

传感器与综合控制技术

第三章：使用C语言控制核心系统

Introduction

- 绝大多数程序设计初学者忽视算法的重要性。实际上，没有正确的算法就没有正确的程序。通常程序设计在嵌入式系统中是实现测量与控制的关键因素。本章将说明如何使用算法设计、翻译成C语言、再对核心控制系统进行操作这三个步骤来解释本章的主要内容。
- 计算机语言（例如C语言）需要控制硬件工作，算法是语言编写过程中的关键问题。

本章要点

- 本章需要掌握的要点如下：
 - 集成开发环境的搭建（工程建立、工程配置、HEX文件生成）
 - 使用ISP软件下载可执行文件到核心硬件
 - 使用语言初步驱动硬件工作
- 本章需要了解的要点点如下：
 - 算法设计的思想
 - 算法翻译成程序的方法
 - 语言控制硬件初步体会

目录

- **3.1 算法概述**
- **3.2 软件环境搭建**
- **3.3 计算机语言与算法的配合**
- **3.4 程序设计语言C语言与单片机**

3.1 算法概述

- 算法通常被定义为“解题方案的准确而完整的描述，是一系列解决问题的清晰指令，算法代表着用系统的方法描述解决问题的策略机制”。
- 简单理解算法：算法就是为了能够使用计算机语言编程来解决问题的步骤。
- 算法在实际应用上有如下几个关键特征：
 - (1) 必须保证能够解决问题
 - (2) 必须是一系列步骤
 - (3) 必须能够通过某种方式转变为计算机程序

3.1 算法概述

- 3.1.1 简单算法设计思路

是否能够正确编写一个算法，决定了后面编写的程序是否正确。但是掌握算法的设计相对比较难，这是由于算法设计需要两方面的知识：

第一方面是能够找到一个解决问题的方法

第二方面是该方法能够使用计算机语言描述。

3.1 算法概述

- 3.1.1 简单算法设计思路

- **问题** 求圆的面积首先需要知道圆的半径，因此第一步就是给出圆的半径；然后就

- **初步** 用已知的半径计算 πr^2 值；最后这一步尤其重要：显示到屏幕上（这一步很关键，**首先**如果你不告诉计算机显示到屏幕上，计算机将什么都不做，那么计算的结果就不**！序是**知道是什么）。

整理：经过分析，我们整理初步的算法如下：

第一步：给出圆的半径 r

第二步：计算 πr^2 值

第三步：显示计算的结果

3.1 算法概述

- 3.1.1 简单算法设计思路
- 常见错误

第一步：计算
第二步：输入r值
第三步：输出结果

第一步：输入r值
第二步：输出结果
第三步：计算

3.1 算法概述

- 3.1.1 简单算法设计思路
- 最终算法

第一步：提示用户输入半径 r
第二步：从键盘输入半径 r
第三步：计算，并保存结果
第四步：显示该结果

3.1 算法概述

- 3.1.2 算法实现
- C语言程序框架

包含与定义部分;

```
#include <文件名.H> //include可能  
有多行  
或是#include “文件名.H”
```

```
#define 常量名 常量值  
全局变量声明/定义;  
函数声明/定义;
```

返回值 main (参数表)

```
{  
    变量声明/定义部分;  
  
    语句部分  
}
```

3.1 算法概述

- 3.1.2 算法实现
- 对应翻译法：一个算法步骤翻译成一个对应的C语言语句

第一步：提示用户输入半径r	对应翻译：	<code>printf (“请输入半径r: ”);</code>
第二步：从键盘输入半径r	对应翻译：	<code>scanf (“%f”, &r);</code>
第三步：计算，并保存结果	对应翻译：	<code>result = pi * r * r;</code>
第四步：显示该结果	对应翻译：	<code>printf (“\n圆的面积等于-%f”, result);</code>

3.1 算法概述

- 3.1.2 算法实现
- 完整的程序
- 请大家看正确算法
与错误算法的操作演示

```
#include<stdio.h>
//包含stdio.h库文件是为了能使用printf、scanf等这种标准函数
#define pi 3.14159
//注意define语句最后不要加分号，除非你有特殊用途
void main (void)
{
    //第一部分：变量定义部分
    float result, r;

    //第二部分：语句部分
    //第一步：提示用户输入半径r
    printf (“请输入半径r: ”);
    //第二步：从键盘输入半径r
    scanf (“%f”, &r);
    //第三步：计算，并保存结果
    result = pi * r * r;
    //第四步：显示该结果
    printf (“\n圆的面积等于: %f”, result);
}
```

3.2 软件环境搭建

在工程应用当中，单片机级的嵌入式开发主要采用C语言来进行设计。其软件设计环境不使用VC6.0环境而使用keil环境。本教材采用支持的51单片机的环境keil2版本进行介绍

【注意】 高版本的keil环境是MDK，例如：MDK4.72。并且注意到高版本的MDK也是可以用的

3.2 软件环境搭建

- 3.2.1 Keil集成开发环境简介
- 操作演示

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/416233053153010222>