印	打印当前文件
编译和生成代码		汇编/编译 停止生成	汇编/编译当前文件 停止生成代码
		生成代码	汇编/编译和链接文件。
			

		连接 断开	连接 IDE 和目标板。 断开按钮释放串口
		下载	下载代码到目标硬件 flash
			
调试		运行/停止	开始/停止执行目标处理器中的程序代码
			
		复位	硬件和 IDE 返回调试初态
		单步	执行一条用户代码程序
		多步	执行 N 条用户代码程序
		单步越过	单步越过函数或子程序
		运行到光标	程序运行到光标处代码行
		插入/移出断点	设置/清除光标处断点
		移出所有断点	移出所有断点
		允许/禁止断点	激活/禁止当前断点
		禁止所有断点	禁止所有断点
		内部观察点对话框	打开内部观察点对话框
		刷新	IDE 改变数值后，强制写仿真器
			
调试窗口		SFR 寄存器察看窗	触发察看窗口
		寄存器察看窗	触发察看窗口
		RAM 察看窗	触发察看窗口

		代码察看窗	触发察看窗口
		反汇编察看窗	触发察看窗口
书签		下一个书签	移动光标到下一书签位置
		触发书签	设置/清除光标处书签
		上一个书签	移动光标到前一书签位置。
		移出所有书签	移出所有书签

### 2.2.4 状态栏

状态栏显示您目标系统中使用的 MCU 的型号，程序计数器 PC 的值，观察点的状态，程序的运行状态及光标所在的行和列。

## 2.3 软件的基本操作

### 2.3.1 项目管理（创建和打开项目）

#### (1)、创建项目

项目是用来保存文件、链接工具、目标代码生成和窗口配置信息的。可以使用“Project”菜单中的“New Project”选项或“Save Project As”选项来创建项目。如果使用“Save Project As”选项，将出现“Save Workspace”对话框来选择项目名称和存放的位置(项目文件的扩展名为.wsp)。

一旦项目被保存，将保存如下信息：

- ◆ 当前所有打开的文件。（如果创建了新文件，且未存盘，则 IDE 将弹出对话框提示你保存文件）
- ◆ 集成链接工具的设置
- ◆ 目标生产配置
- ◆ 主 IDE 窗口及已经打开的调试窗口的位置和大小。
- ◆ 编辑器的设置如字体和文字颜色等。

#### (2)、重新打开项目

有两种不同的方法打开项目：

- 1) 选择 File 菜单中的“Recent Projects”子菜单区中列出的最近打开过的项目。
- 2) 使用 Project 菜单中的“Open Project”命令，调用“Open Workspace”对话框允许你浏览计算机中的项目文件 (\*.wsp) 并打开所选文件。



### (3)、保存一个项目

保存项目用“Project”菜单中的“Save Project”选项。项目不必每次打开后都保存，遇到下面的情况需保存项目：

- ◆ 已打开新文件且将作为项目的一部分。
- ◆ 已打开新窗口，且每次打开项目时都需要重新打开这些窗口
- ◆ IDE 窗口的位置和/或大小改变了，且在下次项目重新打开时需保留这种变化。

### (4)、在项目中添加文件

可用下面的方法向已存在的项目中添加文件：



- A. 在项目窗口的 File 选项窗口中添加文件到项目。
  - 1、在项目或组上点击鼠标右键。
  - 2、在弹出菜单点击“Add Files...”菜单选项。
- B. 从 Project 菜单中加文件到一个打开的项目
  - 1、打开项目
  - 2、在 Project 菜单中用“Add Files to Project”选项。
- C. 从“Build Button Definition”对话框中添加文件到项目
  - 1、从“Project”菜单中打开“Target Build Configuration”对话框。
  - 2、点击“Customize”按钮。
  - 3、使用“Add Files to Project”按钮。

### (5)、从项目中移出文件

从已有项目中移出文件可用下面的方法：


在项目窗口的文件窗口中在要移出的文件上点击右键，选择 Remove filename from project。

## 2.3.2 源程序的编辑

IDE 包括一个全功能的编辑器。可用文件菜单中的“New File”命令来新建文件，或用文件工具栏中的“New”按钮，然后开始键入源程序。只有当文件的扩展名为.asm 或 .c 时，才具有源程序关键字符彩色显示功能。可用文件保存按钮，或用文件菜单中的“Save”或“Save As”命令保存文件。然后再将编辑好的源代码添加到您的项目中。

## 2.3.3 源程序的编译和链接


### (1)、汇编和编译

可用生产工具栏中的汇编/编译按钮或“Project”菜单中的“Assemble/Compile File”命令来汇编/编译一个文件。如果一个项目或文件是打开的，那末当前活动的文件将被汇编/

编译。

当汇编/编译完成后，将在输出窗口的“Build”选项窗中显示汇编/编译结果，如果您的源程序有错误，将在输出窗口中提示，双击错误提示，在编辑窗口中将显示源代码错误行。如果产生列表文件，那末将在输出窗口的“List”选项窗中显示。

## (2)、链接

可用生成工具栏中的生成按钮，或用项目菜单中的“Build/Make Project”命令来生成项目。如果没有打开的项目此命令是被禁止的。


当汇编/编译和链接完成后，结果将显示在输出窗口的“Build”选项窗中。如果产生列表文件，文件将显示在输出窗口的“List”选项窗中。

### 2.3.4 集成开发环境与您的目标系统连接

在 IDE 与硬件连接之前，确保，

- a) USB 串行电缆已经连接 PC 机和 UEC5(调试模块)。
- b) 电源已经接到目标硬件
- c) 在 IDE 的 Options 菜单的 Serial Port 子菜单中选定的 USB 接口。
- d) 在 IDE 的 Options 菜单中的 Debug Interface 子菜单中选择的调试接口的正确：
  - ◆ 如果是 C8051F3XX 器件选择“Silicon Labs 2-Wire”。
  - ◆ 如果是 C8051F 系列的其它器件选择“JTAG”。


当所有的硬件已连接，已在 IDE 中选择了串行接口，现在可将 IDE 与硬件连接。可用

生成工具栏中的连接按钮或使用调试菜单中的“Connect”命令来完成连接。如果 IDE 不能访问串行口将报告出错。这可能是由于串口被其它程序占用，如果是这种情况关闭其它应用程序重试连接。

注意：如果其它应用需要使用串口，可以用“Disconnect”命令或生成工具栏中的断开连接

按钮来断开连接。

### 2.3.5 下载代码到 FLASH

简单地按下生成工具栏中的下载按钮或使用 Debug 菜单中的“Download”命令，就可以下载程序到目标处理器的 flash 中（注意：只有在执行“Connect”命令后才能下载代码到目标硬件）。如果在调用下载命令时有项目或文件已打开，相关的目标文件将被下载。如果当前无文件或项目打开，则将弹出一个对话框要求选择需下载的文件。


IDE 下载的文件格式为 Intel Hex 或 OMF-51 格式（默认）。如果下载的文件是 OMF-51 文件并带有调试信息，则 IDE 将打开所有相关的源文件并开始源级调试。这一功能不支持

不带调试信息的十六进制或 OMF-51 文件。

一旦程序被下载，你就可以在目标硬件上调试和运行你的程序，而不是在仿真器上。（一旦程序被下载到目标硬件，所有的调试按钮，如 Go, Stop, 和 Step 等都将允许）。

### 2.3.6 设置断点和观察点

#### (1)、设置断点

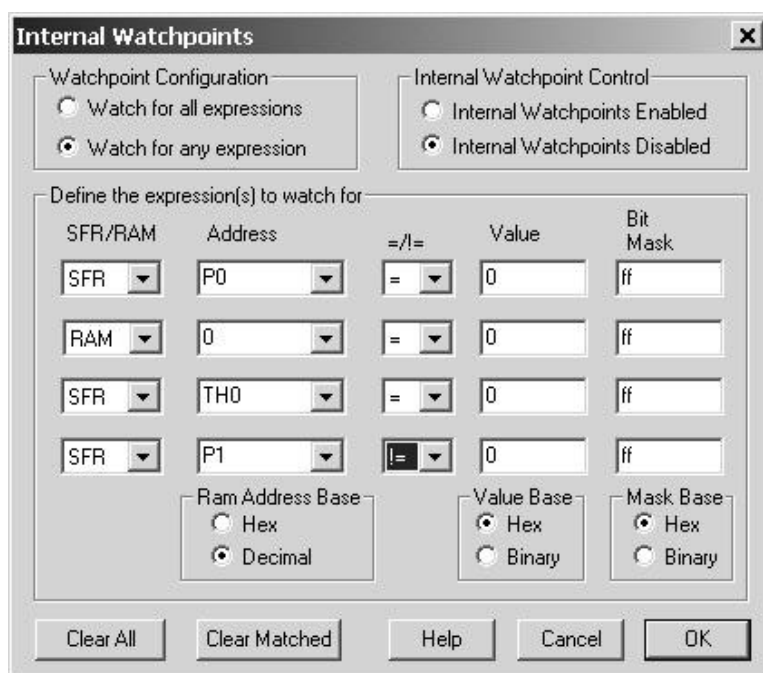
简单地按下工具栏中的断点设置按钮 ，即可在源代码所在的行处设置和取消断点（注意此断点为硬件断点，最多可设置 4 个）。

#### (2) 设置观察点

观察点是由用户设置的软件断点，当设定值在程序运行时匹配是程序停止运行。

- 如果想察看 SFR 存储单元跳到下一步。打开项目窗口的“Symbol”选项窗，找到你想要观察的符号/变量。
- 打开观察点对话框，在四个观察点位置选择 RAM 或 SFR，并复制 RAM 单元到相应的观察点地址框或从地址框中选择 SFR。  
注意：你可以按十六进制或十进制指定 RAM 地址。
- 选择是在与你指定的值相匹配或不相匹配是停止。
- 指定与符号/变量的比较值。注意：你可以按十六进制或二进制指定值。
- 如果你只想观察某些位而忽略其它位，指定要屏蔽的位。逻辑 1 察看，反之逻辑 0 忽略。注意：可以按十六进制或二进制指定屏蔽位。
- 在四个位置重复上述过程加入要观察的变量。
- 在观察点配置框中选择你要观察的是 ANY 或 ALL。如果选择 ANY，当四个观察点中有一个匹配时 IDE 将停止并显示观察点对话框。如果选择 ALL，当所有的四个观察点都匹配时 IDE 将停止并显示观察点对话框。
- 在 Internal Watchpoint Control 中选择 Internal Watchpoints Enabled。
- 点击 OK 按钮，现在你可以单步或运行代码，当观察点匹配时 IDE 将停止运行并显示观察点对话框。
- 一旦遇到匹配值 IDE 停止并显示观察点对话框，为了 IDE 能够继续运行而不会因当前匹配而再次停止，匹配值必须清除或改变。Internal Watchpoints 对话框提供了“Clear All”和“Clear Matched”按钮，使清除更加容易。







Internal Watchpoints 对话框



### 2.3.7 运行和停止

SILICON LABS IDE 包括一个完整的调试器，它运行在实际的产品硬件上,而不是仿真器。

注意：在你使用调试器之前，你的 PC 机必须与 EC2 连接,而 EC2 必须与目标板连接，而且程序代码必须下载到目标处理器 flash 中。

- “Go”  和 “Stop”  按钮？开始和停止目标用户代码执行。
- “Step”  按钮？单步执行代码，一次一条源级指令（包括中断服务程序）。
- 可配置的“Multiple Step”  按钮？执行 N 步
- “Step Over”  按钮越过函数或子程序和“Run to Cursor”  按钮运行到光标处。

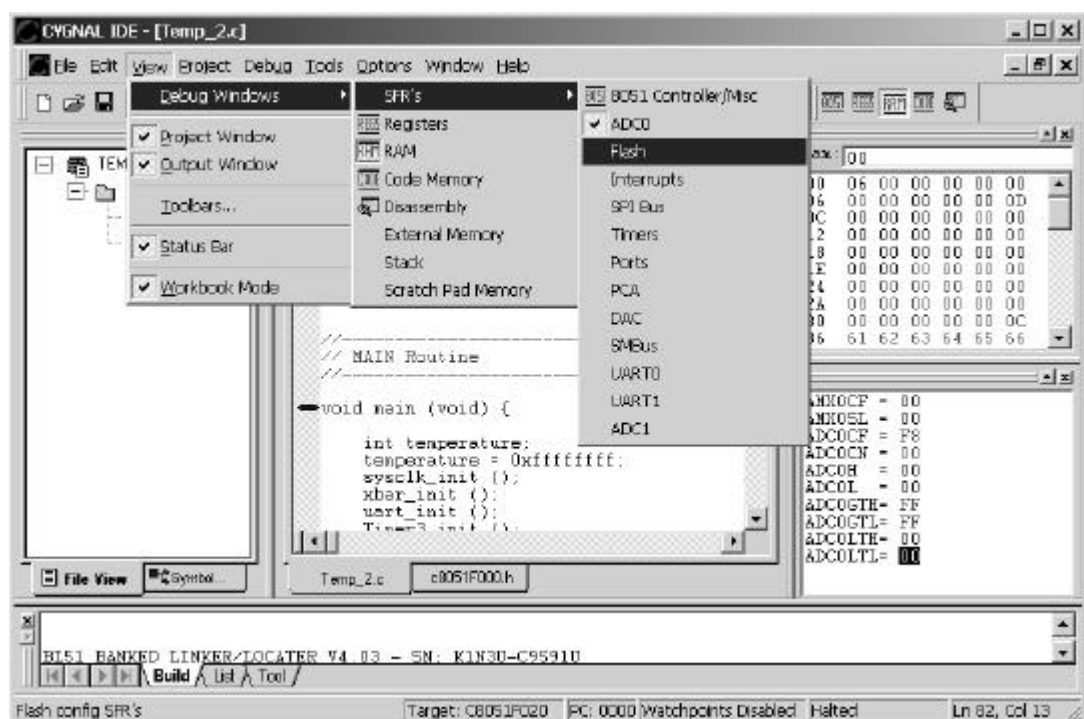
### 2.3.8 查看和修改存储器、寄存器和变量

#### (1) 打开调试窗口

查看和修改存储器、寄存器和变量是在 Debug Windows 中实现的。集成开发环境包含很多调试窗口，在调试期间用它来察看和修改存储器和寄存器的信息。你可以通过“View”菜单的 Debug Windows 来激活调试窗口，也可以通过点击工具栏中的图标按钮激活某些调


