

项目 1 认识 C 语言

教学目的

- 了解 C 语言的发展历史及特点;
- 初步认识和掌握 C 语言的程序结构;
- 了解 C 语言程序开发软件;
- 熟练掌握 Keil 软件的使用。

重点和难点

- C 语言的特点;
- C 语言的程序结构;
- Keil 软件的使用;
- 搭建第一个工程项目。



项目任务

任务 1.1: 认识第一个 C 程序

任务 1.2: 用 Keil 软件搭建第一个工程项目



相关知识

要使用 C 语言编程,必须先了解 C 语言,并能熟练掌握 C 语言语法知识和 C 程序调试方法。本书将通过项目任务形式,由浅入深地介绍与 C 语言相关的语法知识和调试方法,使读者能熟练掌握如何用 C 语言编写程序以及如何上机调试 C 程序。

1.1 C 语言的发展

C 语言问世于 20 世纪 70 年代初,1978 年由美国电话电报公司(AT&T)贝尔实验室正式发布。同时由 B.W.Kernighan 和 D.M.Ritchie 合著了著名的《THE C PROGRAMMING LANGUAGE》一书,通常简称为《K&R》,也有人称之为《K&R》标准。但是,在《K&R》中并没有定义一个完整的标准 C 语言,后来由美国国家标准协会(American National Standards Institute)在此基础上制定了一个 C 语言标准,于 1983 年发布,通常称之为 ANSI C。

早期的 C 语言主要用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们认识,到了 20 世纪 80 年代,C 语言开始进入其他操作系统,并很快在各类大、中、小和微型计算机上得到广泛的使用,成为当代最优秀的程序设计语言之一。



C 语言功能强，使用灵活，编写自由、可移植性好，兼有高级语言和低级语言的优点，既适合编写应用程序，又适合开发系统软件，是国内外广泛流行的计算机高级语言。其实就 C 语言的特点来看，C 语言更适合于解决某些小型程序的编程。C 语言作为传统的面向过程的程序设计语言，在编写底层的设备驱动程序和内嵌应用程序时，往往是更好的选择。因此，C 语言在单片机嵌入式技术领域得到了广泛的应用，现在越来越多的工程技术开发人员使用 C 语言，用于 51 单片机的 C 语言就称为 C51 语言。目前，单片机应用系统开发中广泛使用 C 语言，在大型、复杂的单片机应用系统开发中都通过 C 语言来设计程序。

1.2 C 语言的特点

与其他计算机高级语言相比，C 语言具有它自身的特点。我们可以用 C 语言来编写科学计算或其他应用程序，但 C 语言更适合于编写计算机的操作系统程序以及其他一些需要对机器硬件进行操作的场合，有的大型应用软件也采用 C 语言进行编写。这主要是因为 C 语言具有很好的可移植性和硬件控制能力，表达和运算能力也较强。许多以前只能用汇编语言来解决的问题现在可以改用 C 语言来解决。概括起来说 C 语言具有以下一些特点。

1. 语言简洁、紧凑，使用方便、灵活

C 语言把高级语言的基本结构和语句与低级语言的实用性结合起来，可以像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元。

C 语言只有 32 个关键字，9 种控制语句，程序书写形式自由，源程序短。

2. 运算符丰富

C 语言的运算符包含的范围很广泛，共有 34 种运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理，从而使 C 语言的运算类型极其丰富，表达式类型多样化。

3. 数据结构丰富

C 语言具有现代化语言的各种数据结构，包括整型、浮点型、字符型、数组类型、指针类型、结构体类型、共用体类型等，能用来实现各种复杂的数据结构的运算，并引入了指针概念，使程序效率更高。

4. 具有结构化的控制语句，可进行结构化程序设计

C 语言是完全模块化和结构化的语言，具有结构化的控制语句，如 if-else 语句、while 语句、do-while 语句、switch 语句、for 语句等。C 语言用函数作为程序的模块单位，便于实现程序的模块化。

5. 允许直接访问物理地址，对硬件进行操作

C 语言允许直接访问物理地址，能进行位操作，可以直接对硬件进行操作，因此 C 语言具有高级语言的功能和低级语言的许多功能，可用来编写系统软件。这种双重性，使它既是成功的系统描述语言，又是通用的程序设计语言。



6. 生成的目标代码质量高，程序执行效率高

C 语言描述问题比汇编语言迅速，工作量小、可读性好，易于调试、修改和移植，而代码质量与汇编语言相当。

7. 可移植性好，易于实现模块化设计

C 语言在不同机器上的编译程序，大多数代码是公共的，所以 C 语言的编译程序便于移植。在一个环境上用 C 语言编写的程序，不改动或稍加改动，就可移植到另一个完全不同的环境中运行。

8. 语法限制不严格，程序编写自由

一般的高级语言语法检查比较严格，能检查出几乎所有的语法错误。而 C 语言允许程序编写者有较大的自由度，因此放宽了语法检查。如用户编写的用户自定义函数，位置可以放在主调函数的前面，也可以放在主调函数的后面，同时函数与函数间是平行的，互相独立的，可以放在程序的任意位置；又如整型数据与字符型数据以及逻辑型数据可以通用。这些灵活性必然给程序的编写带来极大的方便，使程序的编写自由空间大，且快速高效。

