Keil C51与Proteus软件使用简介

Keil C51 和 Proteus 是使用较多的两种单片机系统开发工具,它们功能强大、简单易用,特别适合初学者。Keil C51 和 Proteus 诞生至今,均经历了多个版本,本章分别以 Keil µ Vision 5 版和 Proteus ISIS 7.8 Professional 版为例,介绍 Keil C51 和 Proteus 软件的工作环境和一些基本操作。

5.1 Keil C51 软件使用

Keil C51 是美国 Keil Software 公司出品的 51 系列兼容单片机 C 语言软件开发系统, 可以在 Windows 98、Windows NT、Windows 2000、Windows XP 等操作系统上运行。Keil 提供了包括编译器、宏汇编、连接器、库管理和一个功能强大的仿真调试器等在内的完整开 发方案,通过一个集成开发环境(µVision)将这些部分组合在一起,支持汇编、C、PL/M 语 言。系统提供丰富的库函数和功能强大的集成开发调制工具,界面友好,易学易用。

μVision 中的文件采用项目方式管理,各种 C51 源程序、汇编源程序、头文件等都放在 项目文件里统一管理。一般操作步骤如下:

- (1) 启动软件;
- (2) 建立项目文件;
- (3) 给项目添加程序文件;
- (4) 编译、连接项目,形成目标文件;
- (5) 仿真运行、调试、观察结果。

接下来以一个简单的实例来完整地介绍 Keil C51 的处理过程和基本操作。

P1 口连接 8 个发光二极管指示灯,编程实现流水灯的控制,从低位到高位轮流点亮指示灯,一直重复。

5.1.1 启动 Keil C51

双击桌面上的µVision 5图标,界面如图 5-1所示,接着出现主界面,如图 5-2所示。

μVision 5 的主界面窗口标题栏下紧接着是菜单栏,菜单栏下面是工具栏,工具栏下面的左 边是项目管理窗口,右边是编辑窗口,可以通过视图菜单(View)下面的命令打开或关闭。 文件操作、编辑操作、项目维护、开发工具选项设置、调试程序、窗口选择和处理在线帮助等 各种操作菜单都由菜单条提供,而由用户自行设置的键盘快捷键由工具条提供。



图 5-1 启动 µVision 5 时显示的界面



图 5-2 µVision 5 的主界面

μVision 5 有编辑模式和调试模式两种操作模式,通过用 Debug 菜单下的 Star/Stop Debugging(开始/停止调试模式)命令切换。在编辑模式中可以建立项目、文件,编译项目、 文件产生可执行程序;调试模式则提供一个非常强劲的用来调试项目的调试器。

5.1.2 新建项目文件

μVision 5 采用项目方式管理,新建项目时可以首先建一个用于存放项目的文件夹,然 后启动μVision 5,建立项目文件,具体过程如下: (1) 在编辑模式下,选择 Project 菜单下的 New μVision Project 命令,弹出如图 5-3 所示的 Create New Project 对话框。

W.	Creat	e New Project			>
⊕ ⊙ - ↑ 📭	这台电脑 🕨 文档 🕨	~ C	搜索"文档"		P
组织 ▼ 新建文件 4	E.			- 12	0
☆ 收職夾	A JDMark 11	Dolphin Emulator	Downloads	Fetion	
■ 桌面 ▲ OneDrive ▲ 家庭組 及 yulan han ↓ .android	FetionBox	KingsoftData	Listings	MATLAB	
文件名(N): ex	ample				
保存类型(T): Pr	oject Files (*.uvproj; *.uv	projx)	保存(S)	取消	

图 5-3 Create New Project 对话框

(2) 在 Create New Project 对话框中选择新建项目文件的位置(为项目建立的文件 夹),输入新建项目文件的名称,项目文件类型固定为 uvproj,例如,项目文件名为 example, 单击"保存"按钮,将弹出如图 5-4 所示的 Select Device for Target 'Target 1'…对话框,用户

