## 二 C语言基础知识

### 1 C语言的构成

本章将介绍单片机C语言编程的各种组成元素。初次学习c语言时，还不知道程序是怎么一回事，就要求修改参数，战战噤噤，不知所错。希望亲爱的你认真研读，体会编程的原理和智慧，知其然，为我们的设计开发达到事半功倍。

一旦你学会编译程序了，只要达到目的，可以完全不理会我们所教的有限知识，毕竟我们不知道你怎样想，运用基本规则，你想怎样编译都可以。

SMARTLIT智能电子积木作为一类单片机，即使再简单，也是需要了解一些基础知识，为将来的设计思路保驾护航。

这里对C语言只是作一个基本了解，有志于深入C语言学习者，可以寻专门一本有关书籍学习。

单片机c语言组成

符号与表达式

语句

数据类型

基本规范

数值表达式

函数

### 2数据类型

变量不是一个具体的数值，而是一个空盒子，等待我们装进各种不同的数据。这个空盒子是多少，可以装多少数值，每个数值的范围是多少？c语言给了我们很大的权限去定义变量。

我们在使用程序前，先要定义变量。如unsigned int i；就是一个定义变量的语句。

#include<REG51.h>

Sbit LED =P1^7; //定义P1^7为LED控制口，低电平使能

void Delay (unsigned int a){ //输入变量值0~65535

unsigned int i;

while(--a !=0){

for(i=0;i<600;i++);

}

}

Void main （void）｛

While（1）｛

LED=~LED;

Delay（250）；

｝

｝

变量定义说明

|  |  |  |  |
| --- | --- | --- | --- |
| **数据类型** | **定义语句** | **占用空间** | **数值范围** |
| 位型 | bit | 1个位 | 0，1 |
| 无符号字符型 | unsigned char | 1个字节 | 0~255 |
| 有符号字符型 | signed char | 1个字节 | -128~127 |
| 无符号整数 | unsigned int | 2个字节 | 0~65535 |
| 有符号整数 | signed int | 2个字节 | -32768~32767 |
| 无符号长整数 | unsigned long | 4个字节 | 0~4294967295 |
| 有符号长整数 | signed long | 4个字节 | -2147483648~2147483647 |
| 浮点型 | float | 4个字节 | +-1.176E-38~+-3.40E+38（6位） |
| 双精度浮点型 | double | 8个字节 | +-1.176E-38~+-3.40E+38（10位） |

在编程中较常见的数据类型是bit、unsigned、unsigned int，这种数据软件基本都支持，可以直接使用。超值问题，即所定义的数值超出了数据类型的边界。在编译时，一般不会报错，但在程序运行时会出现很大问题。大家要注意，当改变一个变量的时候，要看看变量的数值，减少不必要的麻烦。

### 3基本规范

|  |  |  |
| --- | --- | --- |
| **基本规范** | **解释** | **示例** |
| ； | 语句间的分隔 | /\* \*/  #include <REG51.h> //  #define DY\_PORT P1 //  sbit DY\_KEY = p3^7 ; //  void main(void)  {  DY\_KEY = 1; //初始按键  While(1)  {  DY\_PORT = 0xff ;  **If(DY\_KEY = = 0)**  **{**  **DY\_PORT = 0x00;  }**  }  } |
| （ ） | 表达式/优先运算式 |
| [ ] | 数组地址 |
| ｛ ｝ | 语句1；语句2; |
| // | 单据注释 |
| /\* \*/ | 段落注释 |
| 函数声明 | 声明 |
| 变量声明 |
| 一维数组 | 数组 |
| 二维数组 |

数组，就是一组数据的集合，数组分一维数组、二维数组、三维数组和多位数组。

一维数组定义形式：

数据类型 数组名 [数量] = ｛ 数值1，数值2｝；

二维数组定义形式：

数据类型 数组名 [列数量] [行数量] = ｛ 数值1，数值2｝；｛ 数值3，数值4｝；

例如：

unsigned char code name [10] =//一维数组

{0x7F,0x02,0x0C,0x02, 0x7E, 0x3E, 0x41, 0x41, 0x41, 0x3E};

unsigned char code name [3] [6] = //二维数组

{0x7F,0x02,0x0C,0x02, 0x7E, 0x3E};

{0x7F,0x02,0x0C,0x02, 0x7E, 0x3E};

{0x7F,0x02,0x0C,0x02, 0x7E, 0x3E};

**编程风格**

虽然编程书写格式可以有很大的自由程度，但我们还是遵循C语言的标准，在格式上大家有个一致性，这是明智的选择，我们常常把相关的代码对齐，每一类括号上下对齐，并首行缩进2格或4格。

**大括号**

对于定义函数的时候，if语句和for语句的时候，放在语句或函数定义的同一行，或另取一行。

**留白**

编译器会忽略空格或另起行，为了很好看清楚代码，最好分清楚。

int a = 10；

int a=10；这样写都可以，只要保持前后一致就行。

**注释**

注释没有功能，只是代码中的一部部分，用来提醒自己和其他的代码人员，解释代码的意义。有两种方式注释：（1）单行注释用//开头，写在代码行的后面。（2）多行注释用/\*开始，并用\*/结束。