1.3 C 语言的程序结构

1.3.1 一个简单的 C 语言程序

C 语言是一种结构化程序设计语言，程序采用函数结构。

每个 C 语言程序由一个或多个函数组成，在这些函数中至少应包含一个主函数 `main()`，也可以包含一个 `main()` 函数和若干个其他的功能函数。不管 `main()` 函数放于何处，程序总是从 `main()` 函数开始执行，再到 `main()` 函数结束的。在 `main()` 函数中调用其他函数，其他函数也可以相互调用，但 `main()` 函数只能调用其他的功能函数，而不能被其他的函数所调用。功能函数可以是 C 语言编译器提供的库函数，也可以是由用户编写的自定义函数。

在编写 C 程序时，程序的开始部分一般是预处理命令，接下来是函数说明和变量定义，再往下就是对函数的定义等。一个简单 C 程序的流程结构框图如图 1-1 所示。

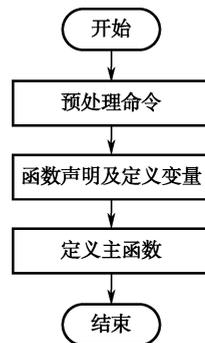


图 1-1 C 程序流程结构框图

为了让读者能更好地理解简单的 C 程序，现在对任务 1.1 源程序代码做一些程序说明。

参考程序段 1:

```

/*****
 *@File      chapter 1-1.c
 *@Function  在电脑屏幕上显示一行信息
 *****/
#include <reg51.h>                                     //预处理命令
  
```



```
#include <stdio.h>
void main() //定义主函数
{ //函数开始的标志
    SCON=0x52; //串口初始化，打开显示窗口
    TMOD=0x20;
    TH1=0xf3;
    TR1=1;
    printf("Hello! My name is C program.\n"); //输出所指定的信息
    while(1); //空循环，程序暂停
} //函数结束的标志
```

（1）#include <stdio.h>的作用

C 语言本身不提供输入和输出语句，输入和输出操作是由函数来实现的。在 C 语言的标准函数库中提供了一个名为“stdio.h”的头文件，它当中定义了 C 语言中的输入和输出函数，如 printf() 函数。当使用输入和输出函数时，需要先用预处理命令将该函数库头文件包含到源文件中。

在代码中加入头文件有两种书写方法，分别为“#include <stdio.h>”和“#include "stdio.h"”。

（2）printf() 函数的作用

printf() 函数称为格式输出函数，其功能是按用户指定的格式，把指定的数据或字符显示输出。该函数是 C 语言提供的标准输出函数，定义在 C 语言的标准函数库（stdio.h）中。

printf() 函数的一般形式为

```
printf("格式控制字符串", 输出列表);
```

格式控制字符串可由格式字符串和非格式字符串组成。

格式控制字符串是以“%”开头的字符串，输出列表在格式输出时才用到，它给出了各个输出项，要求与格式字符串在数量和类型上一一对应。相关知识将在后续内容里介绍。

非格式字符串在输出时原样输出，在显示中起提示作用。本例中用到的就是非格式字符串。“\n”是一个转义字符，其作用是换行。

（3）串口初始化作用

C 语言的一般 I/O 函数库中定义的 I/O 函数都是通过串行接口实现的，串行口的波特率由 MCS-51 单片机中的定时器/计数器 1 溢出率决定。在使用 I/O 函数之前，应先对 MCS-51 单片机的串行接口和定时器/计数器 1 进行初始化。串行口工作于方式 1，定时器/计数器 1 工作于方式 2（8 位自动重载方式），设系统时钟为 12MHz，波特率为 2400bps，则初始化程序如下：

```
SCON=0x52;
TMOD=0x20;
TH1=0xf3;
TR1=1;
```

有关串行口的详细内容将在后续课程中再做介绍，读者在这里只要会用串行口初始化程序即可，其目的是打开串行口窗口，提供程序输出显示界面。



(4) #include <reg51.h>作用及内容

由于在串行口初始化中用到了 SCON、TMOD 等几个特殊功能寄存器，使用它们时必须事先加以定义。Keil 编译器对这些特殊功能寄存器的定义都是放在一个名为 reg51.h 或 reg52.h 头文件里的，所以程序中需要先用预处理命令“#include <reg51.h>”将 51 单片机的特殊功能寄存器定义包含进来，这样使用 51 单片机的特殊功能寄存器才是合法的，否则编译器会报错。

