

第4章 C51语言及通用I/O口应用

C51的程序结构

C51语言概述

C51语言是标准C语言用于51单片机的子集。除了少数扩展功能外，大多数C语言的功能，在C51中都能运行。

C51与标准C的差别：
①数据类型方面——51单片机中的特殊功能寄存器SFR、可位寻址存储空间、工作寄存器Rn等，都是51单片机的特有资源，C51具有对其进行访问和操作的特殊规则；
②存储类型方面——51单片机采用哈佛结构存储空间，程序和数分别存储在不同的存储空间中，因而C51有表征不同存储空间变量类型的能力；
③函数属性方面——由于51系统是8位机资源有限，不允许太多的复杂运算，因而标准C的库函数中只有少量可用于C51，且C51的中断函数也是有其特殊规则的；
④输入输出方面——C51的输入输出是通过访问机器端口的映射地址实现的，而标准C语言则是通过如printf和scanf等输入输出库函数实现的，两者做法相差较大。

C51程序的基本结构为：
预处理命令
函数类型声明
全局变量定义
main() {
 局部变量定义
 <程序体>
}

func1() {
 局部变量定义
 <程序体>
}

C51程序开发方法

C51仿真开发方法

- #### 完成C51程序准备
1. 新建项目——选中“创建项目”单选项，以及8051、80C51、Keil for 8051三个选项；
 2. 建立*.c文件——将程序添加到Source Codes编辑区，保存为*.c文件；
 3. 编译程序——单击菜单【构建】—【构建工程】进行程序编译；
 4. 打开调试界面——源代码调试窗口和变量调试窗口；
- #### C51仿真调试方法
- ①开始仿真，快捷键Ctrl+F12，进入代码调试界面等待运行，等同于STEP按钮；
 - ②暂停仿真，无快捷键，暂停仿真运行，等同于PAUSE按钮；
 - ③停止仿真，无快捷键，终止仿真运行，等同于STOP按钮；
 - ④运行仿真，快捷键F12，全速运行程序，遇到断点会停止运行，等同于PLAY按钮；
 - ⑤不加断点仿真，快捷键Alt+F12，全速运行程序，遇到断点也不停止运行；
 - ⑥运行仿真（时间断点），无快捷键，全速运行程序，直到预设的时间才停止运行；
 - ⑦单步执行，快捷键F10，一行程序执行完后就停止运行。如遇到被调用的函数时，会将函数看作是一条语句整体执行；
 - ⑧跳进函数，快捷键F11，单步运行到被调用的函数时，将进入函数中继续单步执行；
 - ⑨跳出函数，快捷键Ctrl+F11，从被调用的函数中跳出，运行到函数后的下一个程序行；
 - ⑩跳到光标处，快捷键Ctrl+F10，从当前程序行开始高速运行，到达光标所在行时停止；
 - ⑪连续单步，快捷键Alt+F10，一行程序执行完后不停下来慢慢执行下一行程序。

实例4.1 循环流水灯

为了使用C51系统的内部库函数，需要在源程序开头添加一条预处理命令“#include <intrins.h>”

I/O口方式的简单应用

基本输入/输出电路

- #### 发光二极管
- 最基本的输出设备，具有电路简单、功耗低、寿命长、响应速度快等特点；
 - 具有低电平驱动和高电平驱动两种驱动方式。
- #### 按键或开关
- CPU通过读I/O口电平即可识别是哪个按键或开关闭合。
- P0口可以检测到1和0两种状态（有上拉电阻时）；
 - P0口只能检测到是否为0（无上拉电阻时）；
 - P1-P3口可以检测到1和0两种状态（无论有无上拉电阻时）。
- #### 实例4.2 独立按键
- 编程要点：
• key = P0 & 0x0f，使key值高4位始终为0；
• if (key! = 0x0f) P2=key，保持先前的亮灯状态直到有新的键值刷新。
- #### 实例4.3 键控流水灯
- 编程要点：
• 设立两个可根据键值修改的标志变量，根据标志变量的组合关系控制流水灯的流向与启停；
• 将预存在显示数组里的编码数据利用数组下标法依次输出到P2口；
• 流水灯切换速度可通过调用延时函数进行控制。

数码管原理与静态显示

- #### 共阳极和共阴极数码管
- 七段LED数码管内部由7个条形发光二极管和一个小圆点发光二极管组成；
 - 共阳极数码管的所有LED阳极连接在一起，为公共端引脚，所有LED阴极单独接出作为段位引脚；
 - 共阴极数码管的所有LED阴极连接在一起，为公共端引脚，所有LED阳极单独接出作为段位引脚；
 - 改变笔顺a~g的组合电平就能形成不同的字形，这种组合电平称为显示字模。
- #### 实例4.5 LED数码管显示
- 编程要点：
• 将显示字模按显示字代表的数值大小顺序存入一字符串数组中；
• 使用时，只需用待显示值作为下标变量调用该数组，即可取得相应的字模。
- #### 实例4.6 计数显示器
- 编程要点：
• 软件消抖法。即当检测到有键按下时，先用软件延时10ms，然后检测按键的状态；
• 计数值先分再显示：将count用取模运算（count%10）折出个位值，用整除运算（count/10）折出十位值。

I/O口方式的进阶应用

数码管动态显示原理与应用

- #### 静态显示接口
- 一个并行口接一个数码管。优点是送入并行口后就不再需要CPU干预，因而显示亮度稳定，但该方法占用资源较多。
- #### 动态显示接口
- 将多个数码管相同段码引脚串联起来接在一个8位并行口上，而每个数码管的位码引脚（公共端）分别由一根I/O口线控制；
 - 采用循环导通或循环截止各个数码管的做法，即快速（如10ms）切换段码值和位码值，使每一时刻只有一个数码管被驱动；
 - 动态显示接口的突出特点是占用资源较少，但由于显示值需要CPU随时刷新，故其占用时较多。
- #### 实例4.7 数码管动态显示
- 编程要点：
• 将位码0x02和0x01先后送入P3口，然后依次使能左、右两个数码管；
• 利用数组下标值计算LED位码，即led_point=1-led_point；