Select	Device for Target 'Target 1'
Device	
	¥.
Vendor: Microchip	
Device: AT89C51	
Toolset: C51	
Search: AT	Use Extended Linker (LX51) instead of BL51
	Use Extended Assembler (AX51) instead of A51 Description:
	Description.
AT87F51RC	8051-based Fully Static 24MHz CMOS controller with 32 I/O Lines,
- C AT87F52	Three-Level Program Memory Lock, 4K Bytes Flash Memory,
AT87F55WD	128 Bytes On-chip RAM
AT89C1051	1
AT89C1051U	
AT89C2051	
AT89C4051	
AT89C51	
AT89C5115	
AT89C5130	c >
AT89C5130	0K Cancel Hel;

图 5-4 Select Device for Target 'Target 1'…对话框

可以根据使用情况选择单片机型号,如选择 AT89C51。μVision 5 几乎支持所有的 51 核心 的单片机,并以列表的形式给出。选中芯片后,在右边的描述框中将同时显示选中芯片的相 关信息以供用户参考。

(3)选择好单片机芯片后,单击"确定"按钮,此时弹出 Copy Standard 8051 Startup Code to Project Folder and Add File to Project 确认对话框,如图 5-5 所示。若用 C51 语言 编写程序,则选择"是";若用汇编语言编写程序,则选择"否"。本节实例用 C51 语言编写, 所以选择"是"。此时,项目文件就创建完成。项目文件创建后,界面如图 5-6 所示,从左边 的项目管理器窗口中可以看到新建的项目"Project: example",此时的项目只是一个框架, 随后需要向项目文件中添加相应的程序文件内容。



图 5-5 确认对话框

File Edit View Project Plash Debug Periphenals Tools SVCS Window Help Image:	C:\Users\yulan\Documents\example.uvproj -	µVision – 🗆 🗙
回答 通答 1	File Edit View Project Flash Debug Peripherals Tools SVCS Window Help	
Image: Section of the section of t	□22日間 3 日間 9 0 4 → 2 2 2 2 3 2 1 1 1 1 1 2 2 3 3 4 4 4 4 5 2 2 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	ا ا ا ا ا ا ا ا ا ا ا ا ا ا ا ا ا ا ا
Project a □ ¹ S Project ccample □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	🐵 🏻 🕮 😻 - 殿 🛱 Target 1 💿 🔊 🛔 틈 🗇 🐡 🎰	
Image: Second State Image: Second State Imag	Project 🛛 📮 🖸	
Build Output		
 Simulation 	Build Output	۵ 🖬
Simulation	<	~
	Si	mulation

图 5-6 新建项目文件后的工作界面

5.1.3 添加程序文件

项目文件建立好后,开始编写程序,给项目文件加入程序文件。µVision 5 支持 C 语言

程序,也支持汇编语言程序。若程序文件已经建立好了,可直接添加;若还没有程序文件, 须先建立程序文件再添加,过程如下:

(1) 如果没有程序文件,则应先用 File 菜单下的 New 命令建立程序文件,输入文件内容,并通过 File 菜单下的 Save as 命令输入文件名,保存文件。

需要注意的是,输入文件名时,必须输入正确的扩展名。若用 C 语言编写程序,则扩展 名为. C; 若用汇编语言编写程序,则扩展名为. ASM。

这里新建一个 C 语言程序,用于实现 P1 口连接的 8 个发光二极管指示灯,从低位到高 位轮流点亮,一直重复,存盘为 Text1.C 文件,具体内容如下:

```
# include < reg51. h>
# include < intrins. h>
# define uchar unsigned char
# define uint unsigned int
void delay(uint);
void main(void)
{
uint i;
uchar temp;
while(1)
{
temp = 0xFE;
for(i = 0; i < 8; i++)</pre>
{
P1 = temp;
delay(500);
temp = _crol_(temp,1);
} } }
void delay(uint t)
{
register uint bt;
for(;t;t-- )
for(bt = 0; bt < 255; bt++);</pre>
}
```

(2)程序文件建立好后,在项目管理器窗口中,展开 Target 1项,可以看到 Source Group 1 子项。

(3) 右击 SourceGroup 1,在菜单中选择 Add New Item to Group 'Source Group 1'命 令,如图 5-7 所示。

(4) 弹出 Add Files to Group 'Source Group 1'对话框,如图 5-8 所示。在该对话框中选择需要添加的程序文件,单击 Add 按钮便可以把所选文件添加到项目中。