打开 reg51.h 头文件可以看到这样一些内容：

```
/*-----  
REG51.H  
  
Header file for generic 80C 语言 and 80C31 microcontroller.  
Copyright (c) 1988-2002 Keil Elektronik GmbH and Keil Software, Inc.  
All rights reserved.  
-----*/  
  
#ifndef __REG51_H__  
#define __REG51_H__  
  
/* BYTE Register */  
sfr P0    = 0x80;  
sfr P1    = 0x90;  
sfr P2    = 0xA0;  
sfr P3    = 0xB0;  
sfr PSW   = 0xD0;  
sfr ACC   = 0xE0;  
sfr B     = 0xF0;  
sfr SP    = 0x81;  
sfr DPL   = 0x82;  
sfr DPH   = 0x83;  
sfr PCON  = 0x87;  
sfr TCON  = 0x88;  
sfr TMOD  = 0x89;  
sfr TL0   = 0x8A;  
sfr TL1   = 0x8B;  
sfr TH0   = 0x8C;  
sfr TH1   = 0x8D;  
sfr IE    = 0xA8;  
sfr IP    = 0xB8;  
sfr SCON  = 0x98;  
sfr SBUF  = 0x99;  
  
/* BIT Register */  
/* PSW */  
sbit CY   = 0xD7;  
sbit AC   = 0xD6;  
sbit F0   = 0xD5;
```



```
sbit RS1 = 0xD4;  
sbit RS0 = 0xD3;  
sbit OV = 0xD2;  
sbit P = 0xD0;
```

```
/* TCON */
```

```
sbit TF1 = 0x8F;  
sbit TR1 = 0x8E;  
sbit TF0 = 0x8D;  
sbit TR0 = 0x8C;  
sbit IE1 = 0x8B;  
sbit IT1 = 0x8A;  
sbit IE0 = 0x89;  
sbit IT0 = 0x88;
```

```
/* IE */
```

```
sbit EA = 0xAF;  
sbit ES = 0xAC;  
sbit ET1 = 0xAB;  
sbit EX1 = 0xAA;  
sbit ET0 = 0xA9;  
sbit EX0 = 0xA8;
```

```
/* IP */
```

```
sbit PS = 0xBC;  
sbit PT1 = 0xBB;  
sbit PX1 = 0xBA;  
sbit PT0 = 0xB9;  
sbit PX0 = 0xB8;
```

```
/* P3 */
```

```
sbit RD = 0xB7;  
sbit WR = 0xB6;  
sbit T1 = 0xB5;  
sbit T0 = 0xB4;  
sbit INT1 = 0xB3;  
sbit INT0 = 0xB2;  
sbit TXD = 0xB1;  
sbit RXD = 0xB0;
```

```
/* SCON */
```

```
sbit SM0 = 0x9F;  
sbit SM1 = 0x9E;  
sbit SM2 = 0x9D;  
sbit REN = 0x9C;  
sbit TB8 = 0x9B;
```



```
sbit RB8 = 0x9A;
sbit TI = 0x99;
sbit RI = 0x98;

#endif
```

以上内容涉及 51 单片机的内部结构，现在读者还不熟悉，没有关系。其实这里都是一些符号的定义，即规定符号名与地址的对应关系。

① 特殊功能寄存器定义。

例如：

```
sfr P1 = 0x90;
```

定义 P1 与地址 0x90 对应，P1 口的地址就是 0x90（0x90 是 C 语言中十六进制数的写法，相当于汇编语言中的 90H）。

从上面的头文件中读者可以看到一个频繁出现的词：`sfr`。`sfr` 并不是标准 C 语言的关键字，而是 Keil 软件里为能直接访问单片机中的 SFR 而提供的一个新的关键词，其用法如下：

```
sfr 变量名=地址值;
```

② 特殊位定义。

例如，符号 `P1_0` 用来表示 P1.0 引脚。

```
sbit P1_0=P1^0;
```

在 C 语言里，如果直接写 P1.0，C 编译器并不能识别，而且 P1.0 也不是一个合法的 C 语言变量名，所以得给它另起一个名字，这里起的名称为 `P1_0`，可是 `P1_0` 是不是就是 P1.0 呢？C 编译器并不这么认为，所以必须给它们建立联系。这里使用了 Keil C 的关键字 `sbit` 来定义，`sbit` 的用法有三种：

```
第一种方法：sbit 位变量名=地址值;
第二种方法：sbit 位变量名=sfr 名称^变量位地址值;
第三种方法：sbit 位变量名=sfr 地址值^变量位地址值;
```

如定义 PSW 中的 OV 可以用以下三种方法：

```
sbit OV=0xd2;      说明：0xd2 是 OV 的位地址值
sbit OV=PSW^2;    说明：其中 PSW 必须先用 sfr 定义好
sbit OV=0xD0^2;   说明：0xD0 就是 PSW 的地址值
```

因此这里用符号 `P1_0` 来表示 P1.0 引脚，也可以起类似 P10 的名字，只要下面程序中也随之更改就行了。

(5) “while(1);” 的作用

`while()` 是 C 语言里的循环控制语句，它的具体用法将在项目 5 里介绍，这里讲解为什么要加上这个循环语句。

当程序执行完 `printf()` 函数后，它还将向下执行，但后面的空间并没有存放程序代码，这时程序会乱运行，也就是说出现了跑飞现象。加上 “`while(1);`” 语句，是让程序一直停止在这里不再往下运行，即防止程序跑飞。



1.3.2 源程序的另外一种书写形式

C 语言采用模块化程序设计，一个模块完成一个特定的功能，即将能实现某一特殊功能的程序段独立成一个模块。程序形式上变了，但实现的功能是相同的。

现在可以把上面的源程序中的串行口初始化部分写成一个独立的子函数，源程序修改成如下形式。在本项目的后面进行上机调试时，读者会看到，其运行显示结果与当初的源程序段是一样的。

```
/******  
    *@File      chapter 1-2.c  
    *@Function  在电脑屏幕上显示一行信息  
*****/  
#include <reg51.h>           //预处理命令  
#include <stdio.h>  
void uart_init()           //定义串行口初始化函数  
{  
    SCON = 0x52;  
    TMOD = 0x20;  
    TH1 = 0xf3;  
    TR1 = 1;  
}  
void main()                //定义主函数  
{                          //函数开始的标志  
    uart_init();           //调用串行口初始化函数，打开显示窗口  
    printf("Hello! My name is C program.\n"); //输出所指定的信息  
    while(1);             //空循环，程序暂停  
}                          //函数结束的标志
```

1.3.3 C 语言的组成部分

由上述简单的 C 程序可知，C 语言程序由以下几部分组成。

(1) 一个源程序文件中可以包括三个部分。

① 预处理命令：总是放在源程序的开头，如#include <stdio.h>、#include <reg51.h>等。

② 函数声明或变量定义：在源程序中用到的所有函数都要事先声明，用到的所有变量都要先进行定义才能使用。

③ 定义函数：一个 C 程序是由一个或多个函数组成的，必须包含一个 main() 函数（且只能有一个），程序总是从 main() 函数开始执行的，每个函数用来实现一个或几个特定的功能。