试窗口。

下图中显示了如何激活调试窗口。



## (2) 修改存储器和寄存器值

可以在光标处键入数值来修改寄存器原值。修改后的值可以在执行用户代码( 点击"Go"

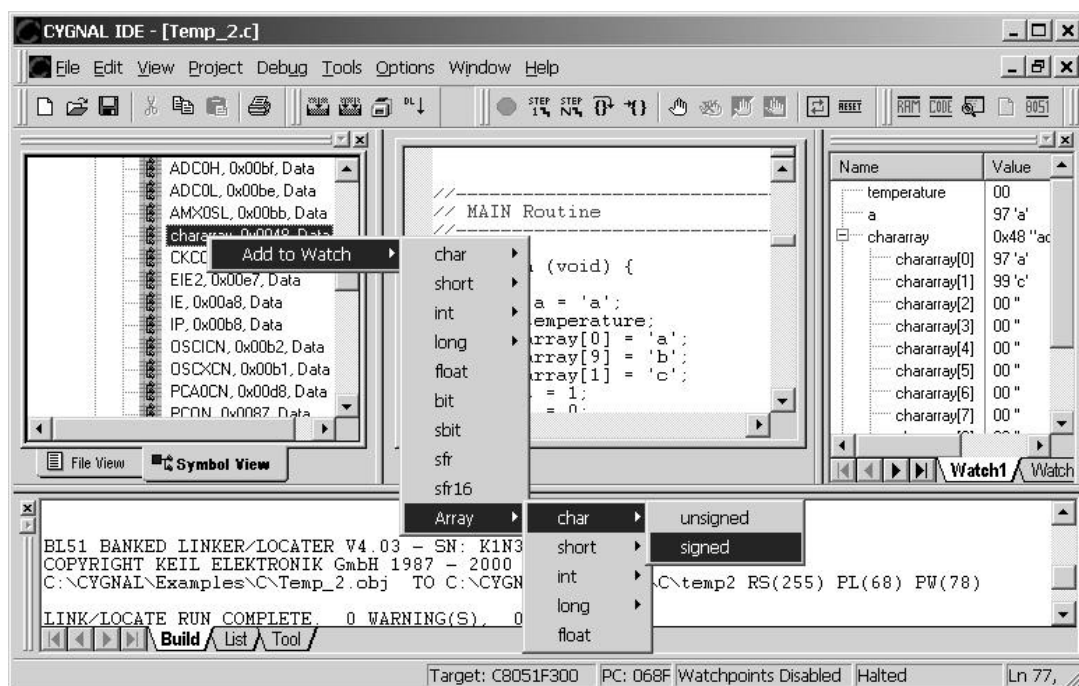
或"Step"按钮) 前下载到硬件。方法是用 " Refresh"按钮  强制写入。这样修改后的值被写入仿真器。寄存器窗口将重读仿真器，窗口将被刷新，所有变化的值以红色显示。

注意：修改寄存器的值只能在调试器处于停止状态时进行。目标处理器正在执行用户代码时不允许写入。

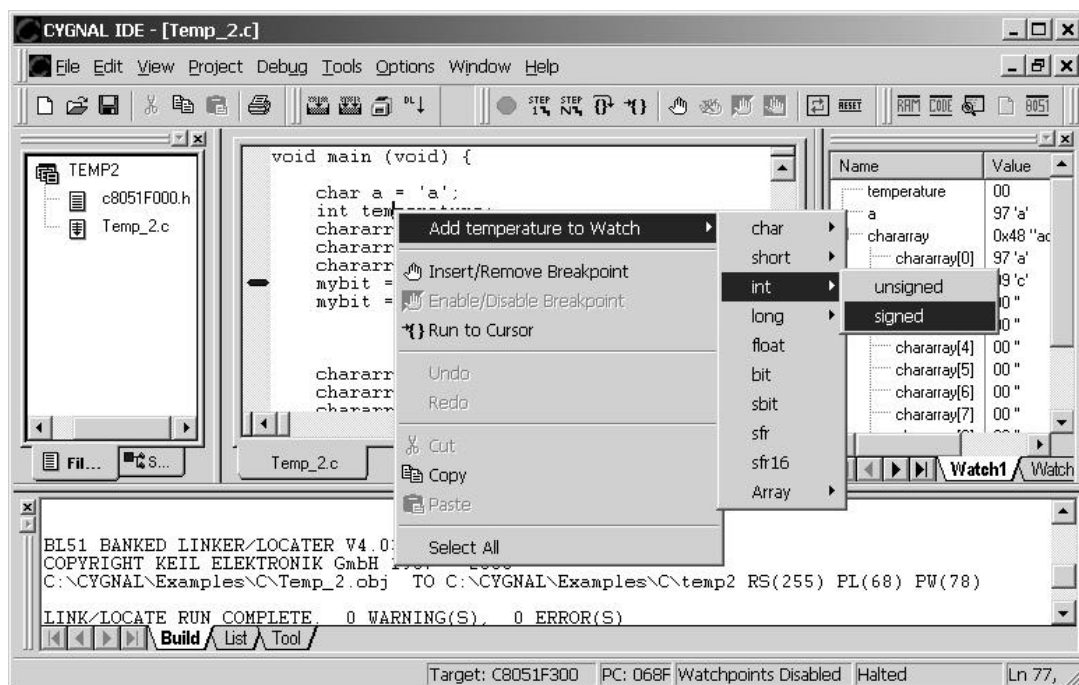
## (3) 如何向观察窗口 ( Watch Window ) 中添加变量

在生成和下载程序代码后可以将要观察的变量加到观察窗口，有两种方法可将变量加到观察窗口：

在符号观察窗口中找到要加入的变量，在变量上点击鼠标右键并选择变量类型。



在源程序代码中找到你要加入到观察窗口的变量，然后在变量上点击鼠标右键。从弹出菜单选择“Add”变量名到观察窗口，并选择变量类型。窗口大小是可调整的，在窗口中删除变量的方法是选定变量然后按下 Delete 键。



第三章 Silicon Labs 单片机实验

3.1 C8051F 单片机 I/O 口交叉开关设置

3.1.1 优先权交叉开关译码器

优先权交叉开关译码器，或称为“交叉开关”，按优先权顺序将端口 0 – 3 的引脚分配给器件上的数字外设（UART、SMBus、PCA、定时器等）。端口引脚的分配顺序是从 P0.0 开始。为数字外设分配端口引脚的优先权顺序列于下图。

优先权交叉开关译码表（EMIFLE = 0；P1MDIN = 0xFF）

	P0							P1							P2							P3							交叉开关寄存器位						
引脚 I/O	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7			
TX0																																	UART0EN:XBR0.2		
RX0																																			UART0EN:XBR0.2
SCK																																			SPI0EN:XBR0.1
MISO																																			
MOSI																																			
NSS																																			
SDA																																			
SCL																																			SMB0EN:XBR0.0
TX1																																			UART1EN:XBR2.2
RX1																																			
CEX0																																			PCA0ME:XBR0.[5:3]
CEX1																																			
CEX2																																			
CEX3																																			
CEX4																																			
ECI																																			ECI0E:XBR0.6
CP0																																			CP0E:XBR0.7
CP1																																			CP1E:XBR1.0
T0																																			T0E:XBR1.1
/INT0																																			INT0E:XBR1.2
T1																																			T1E:XBR1.3
/INT1																																			INT1E:XBR1.4
T2																																			T2E:XBR1.5
T2EX																																			T2EXE:XBR1.6
T4																																			T4E:XBR2.3
T4EX																																			T4EXE:XBR2.4
/SYSCLK																																			SYSCKE:XBR1.7
CNVSTR																																			CNVSTE:XBR2.0
						ALE											AIN1.0/A8																		
						/RD											AIN1.1/A9																		
						/WR											AIN1.2/A10																		
																	AIN1.3/A11																		
																	AIN1.4/A12																		
																	AIN1.5/A13																		
																	AIN1.6/A14																		
																	AIN1.7/A15																		
																	A8m/A0																		
																	A9m/A1																		
																	A10m/A2																		
																	A11m/A3																		
																	A12m/A4																		
																	A13m/A5																		
																	A14m/A6																		
																	A15m/A7																		
																	AD0/D0																		
																	AD1/D1																		
																	AD2/D2																		
																	AD3/D3																		
																	AD4/D4																		
																	AD5/D5																		
																	AD6/D6																		
																	AD7/D7																		
	AIN1 输入/非复用地址高								复用地址高/非复用地址低								复用数据/非复用数据																		

### 3.1.2 交叉开关引脚分配

当交叉开关配置寄存器 XBR0、XBR1 和 XBR2 中外设的对应允许位被设置为逻辑‘1’时，交叉开关将端口引脚分配给外设，相关的特殊功能寄存器的定义见数据手册或相关书籍。给 IO 口分配数据外设的方法有两种：第一，按照“优先权交叉开关译码表”并参考相关特殊功能寄存器的值来定义(参考下面的例子)。另一种方法是用配置向导，在 Silicon Labs IDE 中，选择 Tools->Silicon Labs Configuration Wizard，进入配置向导界面进行 IO 口配置。

#### 交叉开关引脚分配示例

在本例中，我们将配置交叉开关，为 UART0、SMBus、UART1、/INT0 和 /INT1 分配端口引脚（共 8 个引脚）。另外，我们将外部存储器接口配置为复用方式并使用低端口。我们还将 P1.2、P1.3 和 P1.4 配置为模拟输入，以使用 ADC1 测量加在这些引脚上的电压。配置步骤如下：

1. 按 UART0EN = 1、SMB0EN = 1、INT0E = 1、INT1E = 1 和 EMIFLE = 1 设置 XBR0、XBR1 和 XBR2，则有：XBR0 = 0x05，XBR1 = 0x14，XBR2 = 0x02。
2. 将外部存储器接口配置为复用方式并使用低端口，有：PRTSEL = 0，EMD2 = 0。
3. 将作为模拟输入的端口 1 引脚配置为模拟输入方式：设置 P1MDIN 为 0xE3（P1.4、P1.3 和 P1.2 为模拟输入，所以它们的对应 P1MDIN 被设置为逻辑‘0’）。
4. 设置 XBARE = 1 以允许交叉开关：XBR2 = 0x42。
  - UART0 有最高优先权，所以 P0.0 被分配给 TX0，P0.1 被分配给 RX0。
  - SMBus 的优先权次之，所以 P0.2 被分配给 SDA，P0.3 被分配给 SCL。
  - 接下来是 UART1，所以 P0.4 被分配给 TX1。由于外部存储器接口选在低端口（EMIFLE = 1），所以交叉开关跳过 P0.6(/RD)和 P0.7(/WR)。又因为外部存储器接口被配置为复用方式，所以交叉开关也跳过 P0.5(ALE)。下一个未被跳过的引脚 P1.0 被分配给 RX1。
  - 接下来是 /INT0，被分配到引脚 P1.1。
  - 将 P1MDIN 设置为 0xE3，使 P1.2、P1.3 和 P1.4 被配置为模拟输入，导致交叉开关跳过这些引脚。
  - 下面优先权高的是 /INT1，所以下一个未跳过的引脚 P1.5 被分配给 /INT1。
  - 在执行对片外操作的 MOVX 指令期间，外部存储器接口将驱动端口 2 和端口 3。
5. 我们将 UART0 的 TX 引脚（TX0，P0.0）、UART1 的 TX 引脚（TX1，P0.4）、ALE、/RD、/WR（P0.[7:3]）的输出设置为推挽方式，通过设置 P0MDOUT = 0xF1 来实现。
6. 我们通过设置 P2MDOUT = 0xFF 和 P3MDOUT = 0xFF 将 EMIF 端口（P2、P3）的输出方式配置为推挽方式。
7. 我们通过设置 P1MDOUT = 0x00（配置输出为漏极开路）和 P1 = 0xFF（逻辑‘1’，选择高阻态）禁止 3 个模拟输入引脚的输出驱动器。