注意,添加文件时,文件类型默认为*.c,当添加汇编程序时,需在文件类型选择框中选择*.a。另外,在项目管理器的 Source Group 1 下面选中对应的文件,在右键菜单中可以执行 Remove File(移除)等命令。

W.		C:\Users\yulan\Documents\example.uvproj - µVision	×
File Edit Vie	w	Project Flash Debug Peripherals Tools SVCS Window Help	
0000	3	3 四路 ウ ♡ ← → 門 忠 忠 忠 宇 定 止 版 図	2
0 I II I	. ھ	- 副 第 Target 1 🔍 系 晶 音 参 今 働	
Project		a 🖬 📄 Text1.c	▼ ×
□ 🕆 Project:	exam et 1	mple 1	
	Å.	Options for Group 'Source Group 1' Alt+F7	
		Add New Item to Group 'Source Group 1'	- 1
		Add Existing Files to Group 'Source Group 1' Remove Group 'Source Group 1' and its Files	
	(44)	Rebuild all target files	
		Build Target F7	- 1
	æ	Manage Project Items	
	1	Show Include File Dependencies	- 1
• . • P	} F	• [0, 7] <	>
Build Output			9 🖬
			< >
<			>
Add a new Item I	o Gr	iroup Simulation	L:1 _:



	C:\Users\yulan\Docu	ments\example.uvproj - µVisio	n	- 🗆 ×
File Edit View Project Flas	h Debug Peripherals Tools SVCS Wir	ndow Help		
	の で ← → 伊 県 内 段 津	谭 //= //=	Q .	000
◎ 田田 ※ • 殿 異	Target 1 🔍 🔊 🛔 着 🚸	令曲		
Project 🛛 🖬	Text1.c			▼ ×
Project: example Definition Target 1	1			
🖻 🗁 Source Group	Add Files to	Group 'Source Group 1'	×	
Text1.c	查找范围(I): ① 文档	· ← C ↔	•	
	名称	修改日期	美型 ^	
	 Visual Studio 2013 WeChat Files 手机模拟大师 我的形状 	2020/12/27 9:47 2021/2/1 21:54 2020/12/22 19:31 2014/10/17 23:07	文件 文件 文件 文件	
	副 日定义 Office 機板 Pl Text1.c	2015/1/21 20:37	C Sc	
	<		>	
	文件名(N): Text1.c		Add	
	文件美型(T): C Source file (*.c)	• .	Close	
• • • • • • • • • • • • • • • • • • •			li.	>
Build Output				a 🖸
				^
<				>
		Simulation	C	L1

图 5-8 Add Files to Group 'Source Group 1'对话框

5.1.4 编译连接成目标文件

程序文件添加到项目文件后,用 Project 菜单下的 Build Target 命令(或快捷键 F7),进行编译、连接,以形成目标文件。编译、连接后的界面显示情况如图 5-9 所示。



图 5-9 编辑、连接后的界面

编译成功后,在界面底部的信息窗口会给出提示信息。如果编译失败,在信息窗口中会 给出错误提示报告。双击错误报告某一行,窗口会自动跳到程序出错处,便于修改。修改后 可以再次进行编译、连接。

5.1.5 运行调试、观察结果

项目编译、连接成功后,进入调试模式,通过仿真运行观察结果。

首先执行 Debug 菜单下的 Start/Stop Debugging 命令(快捷键 Ctrl+F5)进入调试模式,如图 5-10 所示; 然后执行 Debug 菜单下的 Run 命令(快捷键 F5)连续运行; 接着执行 Debug 菜单下的 Stop 命令(快捷键 Esc)停止运行; 接着就可以选择 View 菜单调出各种输 出窗口观察结果,选择 Peripherals 菜单观察 51 单片机内部资源,图 5-11 为调出 Peripherals 菜单下的 P1 口观察的结果; 最后,运行调试完毕,先执行 Stop 命令停止运行,再执行 Debug 菜单下的 Star/Stop Debug Session 命令退出调试模式,结束仿真运行过程,回到编 辑模式。