(2) 一个函数由两部分组成。

① 函数的首部：即函数的第一行。包括函数名、函数类型、函数属性、函数参数（形参）名、参数类型。例如：

```
int max(int x, int y);
```

注意：一个函数名后面必须跟一对括号，即便没有任何参数也是如此。

② 函数体：即函数首部下面的大括号“{}”内的部分。函数体一般包括声明部分和执行部分。



声明或定义部分：在这部分中声明或定义所用到的函数或变量。

执行部分：由若干个语句组成。

(3) C 程序对计算机的操作由 C 语句完成。

C 程序书写格式是比较自由的，一行内可以写几个语句，一个语句可以分写在多行上，每个语句或数据定义的最后必须要有一个分号，分号是 C 语句的必要组成部分，代表语句结束，为清晰起见，习惯上每行只写一个语句。

(4) C 语言允许用两种注释方式，注释是为了增加程序的可读性。

① //：单行注释，只说明一行。

② /*.....*/：多行注释，说明多行，又称块注释。

1.4 Keil 开发软件

C 语言经不同的编译器编译后可用于多种 CPU，目前最流行的 C 语言编译器有 GCC、MS C、Turbo C、Microsoft Visual C++ 和 Keil 软件等。目前在工程上广泛使用 Keil 软件来开发应用程序，本书亦采用 Keil 软件来编译 C 程序。

Keil 软件是美国 Keil Software 公司出品的 51 系列兼容单片机 C 语言软件优秀开发系统，与汇编语言相比，C 语言在功能、结构性、可读性、可维护性上有明显的优势，用过汇编语言后再使用 C 来开发，体会更加深刻。它集编辑、编译、仿真于一体，支持汇编、PLM 语言和 C 语言的程序设计，界面友好，易学易用。

Keil 软件提供丰富的库函数和功能强大的集成开发调试工具，全 Windows 界面。另外，Keil 开发平台下生成的目标代码效率非常高，多数语句生成的汇编代码很紧凑，容易理解，在开发大型软件时更能体现高级语言的优势。

读者可以从网站（www.keil.com）上下载到该软件的安装包，其下载地址：<http://www.keil.com/download/product>。本书使用 Keil C 语言，以及 Keil 软件的可执行文件（文件名为“C 语言 v903”）。

1. 软件安装

(1) 执行 Keil μ Vision 4 安装程序，选择“Eval Version”版进行安装。

(2) 在后续出现的窗口中全部选择“Next”按钮，将程序默认安装在 C:\Program Files\Keil 文件夹中。

(3) 将光盘“头文件”文件夹中的文件复制到 C:\Program Files\Keil\C 语言\INC 文件夹里。

Keil μ Vision IDE 软件安装到计算机上的同时，会在计算机桌面建立一个快捷方式。安装其他版本软件与此类似。

2. 软件使用

(1) 新建工程（新建项目文件）

双击 Keil μ Vision IDE 的图标，启动 Keil μ Vision IDE 集成环境程序，进入 Keil μ Vision IDE 4 的主界面，如图 1-2 所示。

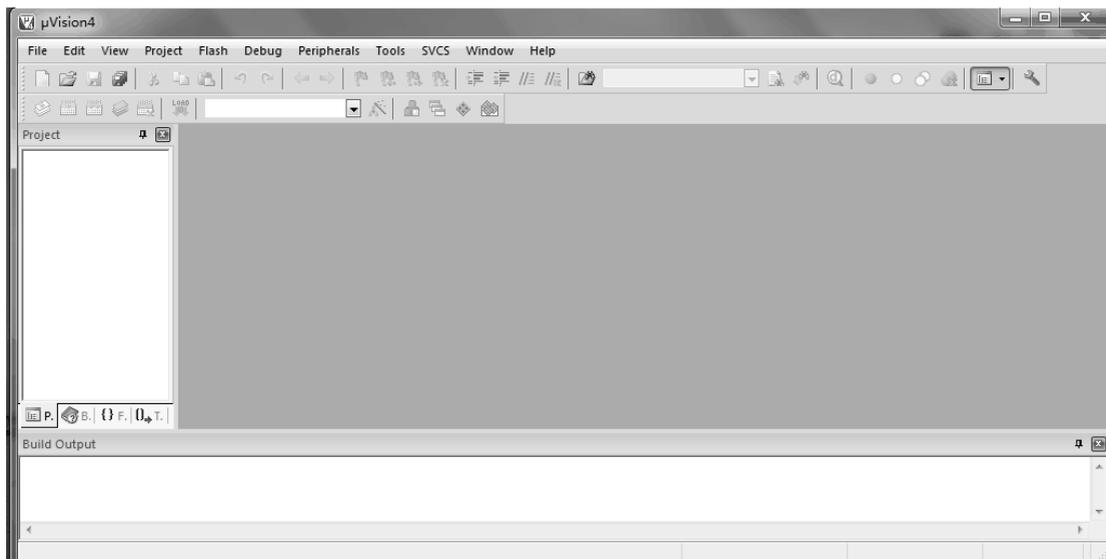


图 1-2 Keil μVision IDE 4 的主界面

在 Keil μVision IDE 4 主界面的顶部是 Keil μVision IDE 4 的主菜单栏，其中包含 11 个菜单选项：File（文件）、Edit（编辑）、View（查看）、Project（项目）、Flash（快闪）、Debug（调试）、Peripherals（外围接口）、Tools（工具）、SVCS（交换虚拟）、Window（窗口）、Help（帮助）。