## 3.2 在 C8051F020 单片机上的初级软件实验(汇编语言)

### 3.2.1 无符号数、十进制加法实验

(1) 实验目的 掌握汇编语言设计和调试方法,熟悉在 SILICON LABS IDE 环境下观察内部 RAM 的方法。

(2) 实验内容 编写并调试一个双字节无符号十进制数加法程序,其功能为将被加数写入 40H、41H 单元,加数写入 51H,50H 单元,运行程序结果写入 52H,51H,50H 单元中,则加法程序功能为:

$$(41H)(40H)+(51H)(52H) = (52H)(51H)(50H)$$

#### (3) 实验参考程序

```
DATA0 EQU 12H
DATA1 EQU 34H
DATA2 EQU 56H
DATA3 EQU 78H
ORG 0000H
AJMP 100H
ORG 0100H
MOV 40H,#DATA0
MOV 41H,#DATA1      ;被加数送 41H,40H
MOV 50H,#DATA2
MOV 51H,#DATA3      ;加数送 51H,50H
MOV A,40H
ADD A,50H            ;(40H)+(50H)  A
DA A
MOV 50H,A            ;保存结果
MOV A,41H
ADDC A,51H           ;(41H)+(51H)+CY  A
DA A
MOV 51H,A            ;保存结果
MOV 52H,#0
MOV A,#0
ADDC A,52H
MOV 52H,A            ;进位  52H
LOOP0:AJMP LOOP0
END
```

#### (4) 调试方法

按前面介绍的方法建立一个新的项目文件,将参考程序加到这个项目文件中,编译连接后,将程序下载到 C8051F020 中。打开 RAM 观察窗口,运行程序。

运行程序时,可单步执行或在送数指令后加断点观察 RAM 存储单元的变化,观察加法结果是否正确。

### 3.2.2 数据传送实验

(1) 实验目的 掌握对 C8051F020 内部 RAM 及外部 RAM 的数据操作，掌握 C8051F020 片内的外扩 RAM 的操作方法。

(2) 实验内容 本例程将内部 RAM 70H~7FH 的 16 个数据送到外部 RAM 7000H~700FH

#### (3) 实验参考程序

```
ORG 0000H
AJMP MAIN
ORG 0100H
MAIN: MOV R7,#16
      MOV R1,#70H           ;内部 RAM 地址 70H 送 R1
      MOV DPTR,#7000H       ;DPTR 指向 XRAM 地址 7000H
LOOP: MOV A,@R1             ;内部 RAM 内容送累加器 A
      MOVX @DPTR,A          ;将累加器 A 的值送到 XRAM
      INC DPTR
      INC R1
      DJNZ R7,LOOP          ;判断数据是否送完
      HERE:AJMP HERE
      END
```

#### (4) 调试方法

可单步执行可设断点执行程序。

打开 RAM 及 EXTERRANL MEMORY 窗口，观察相对应的数据区的数据是否一致。

### 3.2.3 数据排序实验

(1) 实验目的 熟悉 MCS-51 指令系统，掌握排序程序的设计方法。

(2) 实验内容 本例程采用交换排序法将内部 RAM 中的 50~59H 单元中的 10 个单字节无符号二进制数按从小到大的次序排列，并将这一列排序后的数据从小到大依次存贮到外部 RAM 1000H 开始处。

#### (3) 参考程序

```
ORG 0000H
JMP MAIN
ORG 0100H
MAIN: MOV R0,#50H
      MOV @R0,#5FH
      INC R0
      MOV @R0,#56H
      INC R0
      MOV @R0,#5AH
      INC R0
      MOV @R0,#5EH
      INC R0
```

```
        MOV @R0,#51H
        INC R0
        MOV @R0,#5BH
        INC R0
        MOV @R0,#53H
        INC R0
        MOV @R0,#58H
        INC R0
        MOV @R0,#57H
        INC R0
        MOV @R0,#55H    ;将 10 个随机数送入内部 RAM 的 50~59H 单元
        NOP              ;可在此处设置断点
        ACALL QUE        ;调用排序子程序
OUT:    MOV R0,#50H
        MOV DPTR,#1000H
        MOV R7,#10
OUT1:   MOV A,@R0
        MOVX @DPTR,A
        INC R0
        INC DPTR
        DJNZ R7,OUT1
HERE:   AJMP HERE
QUE:    CLR 00H          ;清交换标志
        MOV R1,#50H
        MOV R6,#09H
I3:     MOV A,R6
        MOV R7,A
        MOV A,R1
        MOV R0,A
        MOV A,@R0
I2:     INC R0
        MOV R2,A
        SUBB A,@R0
        MOV A,R2
        JC I1
        SETB 00H
        XCH A,@R0
I1:     DJNZ R7,I2
        NOP              ;可在此处设置断点观察每次排序结果
        JNB 00H,STOP
```

```

MOV @R1,A
INC R1
DJNZ R6,I3
STOP: RET
END

```

## (2) 调试方法

可在程序中的指令 NOP 处设置断点，在第一个断点处可观察 50~59H 单元内容是否为 10 个任意排列原始数据。

在第二个断点处可观察每次排序的结果。

可单步执行程序观察排序过程。

## 3.2.4 整数二翻十实验

(1) 实验目的 掌握计算机数制转换的方法，熟悉 C8051F020 汇编程序的设置方法。

(2) 实验内容 本例程将键盘输入的 1 位十六进制数转换为十进制数，并将转换结果在七段码显示器上显示出来。在数码管上显示二位十进制数，不足二位十位显示 0。显示时用数码管 1。数码管 2 显示，同时将前一显示数据左移。

## (3) 实验参考程序

```

$INCLUDE(C8051F020.INC)
BIT_COUNT DATA 07FH
TIMER DATA 07EH
TIMER1 DATA 07DH
TEN DATA 07CH
DATA_IN DATA 020H
DATA_OUT DATA 021H
CLK BIT P1.6
DAT BIT P1.7

ORG 0000H
JMP START
ORG 0100H
START: MOV WDTCN, #0DEH ; DISABLE WATCHDOG TIMER
MOV WDTCN, #0ADH
MOV OSCXCN, #67H ; ENABLE EXTERNAL CRYSTAL
; OSCILLATOR AT 22.1184MHZ
CLR A ; WAIT AT LEAST 1MS
DJNZ ACC, $ ; WAIT ~512US
DJNZ ACC, $ ; WAIT ~512US

OSC_WAIT: ; POLL FOR XTALVLD-->1
MOV A, OSCXCN
JNB ACC.7, OSC_WAIT

```

```

        ORL    OSC1CN, #08H          ; SELECT EXTERNAL OSCILLATOR AS
                                     ; SYSTEM CLOCK SOURCE
        ORL    OSCXCN, #80H          ; ENABLE MISSING CLOCK DETECTOR

CP1INIT:  MOV CPT1CN, #080H          ; COMPARATOR 1 CONTROL REGISTER
        MOV A, #010H                ;CPT1CN
        DJNZ ACC, $                  ;CPT1CN WAIT AT LEAST 20US
        ANL CPT1CN, #NOT(30H)
        MOV EIP1, #040H
        MOV TIMER,#50

PORTINIT: MOV XBR2, #044H
        MOV P0MDOUT, #001H

DELAY0:   MOV TIMER1,#255
DELAY1:   DJNZ TIMER1,DELAY1
        DJNZ TIMER,DELAY0
        ANL P5,#00H
        MOV DATA_OUT,#10100100B
        CALL SEND
        ORL P5,#80H

MAIN:     MOV A,CPT1CN                ;等待有键按下,发送读键盘指令
        ANL A,#40H
        JNZ MAIN
        MOV DATA_OUT,#00010101B ;
        CALL SEND
        CALL RECEIVE
        ANL P5,#80H
        MOV B,#10
        MOV A,DATA_IN
        DIV AB
        MOV TEN,A
        MOV DATA_OUT,#10100001B
        CALL SEND
        MOV DATA_OUT,#10100001B
        CALL SEND
        MOV DATA_OUT,#10000001B
        CALL SEND
        MOV DATA_OUT,TEN
        CALL SEND
        MOV DATA_OUT,#10000000B
        CALL SEND
```

```
        MOV DATA_OUT,B
        CALL SEND
        ANL P5,#80H
WAIT:    MOV A,CPT1CN                ;等待按键放开
        ANL A,#40H
        ORL A,#00H
        JZ WAIT
        JMP MAIN
SEND:    MOV BIT_COUNT,#8           ;发送字符子程序
        ANL P5,#00H
        CALL LONG_DELAY
SEND_LOOP:MOV C,DATA_OUT.7
        MOV DAT,C
        SETB CLK
        MOV A,DATA_OUT
        RL A
        MOV DATA_OUT,A
        CALL SHORT_DELAY
        CLR CLK
        CALL SHORT_DELAY
        DJNZ BIT_COUNT,SEND_LOOP
        CLR DAT
        RET
RECEIVE:MOV BIT_COUNT,#8           ;接收字符子程序
        SETB DAT
        CALL LONG_DELAY
RECEIVE_LOOP:SETB CLK
        CALL LONG_DELAY
        MOV A,DATA_IN
        RL A
        MOV DATA_IN,A
        MOV C,DAT
        MOV DATA_IN.0,C
        CLR CLK
        CALL SHORT_DELAY
        DJNZ BIT_COUNT,RECEIVE_LOOP
        CLR DAT
        RET
LONG_DELAY: MOV TIMER,#150         ;延时约 200US
DELAY_LOOP: DJNZ TIMER,DELAY_LOOP
```

```

      RET
SHORT_DELAY: MOV TIMER,#20          ;延时约 20US
SHORT_LP:    DJNZ TIMER,SHORT_LP
      RET
      END

```

#### (4) 实验程序参考框图

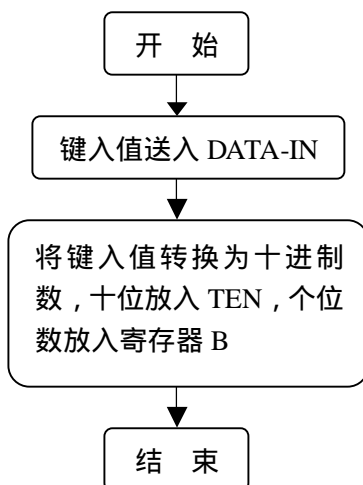


图 3.2.4 整数二翻十实验参考框图

#### (5) 调试方法

在读取键值后设置断点，观察 DATA\_IN 的值是否与键入值相等。  
全速运行程序，观察数码管显示的数值与键入值的关系。

#### 3.2.5 查表实验

##### (1) 实验目的

熟悉 MCS-51 的查表指令功能和使用方法，掌握查表程序的设计和调试。

##### (2) 实验内容

本例程的功能为读键盘输入的 0~7 数字键，通过查表获得相应的数值并在七段码显示器上显示出来。

数字键'0'----8000H	数字键'1'----8001H
数字键'2'----8002H	数字键'3'----8003H
数字键'4'----8004H	数字键'5'----8005H
数字键'6'----8006H	数字键'7'----8007H

##### (3) 实验程序参考框图

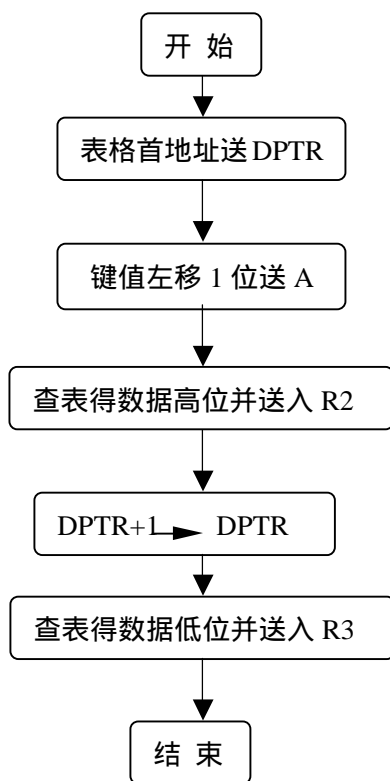


图 3.2.5 查表程序参考图

#### (4) 调试方法

在读取键后设置断点，观察 DATA\_IN 的内容与键入值是否相同。

在调用查表程序后设置断点，观察 R2，R3 的内容与表中的内容是否相同。

全速运行程序，键入数字键，观察显示器上的内容是否为键值对应的数值。

若有错误，应采用单步可断点分段调试。

### 3.3 I/O 口实验 (C 语言)

#### 3.3.1 LED 驱动实验

##### (1)、实验目的

本实验通过 I/O 口来驱动 LED 的亮灭

##### (2)、实验说明

本实验通过 C8051F020 的 P4 口驱动 8 盏 LED 灯，实现 8 盏灯依次循环点亮，实验前用跳线将 LD1~LD7 分别同端口 P4.0~P4.7 相连。