最后，养成良好的代码编写风格，将会节省更多的时间。

### 4数值表达式

|  |  |  |  |
| --- | --- | --- | --- |
| **数值表达式** | **英文表示符号** | **数码符号** | **进位规律** |
| 二进制 | B | 0、1 | 逢二进一 |
| 八进制 | O | 0、1、2、3、4、5、6、7 | 逢八进一 |
| 十进制 | D | 0、1、2、3、4、5、6、7、8、9 | 逢十进一 |
| 十六进制 | H | 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F | 逢十六进一 |
| 可以用电脑上的附件计算器，进行数值转换，详见附件B | | | |

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

为了有效使用内存，我们将内存数值预先设好，只需调用。

U8、u16、u32、s8、s16、s32。

例如，U8（定义 无符号8位= unsigned char 8）

U16（定义 无符号16位= unsigned short int 16）

U32（定义 无符号32位= unsigned int 16）

同理，

S8（定义 有符号8位= signed char 8）

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

### 5函数

|  |  |  |
| --- | --- | --- |
| Main函数 |  |  |
| 子函数 |  |  |

1. 函数是组成c程序的基本构件。
2. 一个函数由两部分组成：

函数的说明部分 函数体

函数体 {

说明部分

执行部分

}

1. 一个函数总是从main函数开始执行。
2. 语句以“；”号结束。
3. 书写格式自由。
4. /\*……\*/做注释

#include”stdio.h”

int max（int x，int y）/\*子函数\*/

｛

int z；

if（x>y）

z = x;

else

z = y;

ruturn z;

｝

Void main(){

int a,b,c;

scanf（”%d,%d”，&a,&b）；

c =max（a，b）;

Printf（“max = %d”,c）；

}

函数

|  |  |
| --- | --- |
| 所有函数形式 | 注释 |
| 无返回值，无传入值 | Void Founction（void）｛｝  Void是无返回值的意思。前面的void可以去掉，括号里的void删除（它只是提醒作用）。Founction是函数名。 |
| 无返回值，有传入值 | for（i=0; i<10; i++）｛｝ |
| 有返回值，无传入值 | U8 |
| 有返回值，有传入值 | （time ） |

### **6符号与表达式**

|  |  |  |
| --- | --- | --- |
| 符号与表达式 | 算术运算符 |  |
| 关系运算符 |  |
| 逻辑运算符 |  |
| 位操作 |  |
| 优先级别：由高到低 | 算术运算符>关系运算符>逻辑运算符>位操作 | |

(1)算术运算符

|  |  |  |  |
| --- | --- | --- | --- |
| 算术运算符 | = | 赋值 | C = a+b |
| + | 加法运算符，正值符号 | a+b |
| - | 减法运算符，负值符号 | a-b |
| \* | 乘法 | a\*b |
| / | 求模（除法） | 11/5 = 2 |
| % | 求余 | 11%5 = 1 |

注意

C语言里是用“==”来表示等于号。“=”是右边的算式的结果赋值给左边。例如c = 1+3，执行程序后，c的值就等于4。

自增或自减运算，i++或i--，相当于i=i+1或i-1。

(2)关系运算符

|  |  |  |  |
| --- | --- | --- | --- |
| 关系运算符 | < | 小于 |  |
| > | 大于 | c>a+bc>(a+b) |
| <= | 小于等于 |  |
| >= | 大于等于 |  |
| = = | 等于 | a==b<c a==(b<c) |
| != | 不等于 |  |

是对两个量进行“比较运算”

用关系运算符将两个表达式连接起来的有意义的式子。

如a ！= 0 ‘a’=’A’



(3)逻辑运算符

用逻辑运算符将关系表达式或逻辑量连接起来的有意义的式子。

|  |  |  |  |
| --- | --- | --- | --- |
| 逻辑运算符 | && | 与 | 都真则真 |
| || | 或 | 有真则真 |
| ！ | 非 | 真则假，假则真 |
| 优先级别：由高到低 | ！> && > || | | |

逻辑量：当一个量（可以是基本类型的常量或变量）用做判断时，c编译系统认为，非0为真，0为假。

如a =4则

！a 0

a&&-5 1

4||0 1





如：判断某年是否是闰年。（闰年：能被4整除，但不能被100整除，或能被400整除）

years%4==0&&years%100!= 0||years%400==0

(4)位操作

|  |  |  |  |
| --- | --- | --- | --- |
| 位操作 | & | 按位与 | 同就同，不同就假 |
| | | 按位或 | 有就真 |
| ^ | 按位异或 | 相异则真 |
| ~ | 按位取反 | 真则假，假则真 |
| << | 位左移 | 向左移动，溢出舍弃，空位补0 |
| >> | 位右移 |  |