主界面的左侧是项目工作区窗口，右侧是程序编辑窗口。项目工作区窗口用来显示所设定的工作区的信息，程序编辑窗口用来输入和编辑源程序。

单击“Project”→“New μVision Project”选项，如图 1-3 所示，出现“Create New Project”的对话框，如图 1-4 所示，在文件名处输入所建工程名称，选择需要保存路径，然后单击“保存”按钮。

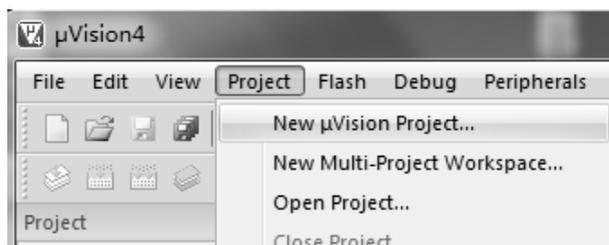


图 1-3 建立工程

工程保存之后，出现如图 1-5 所示对话框，在其中选择生产厂家及 CPU 型号。

本书所使用的是 Atmel 公司的 AT89C51，在列表中找到此款芯片，然后单击“OK”按钮，会弹出如图 1-6 所示对话框，询问是否要加载 8051 启动代码，对于初学者选择“否”。

至此，一个目标工程（项目文件）就建好了，如图 1-7 所示。

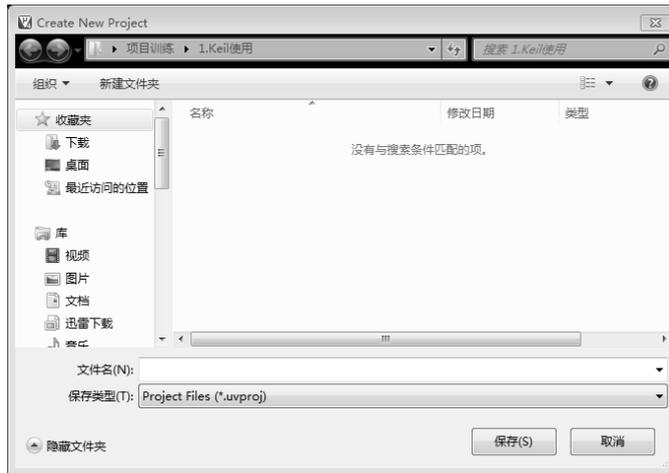


图 1-4 保存工程

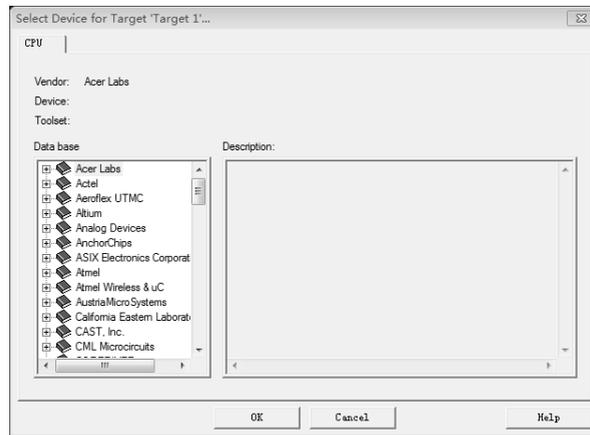


图 1-5 CPU 选择

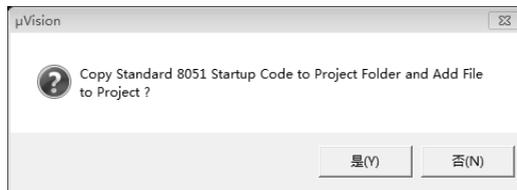


图 1-6 是否加载 8051 启动代码提示窗口

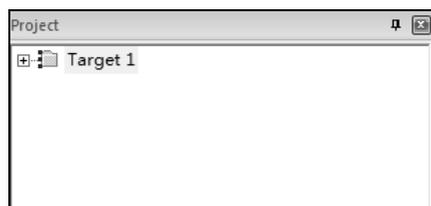


图 1-7 目标工程窗口



(2) 输入和编辑源程序

单击“File”→“New”选项，新建一个空白文本文档，单击“File”→“Save”选项，保存所建文本于项目文件夹中。

保存文件的格式：文件名.扩展名（即后缀），如果是 C 语言编写的程序则扩展名为 C，即“文件名.C”；若是汇编语言编写的程序则扩展名为 ASM，即“文件名.ASM”。