LD1~LD7 的正极通过排阻(RP10)上拉到电源，要点亮哪个 LED，只需将相连的 P4x 置成低电平。

##### (3) 实验程序参考框图



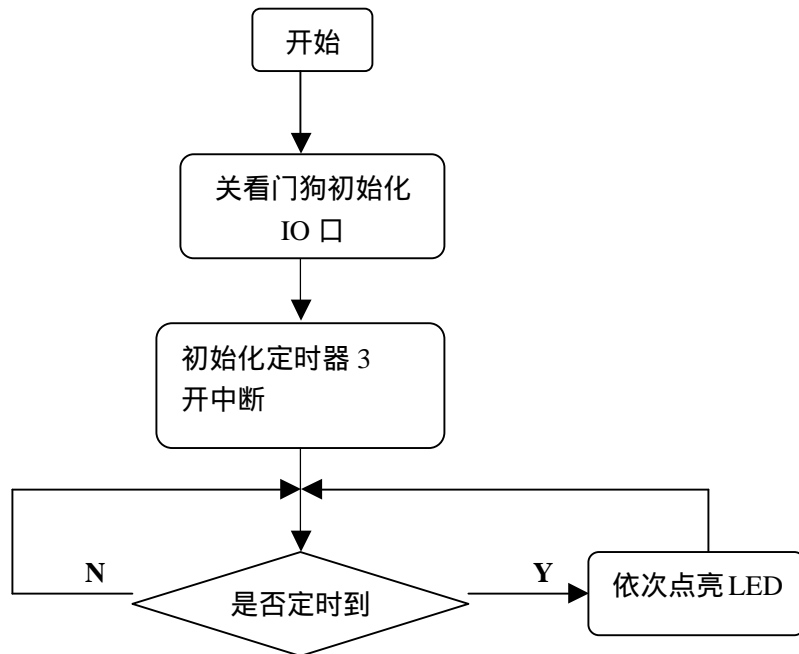


图 3.3.1 LED 驱动程序流程图

#### (4) 调试方法

运行程序,观察 8 盏 LED 灯是否在定时中断下依次循环点亮,若有错可单步执行,排除程序错误。

全速运行程序,实现所要求的显示功能。

### 3.3.2 光隔/继电器驱动实验

#### (1)、实验目的

熟悉光隔/继电器的工作原理,通过 I/O 口控制其导通与截止。

#### (2)、实验说明

实验前用跳线将 P55 与 RELAY, P56 与 PHOT0, P57 与 PHOT1 分别短接,作为光耦/继电器的控制脚, J19 跳接 1, 2 脚,使能 VCC 作为继电器的工作电源。

本实验例程通过 C8051F020 的 P5 口控制光耦/继电器控制脚的导通和截止。

继电器驱动器通过非门(U7E)和与非门(U26B)驱动继电器工作, D10 是续流二极管,是为配合继电器线圈放电用的,当继电器线圈断电时,线圈要放电,用续流二极管导流。

#### (3)、实验程序参考框图

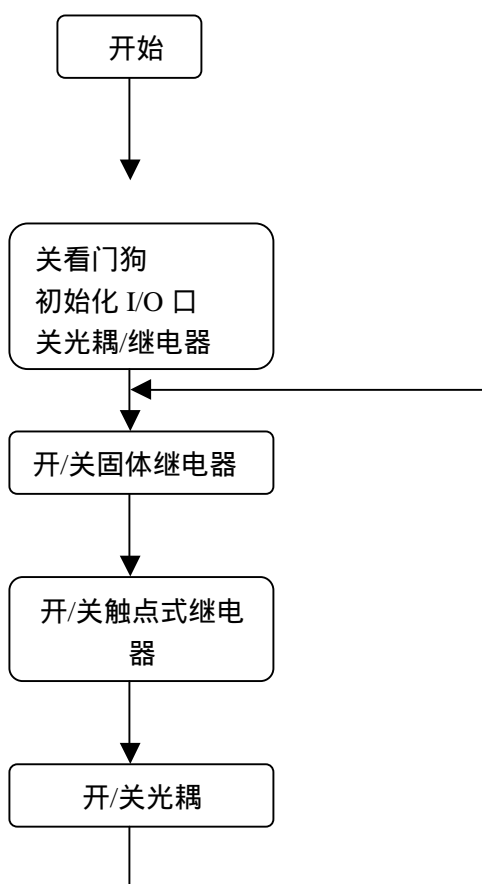


图 3.3.2 光耦/继电器驱动流程图

#### (4)、调试方法

将断点分别断在光耦/继电器的开/关语句的后面，当程序运行到断点处，其中固体继电器的通断可以用万用表测量 JP1 两端的阻值变化来判定（断开时阻值无穷大，导通时有一个小的阻值）；同理光耦的通断可以测量 JP2 两端的阻值来判定；而触点式继电器的通断可以用 D7 和 D13 两 LED 的交替点亮来判定，并伴有继电器通电/断电的吸合声。

### 3.4 串行总线通信实验

#### 3.4.1 RS-232 串行通信实验

##### (1) 实验目的

掌握 RS-232 串口通信程序的设计和操作系统。

##### (2) 实验说明

此例程是用 UART0 作为串口通信，软件将其配置到 P0.0 和 P0.1，实验前将 P0.0 与 TX0, P0.1 与 RX0 用跳线短接。

J16 是串口的外接口，为进行串口通信，在 PC 端用串口调试助手软件配合调试（利用其它串口通信程序也可以），用一根串口线将 J16 同 PC 的串口相连，波特率固定为 9600，通信格式：1 位起始位，8 位数据位，1 位停止位。

U12 是芯片 MAX202，它将单片机发出的 TTL 电平转化为 RS-232 电平信号，收/发信号从 9 芯插座 J16 通过 RS-232 电缆传送到 PC 机。

### （3）实验程序参考框图

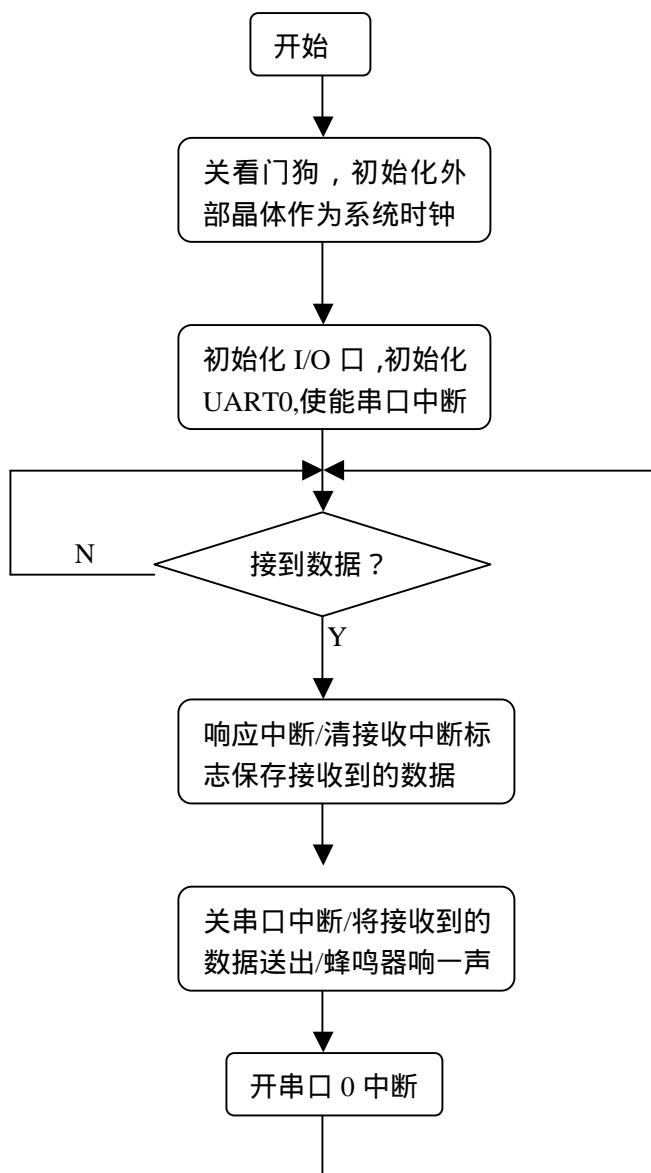


图 3.4.1 串行总线通信流程图

#### (4) 调试方法

在 PC 端运行串口调试助手程序,设置好通信的数据格式,此测试程序用固定的 9600 波特率从串口调试助手发出一个数据,实验板的串口电路处在接收状态,当有数据接收则进入串口中断,经过处理将接收的此数据又发出,并伴随蜂鸣器响一声,在串口调试助手的接收窗口也将显示此数据。

### 3.4.2 RS-485 串口通信实验

(1) 实验目的 利用 C8051F020 串口 UART1 的 TXD.RXD 口,学习 RS-485 差分串行接口的使用。

#### (2) 实验说明

此例程是用 UART1 的 TXD,RXD 作为 RS-485 差分串行接口,软件将此两脚配置到 P0.0 和 P0.1,实验前将 P0.0 与 TX1,P0.1 与 RX1 用跳线短接。

JP3 是 RS-485 的外接口,为进行 RS485 通信,在 PC 端用串口调试助手软件配合调试(利用其它串口通信程序也可以),用一串口转 RS485 的转接头将 JP3 同 PC 的串口相连,注意 JP3 的 A 和 B 分别同转接头 RS485 端的 A 和 B 相连,波特率固定为 9600,通信格式:1 位起始位,8 位数据位,1 位停止位。

U13 是芯片 MAX485,它将单片机发出的 TTL 电平转化为 RS-485 电平信号,其中收发方向由 MCU 的 I/O 口 P32 控制,收/发信号从 2 芯插座 JP3 通过 RS-485 电缆传送到其它 RS-485 总线设备。

#### (3、实验程序参考框图

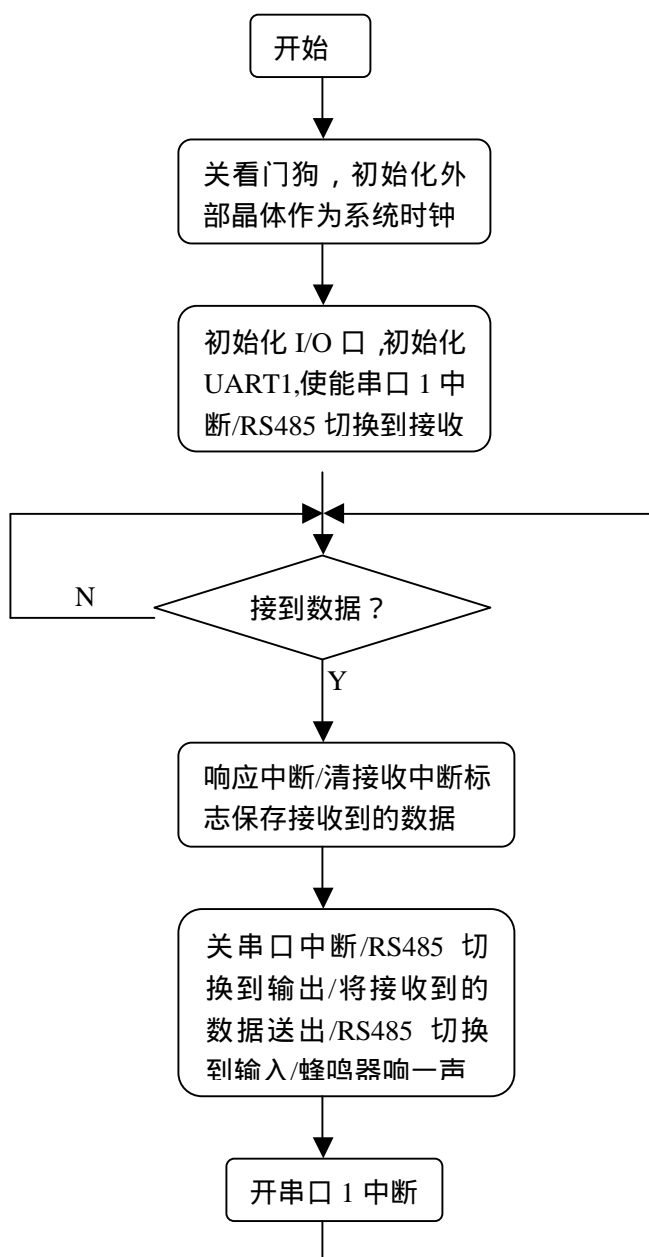


图 3.4.2 RS-485 串口通信流程图

#### (4) 调试方法

在 PC 端运行串口调试助手程序，设置好通信的数据格式，此测试程序用固定的 9600 波特率，从串口调试助手发出一个数据，实验板如果接收正确，将接收的此数据再发出，

并伴随蜂鸣器响一声，在串口调试助手的接收窗口也将显示此数据。

### 3.5 并行总线实验

#### 3.5.1 6 位数码管显示实验

(1)、实验目的 学习 LED 数码管显示器与并行接口扩展电路的设计方法，练习编程，调试的工作过程。

(2)、实验说明

此实验通过 8255 扩展 I/O 芯片作为驱动 7 断数码管的信号，其中 PB 和 PC 配置成 2 个 8 位输出口，74LS244 作为 LED 数码管的 7 段显示驱动，6 位数码管采用共阴极方式，PB 口作为 6 个数码管的位选，ULN2003 是增加驱动能力，PC 口输出 7 段字形码数据输出。

依据电路所示，A 口，B 口，C 口，控制口的地址分别是：  
0x8000,0x8001,0x8002,0x8003.