W.		C	:\Users\yu	lan\Do	cuments	liush	uideng\example.uv	/proj - μVi	sion		-	□ ×
File Edit View	Project Fla	sh Debug	Peripherals	Tools	SVCS Win	dow	Help					
	X 12 18	00	+	1 12 12	1 後 律	津 //	/iz 🗃	5			000	
25 Q Q 7	9 0º 0º 1		a a i	8 4		- 12	· · · · · · · · · · · · · · · · · · ·	•				in the second se
Registers	4 🖬 Di	assembly		1.0.0						-		¢
Register	Value	6:	void main	(void)								
E-Regs		7:	{									
r0	0x00	9:	uchar tem	D:								
-r2	0x00	10:	while(1)									
-r3	0x00	11:	(
-r5	0x00											,
- r6	0x00	Texti.	C STAF	TUP.A51								•
E-Svs	0x00	8 u.	nt i;									
	0x00	9 uc	har temp;									
b	00x00	10 wh	ile(1)									
SD max	0x0a N	11 (man Ov FF .									
dptr	0x000	13 fc	r(i=0:i<8	:1++)								
PC \$	C:0x0 389	14 (
- sec	0.000	15 P1	=temp;									
🖻 – pzw	0x00	16 de	lay(500);		1.11							
		17 te	mp = _cro	1_(temp	p,1);							
		19 } }										
		20 vc	id delay	uint t)							
		21 (
Project ERegi	sters <	22.00	Mistor 111	nt ht.								>
Command						4 🖬	Call Stack + Locals					¢
Running with C	Code Size	Limit:	2K		13	0	Name		Location/Value		Туре	
Load "C:\\User	rs/\yulan	(\Docume	nts//lius	nuideng	\\Object	3/	B- MAIN		C:0x0800			1
						1007	- 9 i		0x0000		uint	_
<						>	@ temp		0x00		uchar	_
>												1)
ASM ASSIGN Bre	akDisabl	e BreakF	able Bre	kKill	BreakLis	E	Call Stack + Locals	Memory 1			_	
int notion bit		- secondo	Dec Dec		NE COREEO	-	and the state + totals	mulation.		44.4	00010450	
							SI	mulation		tie (100019430 560	C 121

图 5-10 启动调试过程



图 5-11 P1 口仿真窗口

当 Keil µVision 5 用于软件和硬件仿真时,如果不是工作在默认情况下,就需要在编译、连接之前对它进行设置,设置须在编译模式下,右击项目窗口中当前的 Target 1,从弹出的快捷菜单中选择 Options for Target 'Target 1'…命令,对如图 5-12 所示的对话框中的 11 个选项卡内容进行设置。

	Opt	ions for 1	Farget 'Target 1'		
Device Target	Output Listing User	C51	A51 BL51 Locate	BL51 Misc Debug	Utilities
Microchip AT89C51					
	Xtal (MHz):	24.0	Use On-chip ROM	(0x0-0xFFF)	
Memory Model:	Small: variables in DATA		•		
Code Rom Size:	Large: 64K program		•		
Operating system:	None		•		
Off-chip Code me	Eprom Start: Eprom Eprom Eprom	Size:	Off-chip Xdata memory	Start: Ram Ram Ram	Size:
	Qued.	End	'far' memory type su	sport	
Code Banking Banks: 2 💌	Bank Area: 0x0000	OxFFFF	Save address exten	sion SFR in interrupt	5

图 5-12 Options for Target 'Target 1'…对话框

值得说明的是,如图 5-13 所示,Output 选项卡中,勾选"Create HEX File"复选框,使程 序编译后产生 HEX 代码,供下载器软件使用,把程序下载到 AT89C51 单片机中。

8			Optio	ons fo	r Targe	t 'Tai	rget 1'				
Device Targ	et Output	Listing	User	C51	A51	BLS	51 Locate	BL51 M	liso Deb	oug	Utilities
Select	Folder for Ob	jects		Nar	me of Exec	utable:	example				
© Create	Executable:	.\Objects\e	cample								
I∕ Del	bug Informatio	on 		frowse in	normation						
I✓ Cre	ate HEX File	HEX Forma	It: THE	K-80		-					
C .\Obje	cts\example.	LIB							Creat	te Bat	ch File