本书用 C 语言编写程序，以 1-1 为文件名，如图 1-8 所示。接下来就可以在新建文本中录入事先编写好的程序，如图 1-9 所示。

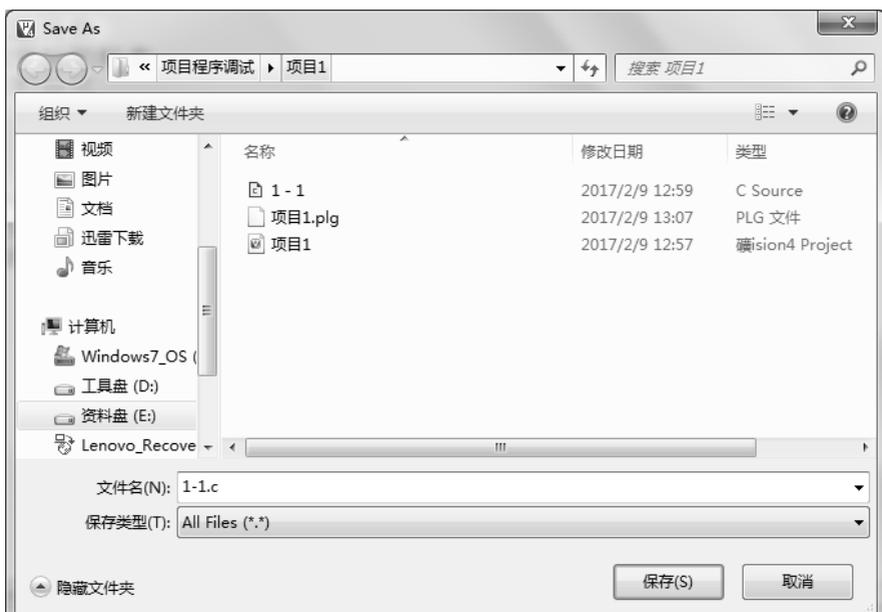


图 1-8 保存源文件

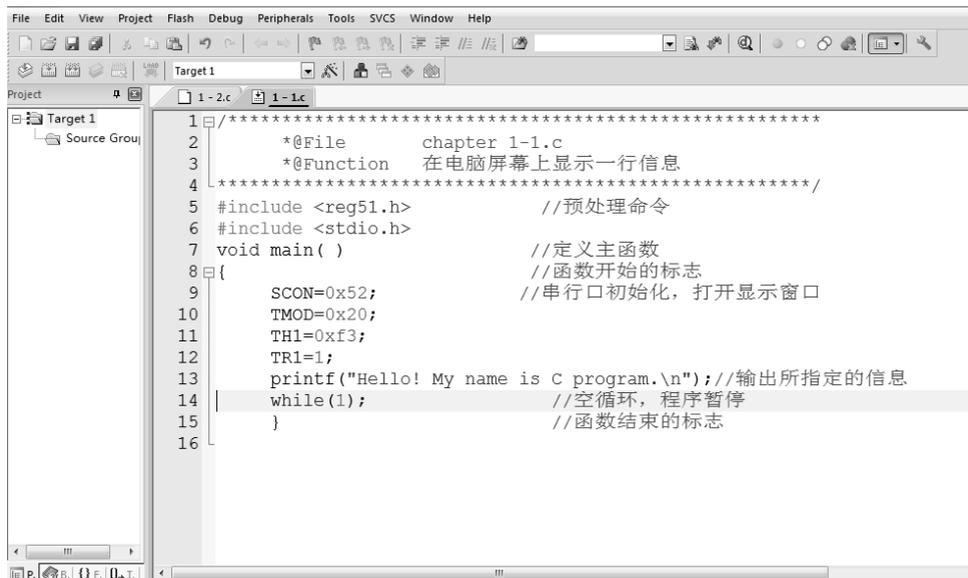


图 1-9 源程序录入



(3) 添加文件至工程项目

单击“Target 1”前的“+”号，出现下一层“Source Group 1”，右击“Source Group 1”出现下拉菜单如图 1-10 所示，选择“Add Files to Group ‘Source Group 1’”，出现图 1-11 所示对话框，选择刚才保存的文件，双击确认，即可将源程序加入到工程项目中，如还需加入程序则继续添加，若不需要添加，则关闭对话框即可。