共阴极 7 段 LED 数码管的显示字形编码表如下：

显示数字：	1	2	3	4	5	6	7	8	9	0	.
字形编码：	0x06	0x5b	0x4f	0x66	0x6d	0x7d	0x07	0x7f	0x6f	0x3f	0x80

(3)、实验程序参考框图

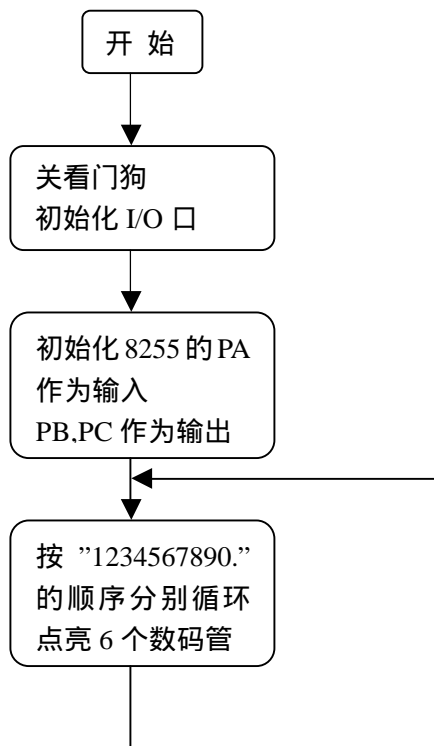


图 3.5.1 6 位数码管显示流程图

(4)、调试方法

运行此程序，将按 ‘1’→‘2’→‘3’→‘4’→‘5’→‘6’→‘7’→‘8’→‘9’→‘0’→‘.’的顺序循环点亮 6 位数码管。

### 3.5.2 键盘实验

(1)、实验目的 熟悉动态扫描式键盘工作原理。

(2)、实验说明

此实验通过 8255 扩展 I/O 芯片实现键盘功能，其中 PA 作为键盘扫描的输入，PB 作为键盘扫描的输出。

其中键值 F1~F4 不是用中断方式，而是用查询方式 (PA4~PA7)，其它键值采用扫描方式工作，不断从 PB0~PB3 口依次循环输出低电平，然后从 PA 口读，如果没有键按下，PA 口在上拉电阻的作用下读到的数据始终是高电平 (0xFF)，如果有按键按下，读到的数据是一个非 0xFF 的值，依据 PB 的输出值和 PA 读入值就可判定哪个按键按下。

(3)、实验程序参考框图

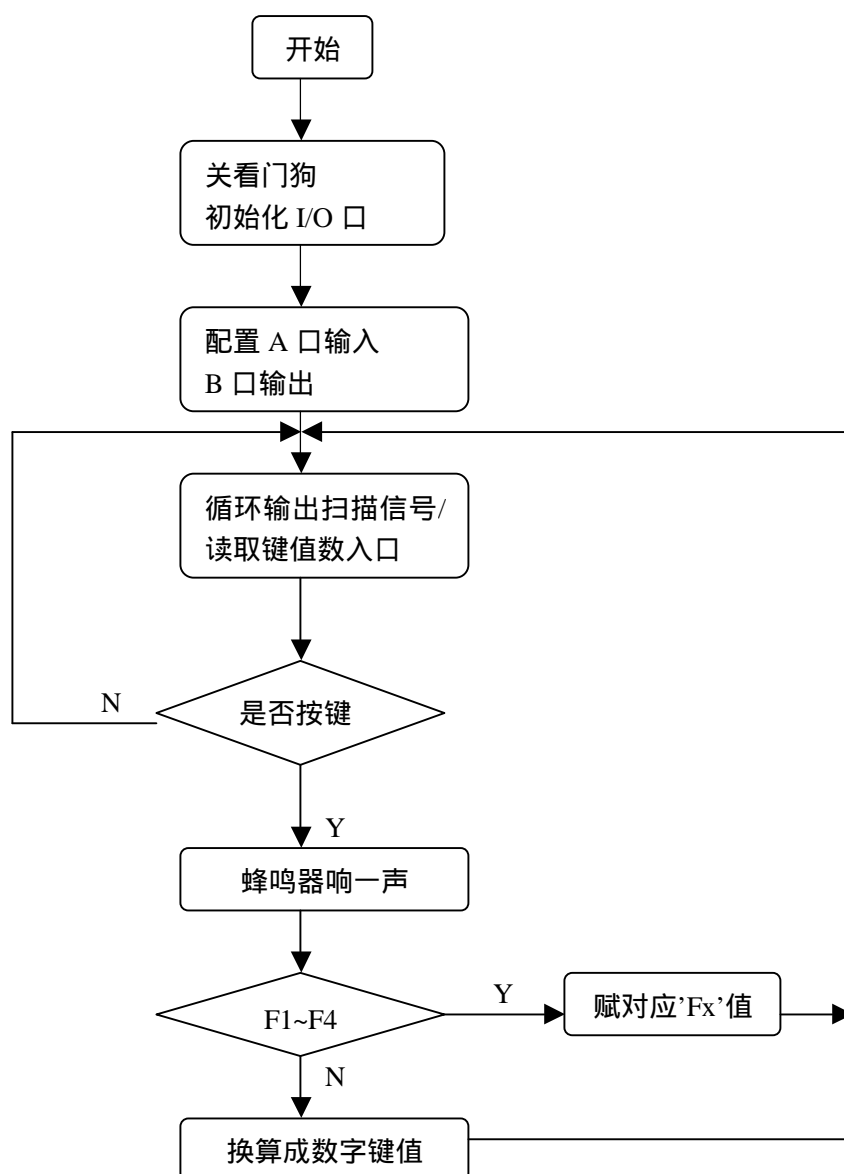


图 3.5.2 键盘流程图

#### (4)、调试方法

将断点设在例程两个\_nop\_()语句上，全速运行程序，当有按键按下时蜂鸣器响一声，程序终止在断点上，在变量观察窗口上（事先打开变量观察窗口，将变量 temp 加入）读取临时变量 temp 的值，看是否是所按下键的键值，如果正确再全速运行程序，循环上述的过程判定所有的键是否正确。

### 3.5.3 实时时钟芯片 DS12887 应用实验

#### (1)、实验目的

掌握实时时钟器件 DS12887 的工作原理及编程方法，与数码管、液晶显示器组成的系统，实现电子时钟。

#### (2)、实验说明

MOT 为模式选择端。MOT 接地时选择 INTEL 模式；MOT 接 VCC 时选择 MOTOROLA 模式。

AD0~AD7 为复用数据/地址线，在进行读/写操作时，AS 端的上升沿把出现在总线上的地址信号锁存在 DS12887 中，应将 AS 端接单片机的 ALE。

/CS 为 DS12887 的片选信号，本教学实验机的片选地址为 83xxH。

DS12887 相关寄存器的说明参考它的数据手册。

#### (3)、实验程序参考框图



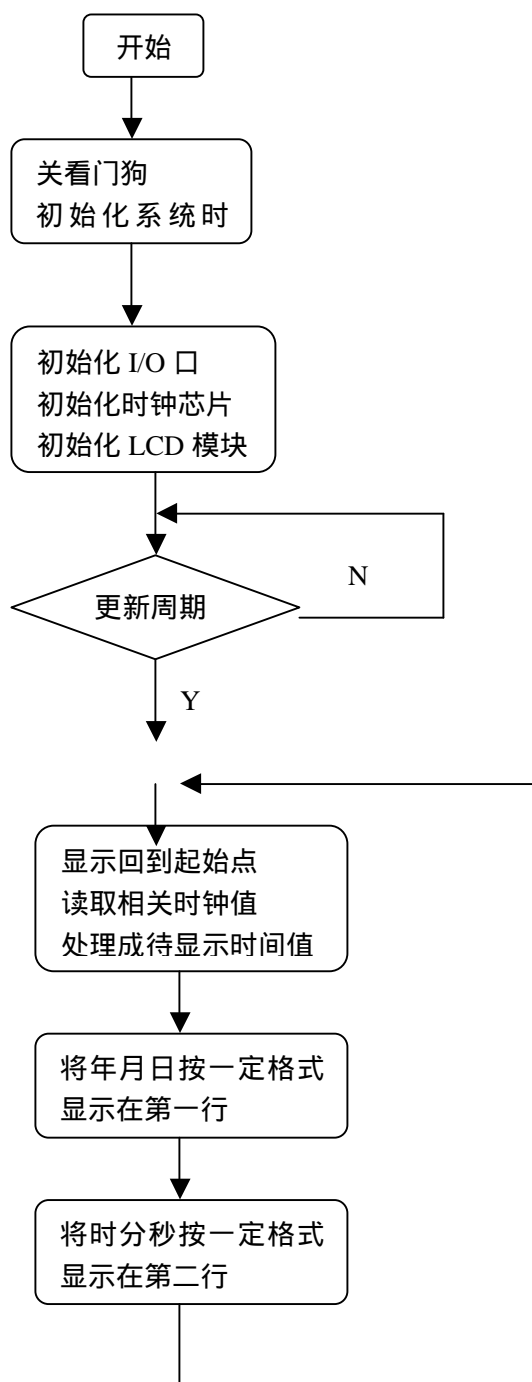


图 3.5.3 实时时钟芯片 DS12887 应用流程图

#### (4)、调试方法

程序运行后初始化的日期是：05 年 07 月 01 日，时间是：23 时 58 分 30 秒，如果程

序运行成功，日期/时间值将显示在液晶屏上，且时间值不断在液晶显示屏上刷新显示。

适当修改时钟初始化子函数的时间值，判定 DS12887 也能正常工作并显示。

同外部精准时钟对比，看 DS12887 的时间是否计时准确。

### 3.5.4 外部扩展 SRAM 实验

#### (1)、实验目的

学习设计存储器扩展电路的方法，熟悉读/写数据存储器的程序设计与实验操作

#### (2)、实验说明

74HC573(U18)用于低 8 位地址锁存，62256 (U15) 是 32K 的数据存储器，有 8 条数据线 and 15 条地址线，通过读，写控制单片机与 62256 之间数据传送的方向。A15 用作片选线，当 A15=0 时，选择 62256 可对它进行读/写。62256 的地址空间范围为 0000H~7FFFH; 当 A15=1 时，选择外部接口芯片，其外设端口地址是：8000H~8700H,实验前 将 J3 用跳线接到 2, 3 脚。

#### (3)、实验程序参考框图

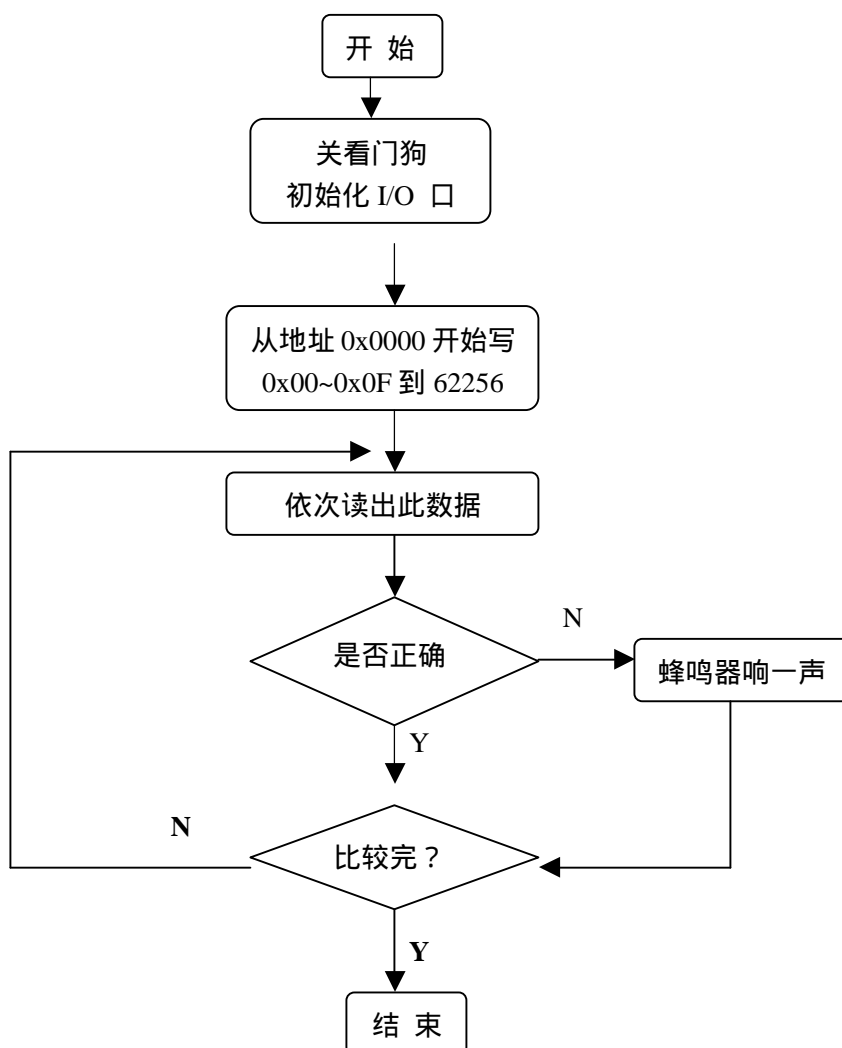


图 3.5.4 外部扩展 SRAM 流程图

(4)、调试方法

看懂实验电路原理图，结合片外 RAM 的读/写指令，理解单片机读/写 RAM 的时序电路。

运行例程判定运行结果是否正确。

改变例程的读/写地址和读/写字节的长度，看程序是否能正常工作。

3.5.5 1602 字符型 LCD 显示实验

(1)、实验目的

通过液晶显示器接口电路的设计和编程，掌握液晶显示器的原理，学会使用液晶显示器，并把它嵌入到便携式电子信息产品中。

(2)、实验说明

液晶显示器模块电路可直接与单片机地 I/O 口或扩展 I/O 口连接，也可挂接在单片机的总线上，用读/写信号实现数据传输，本教学实验机用的是 I/O 驱动。

液晶显示器模块的 RS 接 A10,R/W 接 A9,E 接 A8。

LCD 模块读/写格式

RS(A10)	R/W(A9)	读/写地址	操作功能
0	0	00H	写指令寄存器
0	1	01H	读出 BF 信号和地址计数器 AC 到 D0~D7
1	0	02H	写数据寄存器
1	1	03H	读数据寄存器