图 5-13 Output 选项卡

另外,调试模式下,提供了多种调试方法对程序进行调试,主要方法如下:

(1) 设置和删除断点:设置/删除断点最简单的方法是双击待设置断点的源程序行或 反汇编程序行,或用断点设置命令 Insert/Remove Breakpoint。

(2) 查看和修改寄存器的内容: 仿真式寄存器的内容显示在寄存器窗口,用户除了可 以观察外,还可以自行修改,单击选中一个单元,输入相应的数值,按 Enter 键即可;另外, 可使用底部的命令行窗口进行修改。

(3)观察和修改变量:具体过程为,选择 View→Watch & Call stack Window 菜单命 令,出现相应的窗口,选择 Watch 1~3 中的任意窗口,按下 F2 键,在 Name 栏中填入用户变 量名,但必须是存在的变量。如果想修改数值,可单击 Value 栏,出现文本框后,输入相应的 数值。用户可以连续修改多个不同的变量。

在用户停止程序运行时,移动鼠标光标到要观察的变量上停大约1秒,就会弹出一个变 量提示对话框。

(4) 观察存储器区域: µVision 5 可以区域性地观察和修改所有的存储器数据。

μVision 5 把 MCS-51 单片机内核的存储器资源分成以下 4 个区域:①内部可直接寻址 RAM 区 data, IDE 表示为 D: xx;②内部间接寻址 RAM 区 idata, IDE 表示为 D: xx;③外部 RAM 区 xdata, IDE 表示为 X: xxxx;④程序存储器 ROM 区 code, IDE 表示为 C: xxxx。这4 个区域都可以在 View 菜单下的 Memory Windows 中观察和修改。在地址输入栏内输入待显示的存储器区的起始地址,就可看到各存储器单元的内容。若要修改存储器内容,可以用鼠标对准要修改的存储器单元, 右击, 从弹出的快捷菜单中选择 Modify Memory at 0x…命令, 在接着弹出的对话框文本输入栏内输入相应数值后按 Enter 键, 修改完成, 或者直接双击存储单元, 并输入相应数值即可。注意代码区数据不能更改。

5.2 Proteus 软件使用

Proteus 软件是由英国 Labcenter Electronics 公司开发的 EDA 工具软件,由 ISIS 和 ARES 两个软件构成,其中,ISIS 是一款便捷的电子系统仿真软件平台,ARES 是一款高级 的布线编辑软件。Proteus ISIS 运行于 Windows 操作系统上,是一款集单片机和 SPICE 分 析于一身,功能极其强大的仿真软件,可以仿真、分析(SPICE)各种模拟器件和集成电路。 Proteus ISIS 实现了单片机仿真和分析电路仿真相结合,具有模拟电路仿真、数字电路仿 真、单片机及其外围电路组成的系统的仿真、RS-232 动态仿真、I²C 调试器、SPI 调试器、键 盘和 LCD 系统仿真的功能,支持 68000 系列、8051 系列、AVR 系列、PIC12 系列、PIC16 系 列、PIC18 系列、280 系列、HC11 系列等单片机类型及各种单片机外围电流仿真。同时,该 软件可以提供全速、单步、设置断点等软件调试功能,并可以支持 Keil µVision 等第三方软 件的编译和调试环境。另外,Proteus ISIS 具有强大的原理图绘制功能。

接下来以 5.1 节的流水灯控制实例来完整地介绍 Proteus ISIS 的处理过程和基本操作。在 AT89C51 单片机小系统的基础上,P1 口连接 8 个发光二极管指示灯,编程实现流水灯的控制,从低位到高位轮流点亮指示灯,一直重复。在 Keil μVision 中编程序,形成 HEX 文件,在 Proteus 中设计硬件,下载程序,运行并查看结果。Proteus ISIS 的处理过程一般如下。

5.2.1 启动 Proteus

双击桌面上的 ISIS 7.8 Professional 图标,出现如图 5-14 所示的显示界面后,进入 Proteus ISIS 集成环境,如图 5-15 所示。