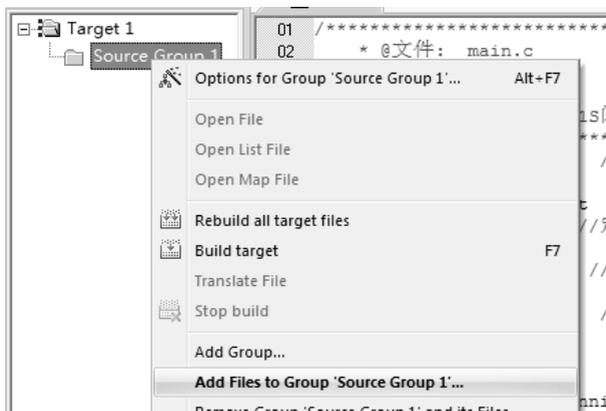


图 1-10 添加源程序

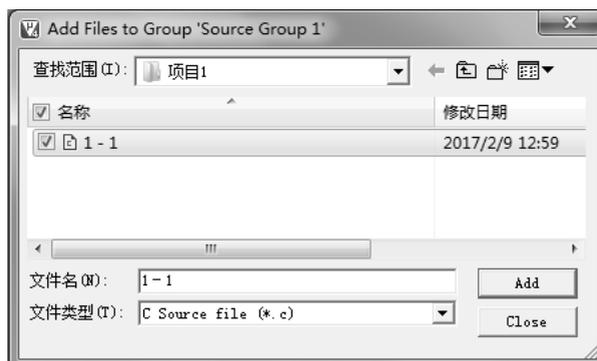


图 1-11 加入程序对话框

(4) 程序编译与连接

接下来的工作就是编译程序，将 C 语言程序编译为 CPU 所能识别的机器代码。Keil 软件使用的编译快捷图标如图 1-12 所示。

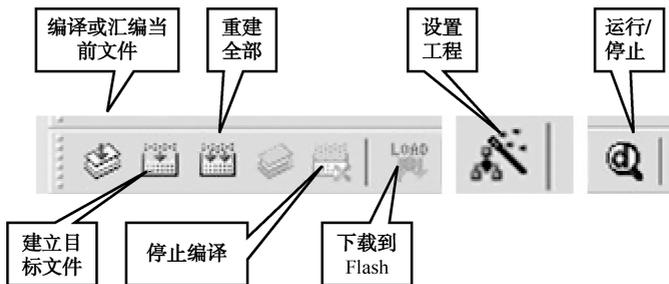


图 1-12 Keil 软件编译快捷图标



单击“Project”→“Build target”，则进行程序编译。如果有程序出错，则在主界面下方的“Build Output”窗口区域会有报错信息提示。

双击错误提示，则可以定位到程序的错误行或者错误行的上一行，然后可以对程序进行修改，之后重新编译，直到“Build Output”窗口出现“0 Error(s), 0 Warning(s)”信息，如图 1-13 所示。至此，一个源程序的编译与连接工作就完成了。

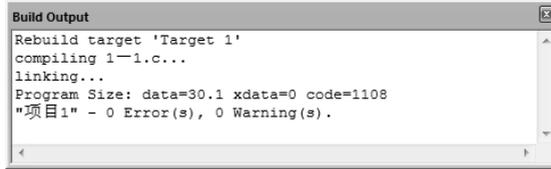


图 1-13 Build Output

(5) 程序的调试与运行

在程序编译与连接成功后，接下来进入程序的调试与运行状态。Keil 软件运行程序非常方便灵活，可以用连续执行键“Run”，也可以用单步执行键“Step”等。Keil 软件使用的运行快捷图标如图 1-14 所示。

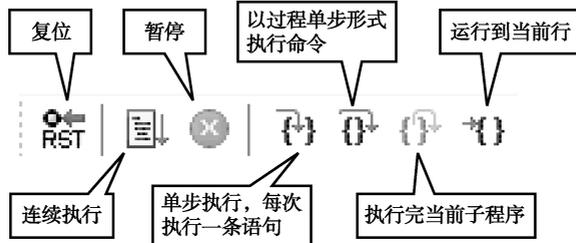


图 1-14 Keil 软件运行快捷图标

单击“Debug”→“Start/Stop Debug Session”或者“Start/Stop Debug Session”的快捷图标，进入程序调试状态，如图 1-15 所示。



图 1-15 进入调试



然后单击串行窗口“Serial Windows”，打开串行窗口，再单击“Debug”→“Run”或“Run”的快捷图标，在打开的“UART #1”串行窗口里就可以看到对应程序的运行结果，如图 1-16 所示。

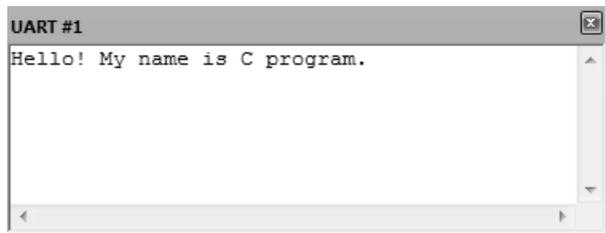


图 1-16 程序运行结果

(6) 使用 Keil 软件注意事项

- ① 在新建项目保存工程时，输入的工程名后不要加后缀。
- ② 保存源程序文件时，输入文件名后一定要加后缀，即“文件名.C”（C 语言程序）或“文件名.ASM”（汇编语言程序）。
- ③ 输入源程序时，务必将输入法切换到英文半角状态，输入的标点符号才正确。
- ④ 输入串行口初始化函数时，所有特殊功能寄存器一律用大写字母，如 SCON、TMOD、TR1、TH1 都要用大写字母，但要注意，TR、TH 后面跟的是数字 1 而不是字母 I 或 l。
- ⑤ 输入十六进制数时第一个字符是数值 0，而不是字母 O。

1.5 任务实现

通过上面的学习，读者已经认识了一个简单的 C 程序，初步了解了 C 语言的程序结构和 Keil 软件的使用。现在使用 Keil 软件完成本项目任务。

任务 1.1：认识第一个 C 程序

要求：在电脑屏幕上输出一行信息。

```
Hello! My name is C program.
```

参考程序段 2：

```

/*****
 *@File      chapter 1-2.c
 *@Function  在电脑屏幕上显示一行信息
 *****/
#include <reg51.h>           //预处理命令
#include <stdio.h>
void uart_init()           //定义串行口初始化函数
{
    SCON = 0x52;
    TMOD = 0x20;
    TH1 = 0xf3;
    TR1 = 1;
}

```



```
}  
void main() //定义主函数  
{ //函数开始的标志  
    uart_init(); //调用串行口初始化函数，打开显示窗口  
    printf("Hello! My name is C program.\n"); //输出所指定的信息  
    while(1); //空循环，程序暂停  
} //函数结束的标志
```

任务 1.2：用 Keil 软件搭建第一个工程项目

要求：完成任务 1.1 源程序，并上机调试与运行。

1. 上机调试与运行步骤

- (1) 建立工程项目；
- (2) 输入源程序并保存；
- (3) 添加源程序至工程项目下；
- (4) 编译连接程序；
- (5) 采用单步（Step）或连续（Run）方式运行程序；
- (6) 打开串行口窗口，观察程序运行结果。