液晶显示模块对比度受第三脚的电压控制，通过适当调节其外接的两分压电阻（R1 和 R9）可得到较好的显示效果。

实验前用跳线将 J1 的 Pin1 和 Pin2 短接，使液晶显示模块的背光点亮。74LS00(U5)与非门的作用是增强信号线的驱动能力。

(3)、实验程序参考框图

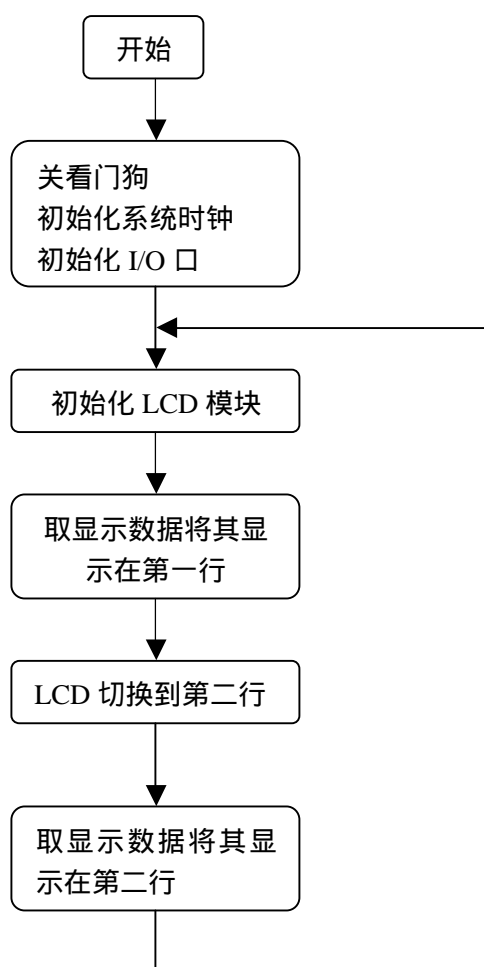


图 3.5.5 1602 字符型 LCD 显示流程图

#### (4)、调试方法

将例程下载到芯片运行，第一行是否会显示“Welcom to NCD”，第二行是否会显示“www.com.cn”。

如果程序显示正确，在此例程的基础上，参考 LCM 的数据手册对其相关命令作些测试，进一步熟悉液晶显示模块。

### 3.6 电机驱动实验

#### 3.6.1 步进电机驱动实验

##### (1)、实验目的

了解步进电机工作原理，掌握使用 C8051F020 控制步进电机的硬件设计方法，熟悉步进电机驱动程序的设计与调试。

## (2)、实验说明

步进电机的工作原理:

以三相反应式步进电机为例:它的定子上有三对磁极,每一对磁极上绕着一相绕组,绕组通电时,这两个磁极的极性相反;三相绕组接成星形,转子铁心及定子极靴上有小齿,定转子齿距通常相等。转子铁心上没有绕组,转子的齿数为 40,相邻两个齿之间夹角为  $9^\circ$ 。

当某一相绕组通电时,由于定转子上有齿和槽,所以当转子齿的相对位置不同时,在磁场的作用下,转子将转动一个角度,使转子和定子的齿相互对齐,这就是使步进电机旋转的原因。

步进电机运转是由脉冲信号控制。通过改变各相通电的次序可以调整步进电机的运转方向。

改变脉冲信号的周期就可以改变步进电机的运转速度。

本实验板选用的是四相步进电机,实验例程采用四相八拍的方式驱动步进电机。

正方向: A-AB-B-BC-C-CD-D-DA-A

反方向: A-DA-D-CD-C-BC-B-AB-A

AA,BB,CC,DD 是步进电机的 4 个线圈,D8,D9,D11,D12 为续流二极管,是为步进电机的 4 个线圈放电用的,当电机断电时,它的线圈要放电用二流管导流。

教学实验机的步进电机驱动信号是 STEPA~STEPD,本例程用 P5.0~P5.3 来控制步进电机的 4 个线圈处于通电或断电状态,所以在运行例程前,用跳线将 STEPA~STEPD 分别同 P5.0~P5.3 连接;同时也应将步进电机的引出线同 CN2 正确连接(VCC—黑,AA—红,BB—黄,CC—白,DD—蓝)。

## (4)、实验程序参考框图

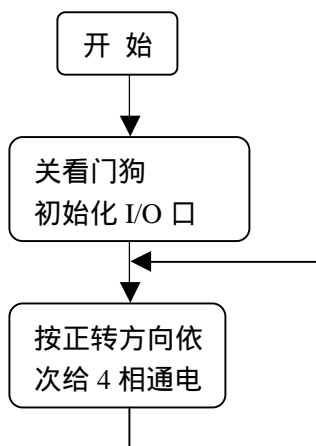


图 3.6.1 步进电机控制程序参考框图

## (5)、调试方法

下载此程序后并运行程序,步进电机是否会正方向旋转。

将旋转顺序改为反方向，步进电机是否会反方向旋转。  
改变每步的延时时间来控制步进电机的速度调节。

### 3.6.2 直流电机驱动实验

#### (1)、实验目的

利用 C8051F020 的 D/A 输出来控制直流电机的转动速度。

#### (2)、实验说明

实验前将 J9 短接以便 DAC0 的输出能控制直流电机的驱动，J15 的两端分别连直流电机的两引出现（P1 脚—红，P2 脚—黑）。

通过 DAC0 的输出来控制直流电机转动的快慢，当 D/A 输出较小时转速慢，D/A 输出较大时转速快。

#### (3)、实验程序参考框图

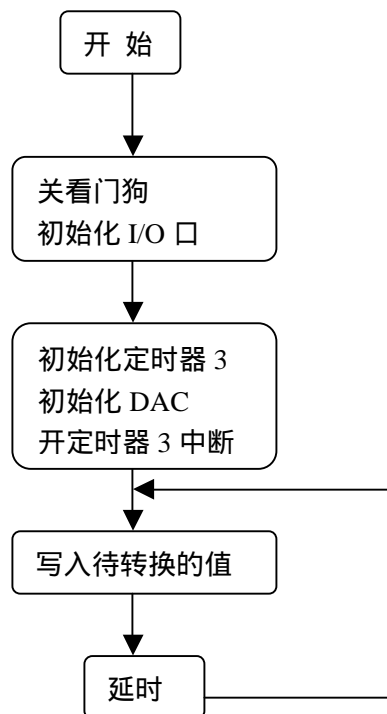


图 3.6.2 直流电机驱动流程图

#### (4)、调试方法

下载例程并运行，直流电机是否旋转。

改变例程待 D/A 的初值，从新编译下载后，直流电机的转速是否有所改变。

所赋的初值不能过小，否则 D/A 的输出不足以驱动开关管 Q1 的导通，致使直流电机不旋转。

### 3.7 可编程计数器阵列(PCA)实验

#### 3.7.1 PCA 频率输出功能实验

##### (1)、实验目的

熟悉 PCA 的频率输出功能。

##### (2)、实验说明

此程序利用捕捉/比较模块 0 实现 PCA 频率输出方式，将捕捉/比较模块 0 的 CEX0 配置在 P0.0 口，用示波器测量 P0.0 的波形，如果正确有不断变化频率的方波输出。

##### (3)、实验程序参考框图

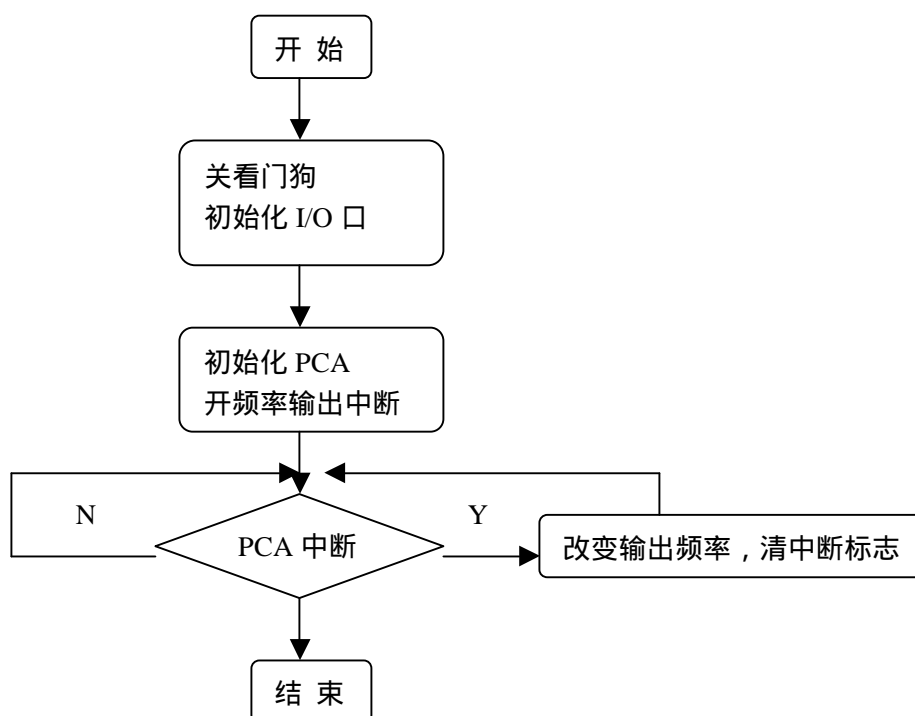


图 3.7.1 PCA 频率输出功能流程图

##### (4)、调试方法

下载例程并运行，用示波器量测 P0.0 的波形观察变化的方波。

在例程里改变 PCA 中断服务子程序( void PCA\_ISR(void) )中 PCA0CPH0 所加的值，看频率变化的幅度比较于上次的是否有所改变。

#### 3.7.2 PCA 捕捉功能实验

##### (1)、实验目的

熟悉 PCA 的捕捉功能。

##### (2)、实验说明

此程序利用捕捉/比较模块 0 实现 PCA 边沿触发的捕捉方式，将捕捉/比较模块 0 的

CEX0 配置在 P0.0 口，且配置成下降沿捕捉。

此例程从 p1.6 脚模拟下降延，实验前用跳线将 P1.6 与 P0.0 短接作为捕捉/比较模块 0 的外部触发沿。

从 p1.6 脚模拟出 60000 个下降延，每一个下降沿捕捉进入一次 PCA 捕捉中断，在中断服务子程序中用一个变量计数，程序运行结束变量值是否是 60000。