图 5-14 启动时显示的界面



图 5-15 Proteus ISIS 的工作界面

Proteus ISIS 的工作界面是一种标准的 Windows 界面,包括标题栏、主菜单、标准工具 栏、绘图工具栏、状态栏、对象选择按钮、方向控制按钮、仿真进程控制按钮、预览窗口、对象 选择器窗口、图形编辑窗口等。

5.2.2 新建电路,选择元件

新建电路,选择元件的具体过程如下:

(1) Proteus ISIS 软件打开后,系统默认新建一个名为 UNTITLED 的未存盘的原理图 文件,如图 5-16 所示。执行 File 菜单下的 Save 命令或者 Save as 命令,可以对该文件进行 存盘,默认扩展名。为了使用方便,通常与 Keil μVision 编写的程序放在同一文件夹中。



图 5-16 Proteus ISIS 窗口

(2) 在主要模型下选择 Component Mode 选择元件工具,然后再单击窗口右上角的按钮 P,如图 5-15 所示,打开如图 5-17 所示的元件选择窗口。

83			Pick Devices	? ×
Keywords:	Besuits (8)			AT89C51 Preview
a/89c51	Device	Library	Description	VSM DLL Model IMCS8051 DLL1
Match Whole Words?	AT89051 AT89051.BUS	M058051 MC58051	8051 Microcontoler (4/B code, 33MHz; 2/16-bit Times; UART) 8051 Microcontoler (4/B code, 33MHz; 2/16-bit Times; UART)	10 XTALI RODADO 70 RODADO 70 RODADO 70
<u>⊆</u> alegory:	AT89C51RB2 AT89C51RB2.BUS	MCS8051 MCS8051	8051 Microcontoller (16kB code, 48MHz, Watchdog Timer, 3x16-bit Timers, UART) 8051 Microcontoller (16kB code, 48MHz, Watchdog Timer, 3x16-bit Timers, UART)	11 XTALZ PU3A03 X PU3A03 X
(AI CMEQUICE) Microprocessor ICs	AT89C51RC2 AT89C51RC2.BUS	MCS8051 MCS8051	8051 Microcontoller (32kB code, 48MHz, Watchdog Timer, 3x16-bit Timers, UART) 8051 Microcontoller (32kB code, 46MHz, Watchdog Timer, 3x16-bit Timers, UART)	
	AT89C51RD2.BUS	MCS8051 MCS8051	8051 Microcontoller (54k8 code, 40MHz, Watchdog Timer, 3x16-bit Timers, UART) 8051 Microcontoller (54k8 code, 40MHz, Watchdog Timer, 3x16-bit Timers, UART)	#20x8 -21 #21x80 -22 #21x80 -27
				20 FIEL FLAND 11 21 ALE FLAND 22 21 ALE FLAND 22 22 FIEL FLAND 22 23 FIEL FLAND 22 24 FIEL FLAND 22 24 FIEL FLAND 22 25 FIEL FLAND
				1 112 P30810 11 2 P11 P33510 11 1 P12 P32610 11 1 P12
				1 P13 P33811 H 1 P13 P33911 H 1 P15 P3511 H 1 P1
				PCB Preview:
				0.6in
<u>Sub-category</u>				1.9in
Manufacture:				
				DIL40
				QK Cancel

图 5-17 元件选择窗口

(3) 在元件选择窗口的 Keywords 文本框中输入元件关键字,搜索元件。Proteus 中部 分常用的元件如表 5-1 所示,找到元件后,双击元件将元件添加到列表栏,如图 5-18 所示, 添加后的元件列于 DEVICES 元件列表栏中。实例中所用到的元件有 AT89C51 单片机、按 钮 BUTTON、晶振 CRYSTAL、发光二极管 LED-RED、电阻 RES 以及电容 CAP。



图 5-18 添加元件到 DEVICES 元件列表栏中

元件名称	中文名说明	元件名称	中文名说明
7407	驱动门	BATTERY	电池/电池组
1N914	二极管	CAP	电容
74LS00	与非门	CAPACITOR	电容器
74LS04	非门	CLOCK	时钟信号源
74LS08	与门	CRYSTAL	晶振
74LS390	TTL 双十进制计数器	FUSE	保险丝
7SEG	7段式数码管开始字符	LAMP	灯
LED	发光二极管	POT-HG	三引线可变电阻器
LM016L	2行16列液晶	RES	电阻
MOTOR	马达	RESISTOR	电阻器
SWITCH	开关	RESPACK	排阻
BUTTON	按钮	8051	51 系列单片机
Inductor	电感	ARM	ARM 系列
Speakers & Sounders	扬声器	PIC	PIC 系列单片机
ALTERNATOR	交流发电机	AVR	AVR 系列单片机