2. 上机调试与运行结果

打开 Keil 软件，按上机调试与运行步骤分别调试运行参考程序段 1 和参考程序段 2，其运行显示结果如图 1-17 和图 1-18 所示。

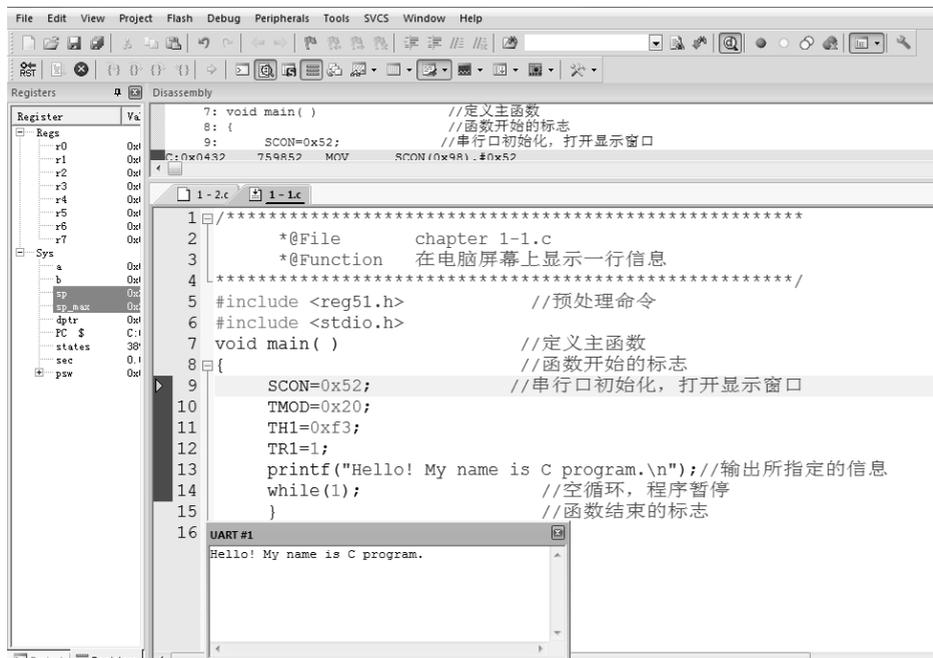


图 1-17 程序调试显示窗口 1

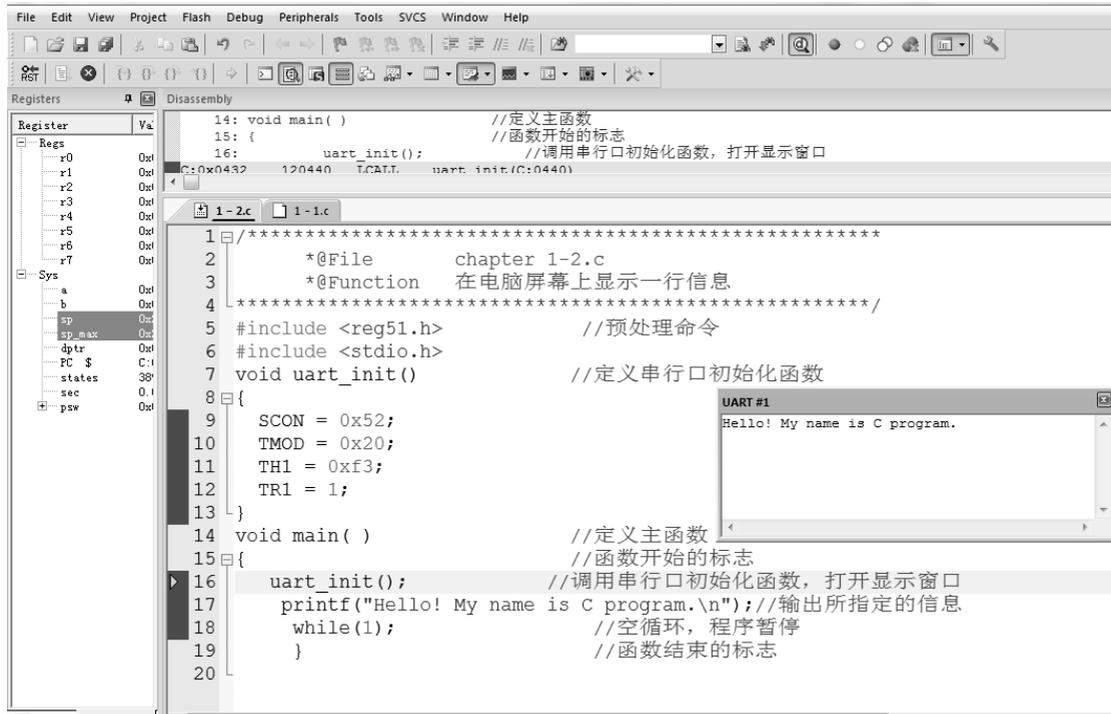


图 1-18 程序调试显示窗口 2



归纳与总结

C 语言是目前应用最广泛的计算机高级语言之一，用 C 语言既能编写系统软件，又能编写应用软件。C 语言采用结构化模块程序设计，编程灵活、自由度大，可移植性好。一个 C 程序是由一个或多个函数组成的，其中必须包含一个 main() 函数，程序总是从 main() 函数开始执行的，而不论 main() 函数在整个程序中的哪个位置。从输入到运行一个 C 程序，一般要经历编辑、编译、连接和调试运行四个步骤。本书选用的开发软件是 Keil 编译系统，它能提供丰富的库函数和功能强大的集成开发调试工具，同时又具有仿真功能，特别适合工程应用。



练习题

请在电脑屏幕上输出以下信息。

```
*****
Hello   Everyone !
My name is ×××(你的名字).
*****
```