### (3)、实验程序参考框图

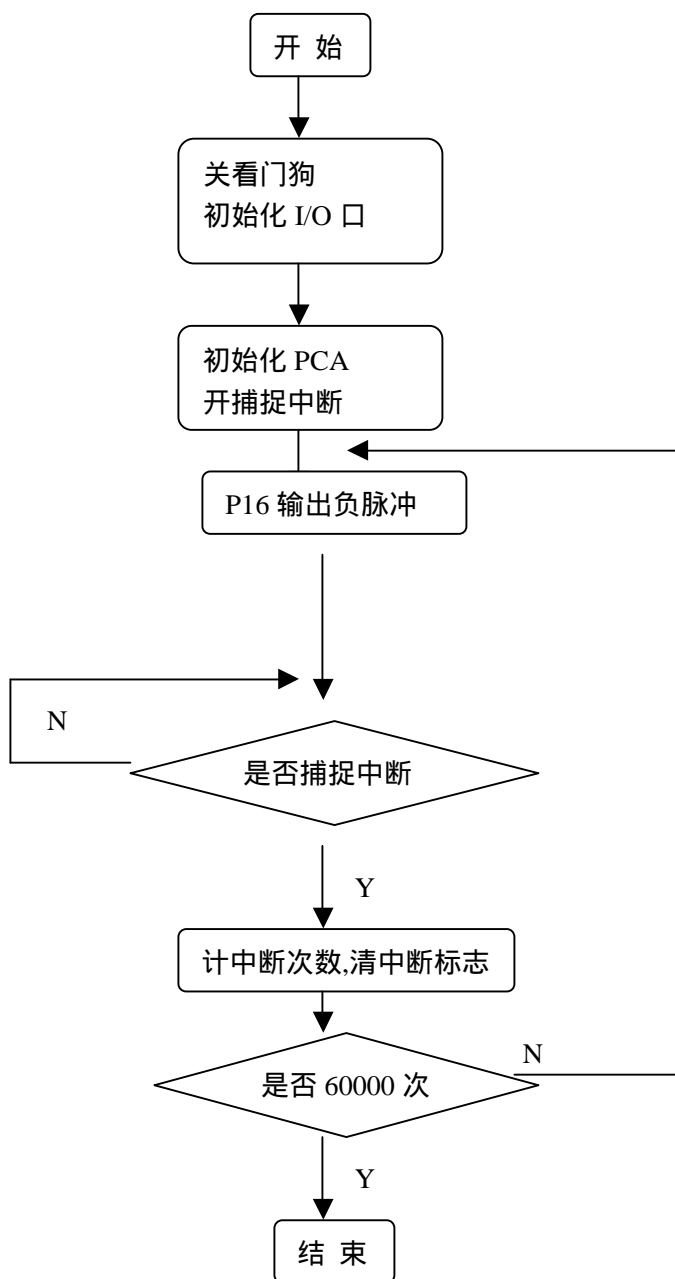




图 3.7.2 PCA 捕捉功能流程图

## (4)、调试方法

下载例程并运行，60000 个模拟下降沿发出后，计数变量 j 的值是否是 60000。  
例程适当修改，用外部的信号源触发捕捉功能，看程序是否正常运行。

## 3.7.3 PCA 输出 16 位 PWM 实验

## (1)、实验目的

熟悉 PCA 的 16 位脉宽调制器方式。

## (2)、实验说明

此程序利用捕捉/比较模块 0 实现 PCA 的 16 位脉宽调制器方式，将捕捉/比较模块 0 的 CEX0 配置在 P0.0 口，用示波器量测 P0.0 的波形会有 PWM 波形输出。

## (3)、实验程序参考框图

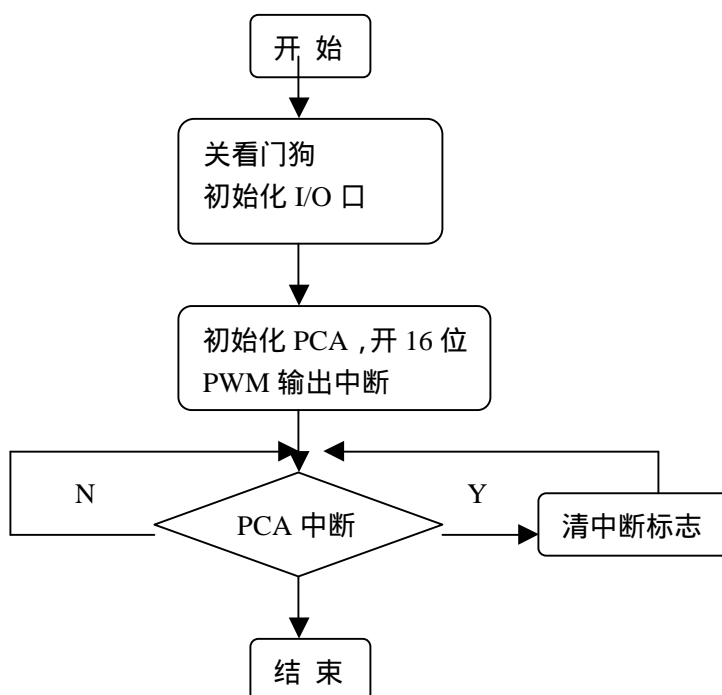


图 3.7.3 PCA 输出 16 位 PWM 流程图

## (4)、调试方法

下载例程并运行，用示波器量测 P0.0 的波形观察 PWM 的波形。

在例程里改变 PCA 初始化子程序 (void PCA\_Init (void)) 中捕捉模块 0 寄存器 PCA0CPH0 和 PCA0CPL0 的值，看脉宽比较于上次的是否有所改变。

### 3.8 A/D 转换实验

#### (1)、实验目的

熟悉 C8051F020 的 A/D 转换功能。

#### (2)、实验说明

此例程利用 ADC0 实现 A/D 转化的功能,使用内部电压基准作为 A/D 的参考电压,且教学实验机上 VREF 已经同 VREF0 短接。

例程的输入通道选择的是 AIN0.0,实验前将 S6 的拨码开关 1 设置到 ON,其余开关拨到 OFF。

待转化的输入信号是一个电压信号,此信号通过一个可调电阻将电源电压进行分压,实现变化的输入电压信号,A/D 转换后的数据不断送到液晶显示屏上显示,供实验时数据参考。

#### (3)、实验程序参考框图

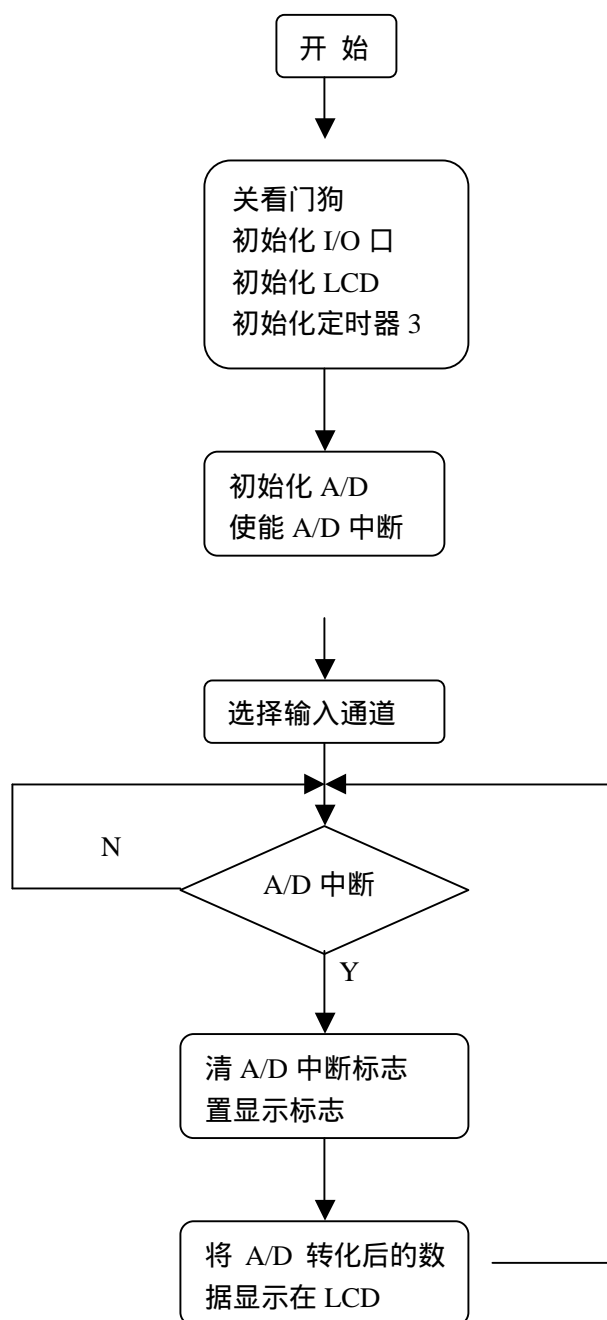


图 3.8 A/D 转换流程图

#### (4)、调试方法

下载例程并运行，如果模/数转换正确，在液晶显示屏会显示当前输入值转换后的数值大小，每成功转换一次则送到液晶屏显示一次，在输入信号保持不变的情况下，所显示

的数值基本不变，最多有几个字的上下波动。

旋转可调电位器 R47 改变输入的电压值，液晶屏上显示的数据会依据输入电压的变化相应的增大或缩小。

适当修改例程，选择不同的输入通道（拨码开关相应设置），重复上面的过程。条件允许可以输入外部待转化的信号，但必须保证不能超过极限值。

### 3.9 比较器功能实验

#### (1)、实验目的

掌握 C8051F020 比较器的使用。

#### (2)、实验说明

程序可完成 CP1 比较器的比较功能，使用内部电压基准，用基准电压作为比较信号电压，实验开始前将 CP1N 用跳线接至 VREF 脚。

#### (3)、实验程序参考框图

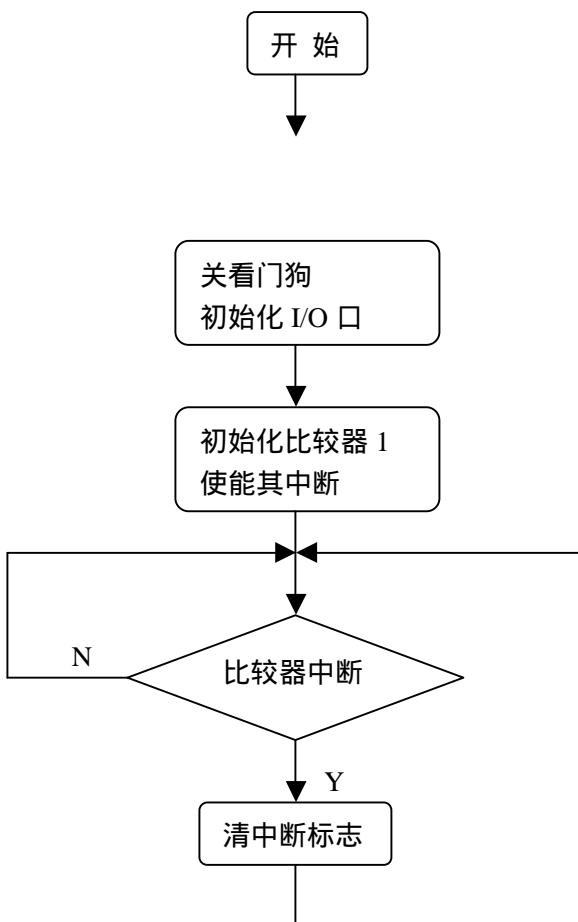


图 3.9 比较器功能流程图

#### (4)、调试方法

例程运行前由于 CP1N 接至 VREF(2.4V)，而 CP1P 悬空电压值是 0V 左右， $CP1N > CP1P$ ，比较器输出为负，当将 CP1P 同电源电压轻触仪下，在此瞬间 CP1P 接至 3V 左右，有  $CP1P > CP1N$ ，比较器输出位正，例程进入比较器中断，以此判定比较器工作正常，而如果轻触模拟地，由于还是  $CP1N > CP1P$ ，比较器不触发工作。

另外，CP1P 的高/低电平可以从 CN6/CN7 通过跳线取得，其中 CN6 通常被上拉到 3V，当 KEY5 被按下时，CN6 的电压经过两电阻的分压得到一低压，低压的大小可适当调节两电阻的阻值；而 CN7 通常被下拉到地，当 KEY6 被按下时，CN7 的电压经过两电阻的分压得到一高压，高压的大小可适当调节两电阻的阻值。

### 3.10 红外收发实验

#### (1)、实验目的

熟悉红外收/发器的工作原理，理解红外收/发器的电路设计，学会应用红外收/发器。

#### (2)、实验说明

HS0038A2 是集成红外接收头。因为它的输出信号与解码器的输入信号反向，故加三极管 Q2 作为反向器。当 MC145027 收到信号并解码成功后，其 VT 端出现高电平，同时输出解码后的 4 位数据 (HD0~HD3)。将 VT 信号反向，即可产生中断信号向 MCU 申请中断 (实验前将 J17 的 Pin1 和 Pin2 短接，用外部中断 6 作为红外接收数据的中断信号源)。

MC145026 与 MC145027 配对工作，各有 5 条地址线，4 条数据线。

红外接收电路在主板上，而红外发送电路是另外独立的模块，两者配合使用，工作前 J17 的 1,2 脚短接，用外中断 6 作为红外接收中断处理信号，同时拨码开关 S2 的拨码开关都拨到 ON，配套使用红外发送模块的拨码开关也都拨到 ON。

#### (3)、实验程序参考框图

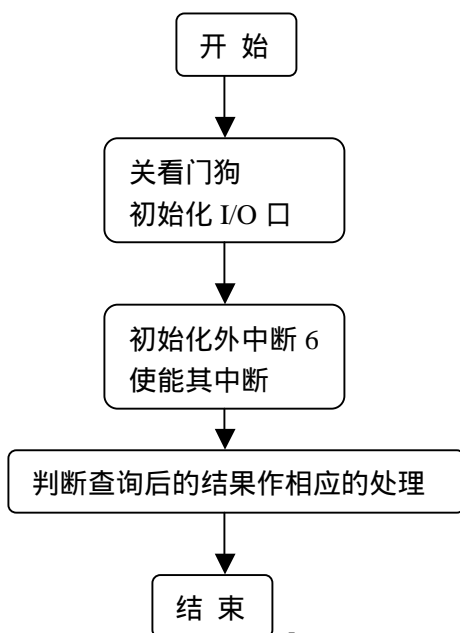


图 3.6.2 红外接收流程图

#### (4)、调试方法

将红外发射电路板上的拨码开关和红外接收电路的拨码开关都拨到 ON。

打开主板和发射板电源，将发射板与主板上的红外接收头正面相对，按动红外发射板上的 8 个按键，在主板上运行例程，观察主板上的接受的数据是否是红外发射电路板发送的信号，如果接收正确，蜂鸣器响一声。