表 5-1 Proteus 中部分常用的元件

5.2.3 放置和调整元件

放置元件的过程如下:

(1) 选择 Component Mode 工具。

(2)单击 DEVICES 元件列表中的元件名称,选中元件,这时在预览窗口将出现该元件 的形状,此时选择方向工具,可改变元件的放置方向。移动鼠标到编辑窗口,单击,在鼠标指 针处会出现元件形状,再移动鼠标,把元件移动到合适的位置,单击,元件就被放在相应的位 置上。通过相同的方法,把所有元件放置到编辑窗口相应的位置,电源和地是在配件的终端 接口 **=** 中。本实例的放置情况如图 5-19 所示。



图 5-19 放置元件

元件放置后,可通过移动、旋转、删除、属性修改等操作对元件进行编辑。首先要选中元件。元件选择有以下几种方式:①单击选择;②对于活动元件,如按钮 BUTTON等,通过鼠标左键拖动选择;③对于一组元件的选择,可以通过鼠标左键拖动选择框内的所有元件,也可按住 Ctrl 键再用鼠标左键依次单击要选择的元件。选中元件后,用鼠标左键拖动所选元件可以移动元件;按键盘的 Delete 删除键或者在选中的元件上右击,从弹出的快捷菜单中选择 Delete Object 命令可以删除元件;在右键菜单中选择相应的旋转选项可以旋转元件或修改元件属性。不同元件的元件属性对话框不同,如图 5-20 所示为电阻属性对话框。

		Edit Compo	nent		
Component <u>R</u> eference:	R1		Hic	iden: 🥅	<u>0</u> K
Resistance:	10k		Hic	iden:	Help
Model Type:	ANALOG		✓ Hide Al	•	Cancel
PCB Package:	RES40	-	? Hide Al	· •	
				 	
				~	
Exclude from <u>S</u> imulati Exclude from PCB <u>La</u> Edit <u>a</u> ll properties as t	on yout ext	Hide common	shy <u>m</u> odule 1 pins		

图 5-20 电阻属性对话框

5.2.4 连接导线

元件放置并调整完成后,需要用导线将各元件连接,形成电路图。在 Proteus 中,元件 引脚间的连接有导线和总线两种连接方式。导线连接简单,但不适用于复杂电路,总线方式 连接复杂,但适合连线较多的情况,而且电路美观。

1. 导线连接方式

导线连接过程如下:

(1)把鼠标指针移动到第一个元件的连接点,鼠标指针前会出现"."形状,单击,这时会 从连接点引出一条导线。

(2)移动鼠标指针到第二个元件的连接点,在第二个元件的连接点上,鼠标指针前也会 出现"."形状,单击,则在两个元件连接上导线,这时导线的走线方式是系统自动的而且是走 直线,如果用户要控制走线路径,只需在相应的拐点处单击,如图 5-21 所示。



图 5-21 导线的连接

用户也可用工具(Tools)菜单下面的自动走线命令(Wire Auto Router)取消自动走线, 这时连接形成的就是直接从起点到终点的导线。另外,如果没有到第二个元件的连接点就 双击,则从第一个元件的连接点引出一段导线。

(3)导线加标签。也可通过加标签的方法进行导线连接。给导线加标签用主要模型中

的放置线标签 III 工具。处理过程如下:单击放置线标签 III 按钮,移动鼠标到需要加标签 的导线上时,鼠标指针前会出现"×"形状,单击,弹出编辑线标签窗口,如图 5-22 所示。在 String 窗口组合框中输入线标签名。在一个电路图中,标签名相同的导线在逻辑上是连接 在一起的。

	Edit Wire Label	
Label Style		
String:	Eotate Botate Middle	
	<u>D</u> K <u>C</u> ancel	