行列式键盘原理与应用

- #### 独立式键盘
- 每个键分别接在一根I/O口线上。其特点是电路简单，易于编程，但占用的I/O口线较多。
- #### 行列式键盘
- 接线——将I/O口线分为行线和列线，按键设置在跨接行线和列线的文点上，列线通过上拉电阻接正电源；
特点——占用I/O口线较少，但软件部分较为复杂。
- #### 行列式键盘的检测
- 软件扫描查询法——根据按键按下前后，所在行线的I/O口电平是否出现翻转，判断有无按键闭合动作。具体步骤：
1. 键盘扫描——使键盘的4个行线电平全为1，列线轮流流有一列0，其余为1；
2. 按键判断——利用条件表达式（P2&0x0f）判断有无按键按下；
3. 键值计算——通过逐一对比P2口读入值与键模数组，可求得闭合按键的键值
- #### 实例4.8 行列式键盘
- 硬件连接：4×4行列式键盘和1位共阴极数码管
编程要点：根据上述行列式键盘检测流程
- #### 实例4.9 1位密码锁
- 编程要点：
• 按闭合检测可以采用实例4.8的getKey0函数；
• 根据按闭合值比对开锁密码，然后控制LED显示。

C51的数据结构

C51变量

- #### 存储种类
- C51变量的一般格式如下：
[存储种类] 数据类型 [存储类型] 变量名；
- 自动型 (auto) ——作用域是定义它的函数体或语句组内，当函数调用结束或语句组执行完毕时，自动型变量所占用的存储单元就被释放；
 - 静态型 (static) ——作用域是定义它的函数体、程序文件或语句组内，静态型变量具有变量的隐蔽性、存储持久性和默认0初值3个特点；
 - 外部型 (extern) ——如果变量的定义与使用不在同一个作用域内，则用extern声明后就能将原作用域扩展到声明所在的位置；
 - 寄存器型 (register) ——将变量的存储单元指定为寄存器变量，便于在使用中频繁地与内存进行数据交换。
- #### 数据类型
- ##### 标准C语言传承的
- 基本数据类型
unsigned char表示无符号字符型变量，[signed] char表示有符号字符型变量；
unsigned int表示无符号整型变量，[signed] int表示有符号整型变量；
unsigned long类型说明符表示无符号长整型变量，[signed] long表示有符号长整型变量；
float表示浮点型变量；double表示双精度型变量
- ##### C51特有的
- bit型变量
类型说明符变量名 (=0或1)；
类型说明符为bit，可省略的变量初值为0或1
 - sfr型变量
类型说明符变量名=8位地址常量；
用于8位SFR变量定义的是sfr
用于16位SFR变量定义的是sfr16
不可省略的8位地址常量是指有意义的SFR字节地址
 - sbit型变量
第1种：sbit 位变量名 = 位地址；
第2种：sbit 位变量名 = 可位寻址的SFR字节地址^相对位置
第3种：sbit 位变量名 = 可位寻址变量^相对位置
- ① bit和sbit区别：bit型变量的位地址是由编译器为其在片内RAM的可位寻址区（bdata区）随机分配的，而sbit型变量的位地址则是由用户在可位寻址的SFR单元内指定的。
② sfr型变量和sbit型变量都必须定义为全局变量，即必须在所有C51函数之前进行定义，否则会编译出错。
- “reg51.h”头文件中已对sfr和sbit进行过变量定义，添加预处理命令“#include <reg51.h>”，就无需再对这些变量定义了。

C51指针

- 用来存放变量地址的变量，称作“指针变量”，存放的地址称为“指针”。
- #### 标准C语言指针变量定义
- 数据类型 * 指针变量名 [= &被指向变量名]；
- #### C51指针变量定义
- 数据类型 [存储类型1] * [存储类型2] 指针变量名 [= &被指向变量名]；
- “数据类型”是被指向变量的数据类型
 - “存储类型1”是指被指向变量所在的存储类型
 - “存储类型2”是指指针变量所在的存储类型
 - 指针变量名可按C51变量名的规则选取

C51与汇编的混合编程

- #### 在C51中调用汇编程序
- 混合编程技术可以把C51语言和汇编语言的优点结合起来，即程序的框架或主体部分用C语言编写，对那些使用频率高、要求执行效率高、延时精确的部分用汇编语言编写，这样既保证了整个程序的可读性，又保证了单片机应用系统的性能。
1. 程序的寻址是通过在汇编文件中定义同名的模块标号实现的，例如与max(char a, char b)函数对应的汇编程序名为_MAX；
 2. 要用工作寄存器传递传入参数，例如，传入max(char a, char b)函数的参数char a和char b要用到工作寄存器R7和R5；
 3. 汇编程序的返回值要通过工作寄存器实现，例如与c=max(char a, char b)对应的char c变量要用工作寄存器R7返回。
- #### 在C51中嵌入汇编代码
1. 在C51中嵌入的汇编代码必须放在两条预处理命令#pragma asm和#pragma endasm之间；
 2. 预处理命令必须用小写字母，汇编代码则大小写字母不限；
 3. 需要在Keil编译器中进行SRC设置；
 4. 需要将库文件添加到当前项目文件中；
 5. Proteus v8目前尚不支持SRC操作。