### 3.11 电子音响实验

#### (1)、实验目的

熟悉用单片机的 I/O 口输出特定频率的某段音乐。

#### (2)、实验说明

编谱说明：低音(简谱中数字下面有一个点的)1234567 对应的为小写 cdefgab

中音(简谱中数字上下都没有点的)1234567 对应的也为 1234567

高音(简谱中数字上面有一个点的)1234567 对应的为大写 CDEFGAB

对于降音符 b 或声音符#一律用 # + 合适的音名例如#5

一个音符本身为一拍，加下划线后为半拍加等号为 1/4 拍 如:65\_ 4 = 则

音 6 为一拍，音 5 为半拍,音 4 为 1/4 拍

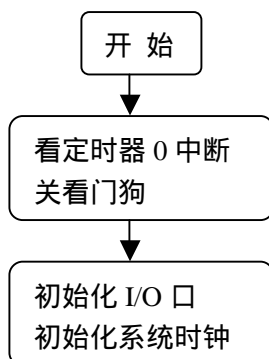
下划线或等号连续书写则音长连续变短

音符后加-或.表示延长。‘ - ’ 延长一拍 ‘ . ’ 延长半拍多加则延长连续增加

每个音符都对应着一个特定的频率，例程运行前用跳线将 P54 同 MUSIC 连接，即用 P54 脚输出特定频率的音符，经音频电路演奏出音乐。

实验前将 CN19 用跳线短接，给语音发声电路供电，实验结束将此跳线取消，以免喇叭产生电流的啸叫声。

#### (3)、实验程序参考框图



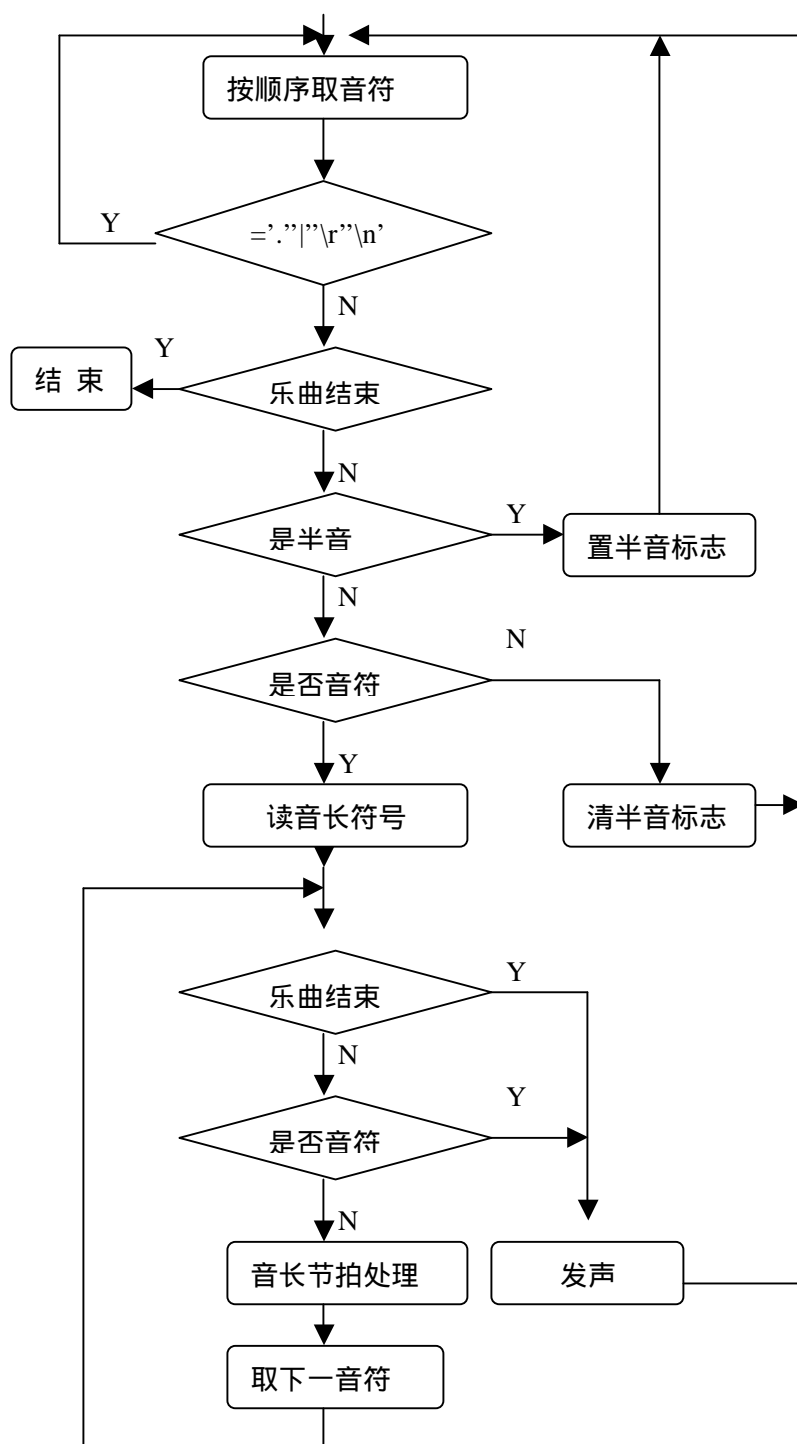


图 3.11 电子音响流程图

## (4)、调试方法

下载程序并运行，看是否会产生一段音乐。

### 3.12 频率计实验

#### (1)、实验目的

掌握频率计的工作原理，熟悉此频率计的电路原理。

#### (2)、实验说明

例程使用实验板上的时钟信号发生器作为它的的脉冲源，而不是从实验板外加入，所以 J6 的 2, 3 脚短接，时钟信号发生器电路拨码开关 S1 在实验时只能有一个拨码开关设置到 ON,其基频是晶体的震荡频率 2MHz,另三个频率输出是它的 2/4/8 分频。

例程只用到 T0 作为定时器 0 的外部计数源,没有用到定时器 1 的外部触发源 T1,所以 J7 不需跳线短接，但需要将 T0 与 P00 用跳线短接。

例程中用到 P31 来控制外部脉冲信号是否被送到定时器 0 的外部输入脚，所以拨码开关 S3 的 2, 4 开关拨到 ON,1, 3 拨到 OFF。

例程是定时器 0 作为外部计数器，定时器 1 作为定时器，在定时 1ms 范围内察看计数器 0 所计的数，以此计算出外部脉冲的频率。

#### (3)、实验程序参考框图



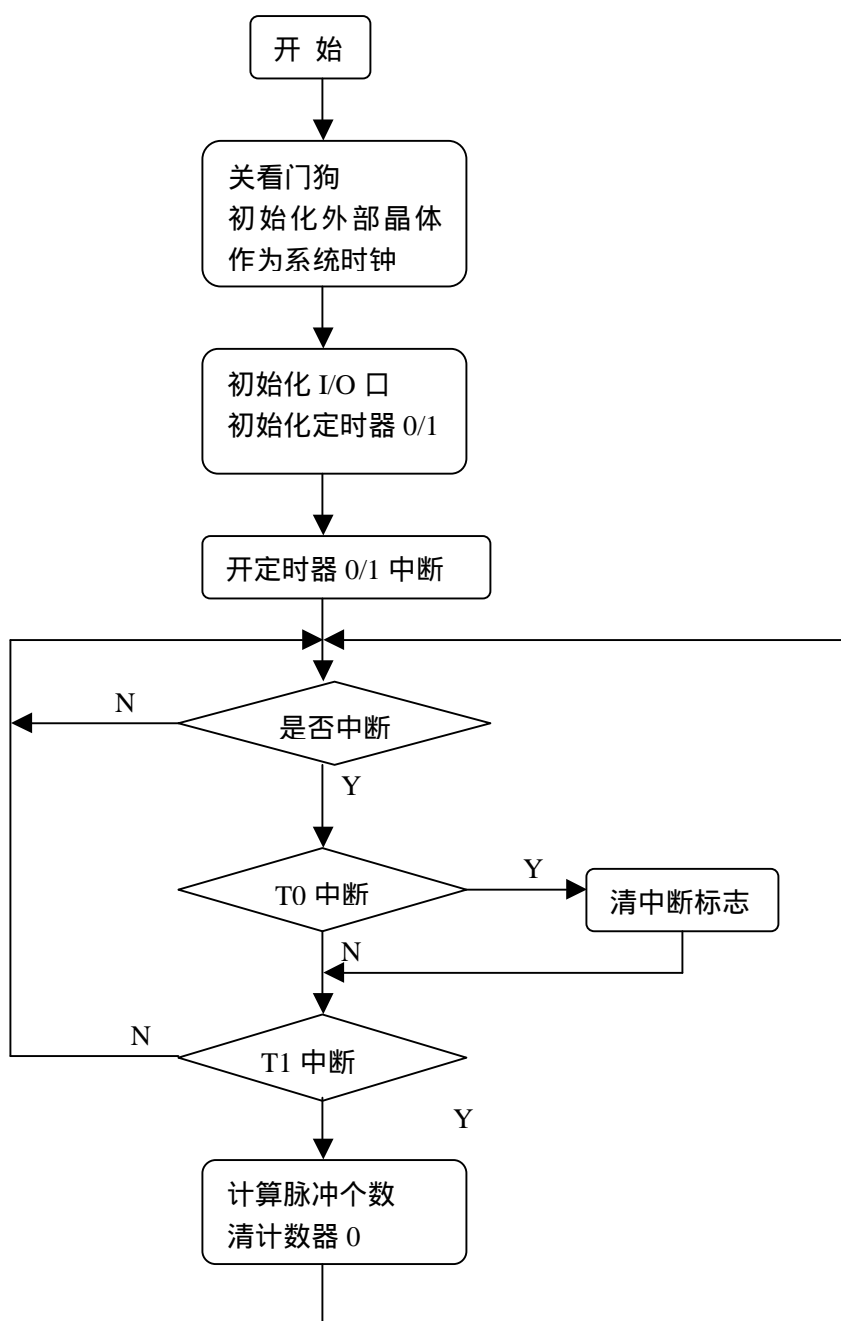


图 3.12 频率计流程图

#### (4)、调试方法

将断点设在定时器 1 中断服务程序里,运行例程直到断点,察看计数变量 Frequency 的值,其所花费得时间为 1ms,据此算出外部脉冲的频率值。

用不同频率的外部脉冲信号(通过设置拨码开关 S1),重复上面过程判定例程是否

工作正常。

将例程稍作修改，用定时器 2 计外部脉冲个数，程序是否也能正常计数。

### 3.13 外部中断实验

#### (1)、实验目的

熟悉 C8051F020 外部中断 6/7 的使用。

#### (2)、实验说明

此程序测试 C8051F020 的外部中断 6、7，可在相应的两个中断服务子程序中设断点观察，平常 INT6/INT7 被上拉到高电平状态，当 KEY3 或 KEY4 有键按下时接地，触发一个下降沿使 INT6 或 INT7 进入外部中断服务子程序。

#### (3)、实验程序参考框图

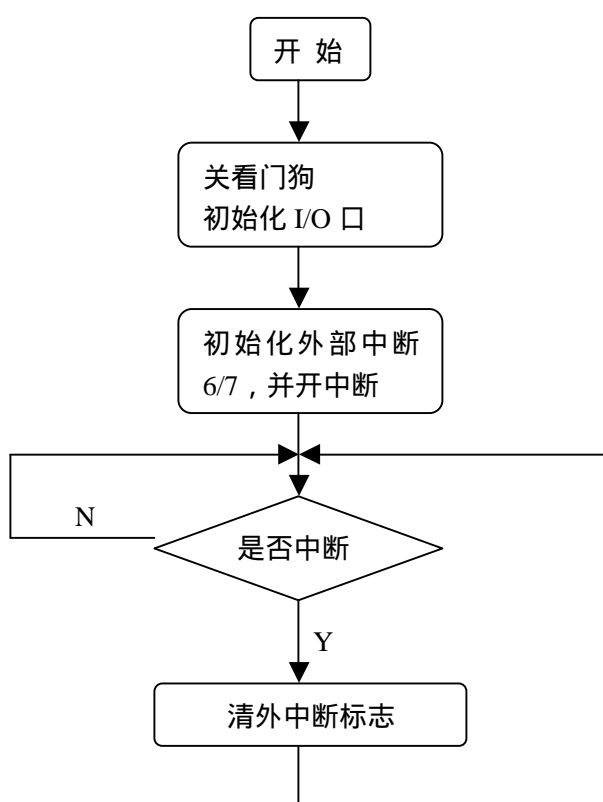


图 3.13 外部中断流程图

#### (4)、调试方法

分别按下 KEY3 或 KEY4，外部中断 6/7 是否会响应中断进入中断服务子程序。

附录 1 C8051F 系列单片机开发工具 JTAG 口线定义

附录 2 系统实验板原理图

附录 3 系统实验板元件排列图

## 附录 1 C8051F 系列单片机开发工具 JTAG 口线定义

引脚	说明
1	3.0 to 3.6VDC 输入
2 , 3 , 9	接地
4	TCK
5	TMS
6	TDO
7	TDI
8 , 10	没连接