图 5-22 编辑线标签窗口

2. 总线方式

总线用于元件中间段的连接,便于减少电路导线的连接,而元件引脚端的连接必须用一般的导线。因此,使用总线时主要涉及绘制总线和导线与总线的连接。

(1) 绘制总线

通过用主要模型中的绘制总线(Buses Mode) + 工具来绘制总线。选中该工具后,移动鼠标到编辑窗口,在需要绘制总线的开始位置单击,移动鼠标,到结束位置再单击,便可绘制出一条总线。

(2) 导线与总线的连接

导线与总线的连接通常是从导线向总线方向连线,连接时一般有直线和斜线两种,如 图 5-23 所示,斜线连接时一般要取消自动走线。



总线绘制完成后,可以同时给总线中的一组信号线加标签,处理过程与导线加标签相同,标签采用例如A[0],A[1],…,A[7]的形式。此时,便给总线中的8根信号线加了标签, 8根信号线的标签名分别为A0,A1,…,A7。对于连接在总线上的导线,如果标签名相同,则它们在逻辑关系上是连接在一起的,如图5-24所示。



图 5-24 总线上信号线的连接

本实例中,电阻 R1 阻值设置为 1kΩ,R2~R9 是 LED 的限流电阻,阻值设置为 200Ω, 线路比较简单,采用导线方式连接,连接后的电路如图 5-25 所示。



图 5-25 导线连接图

5.2.5 给单片机加载程序

各元件连接完成后,就可以给单片机加载程序。加载的程序只能是 HEX 文件,可以在 Keil μVision 软件中来设计,形成 HEX 文件,具体见 5.1.5 节。程序文件通常和硬件电路 文件保存在同一文件夹下。 针对在 Keil μVision 中编译形成的名为 example. hex 的 HEX 文件,其加载过程如下: 在 Proteus 电路图中,选中单片机 AT89C51 芯片,单击(或从菜单中选择 Edit Properties 命 令),打开 AT89C51 的属性对话框,如图 5-26 所示。然后,在 Program File 框中选择 example. hex 文件,将其加载到 AT89C51 芯片中。

	Edit C	omponent	
Component <u>R</u> eference:	U1	Hidden: [
Component <u>V</u> alue:	AT89C51	Hidden [- Help
PCB Package:	DIL40	▼ ? Hide All ▼	Data
Program File:	example.hex	Hide All	Hidden Pins
Clock Frequency:	4MHz	Hide All	
Advanced Properties:			Lancel
Enable trace logging	▼ No	▼ Hide All ▼	
Other Properties:			
		~	
		•	
		<	
Exclude from Simulation	T Attach h	ierarchy <u>m</u> odule	

图 5-26 AT89C51 的属性对话框

5.2.6 运行仿真查看结果

程序加载完成后,可以通过运行按钮运行程序,并查看运行后的结果。本实例仿真结果如 图 5-27 所示。如果需要查看单片机的特殊功能寄存器或存储器中的内容,可以用暂停按钮暂停 程序,然后通过 Debug(调试)菜单下的相应命令打开特殊功能寄存器窗口或存储器窗口查看。



图 5-27 仿真结果

值得说明的是,在仿真调试时,如果程序出错,仿真后不能得到相应的结果,那么需要在 Keil μVision 中修改程序,程序修改后再对程序进行重新编译连接,形成 HEX 文件,但在 Proteus 中不需要再重新加载,可直接运行。

5.3 本章小结

本章以一个简单的实例为例,介绍了 Keil C51 和 Proteus 软件的工作环境和一些基本 操作。详细介绍了 Keil 和 Proteus 软件的开发环境和基本操作;应该熟练使用 Keil 软件进 行程序的编译调试;掌握 Keil 和 Proteus 软件的联调。

习题

简答题

- 1. 简要介绍 Keil µVision 5 的使用方法。
- 2. Keil µVision 5 环境下如何设置和删除断点?
- 3. 简要介绍在 Proteus 中单片机应用系统的仿真过程。
- 4. 在 Proteus 中,导线的连接方式有几种?
- 5. 在 Proteus 中,如何把程序加载到 51 单片机中?
- 6. 在 Proteus 中,仿真调试时如果程序出错,如何操作?