



“十三五”职业教育国家规划教材

工作导向创新实践教材

C51 单片机应用与 C 语言 程序设计（第 4 版） ——基于机器人工程对象的项目实践

秦志强 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书以两轮小型移动机器人的制作与编程项目为主线,通过循序渐进地构建机器人的控制器和传感器电路,并对机器人进行编程和控制,将 C51 单片机的外围接口特性、内部结构原理、综合应用设计和 C 语言程序设计等知识和技能传授给学生,彻底打破了传统的先理论、后实验的教学方法和教学体系,解决了单片机原理与应用、C 语言程序设计等核心专业基础课程抽象、枯燥与教学效果差的难题。

本书可作为职业院校、职业本科院校的“单片机技术与应用”“嵌入式 C 语言程序设计”两门课程的学习教材和教学参考书,也可作为本科院校工程训练、电子制作的实践教材和相应专业课程的实验配套教材,同时还可供广大希望从事嵌入式系统开发和 C 语言程序设计的学生或者个人自学使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

C51 单片机应用与 C 语言程序设计:基于机器人工程对象的项目实践 / 秦志强编著. —4 版. —北京:电子工业出版社, 2022.1

ISBN 978-7-121-37929-1

I. ①C… II. ①秦… III. ①单片微型计算机—C 语言—程序设计—高等学校—教材 IV. ①TP368.1
②TP312.8

中国版本图书馆 CIP 数据核字(2019)第 253137 号

责任编辑:王昭松

印 刷:

装 订:

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×980 1/16 印张:15 字数:345.6 千字

版 次:2007 年 12 月第 1 版

2022 年 1 月第 4 版

印 次:2022 年 1 月第 1 次印刷

定 价:49.80 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888, 88258888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:(010) 88254015, wangzs@phei.com.cn, QQ83169290。

前 言

本书自 2007 年 12 月出版以来，已更新至第 4 版。作为国内第一本将单片机和 C 语言教学结合起来的开创性综合实践教材，本书因符合高职创新教学改革方向、教学效果经受住了实践的检验，深受高职院校师生的欢迎，至今已经在高职院校使用了近 15 年。

在这 15 年的时间里，本书不断修订和完善，先后入选国家“十二五”和“十三五”职业教育国家规划教材，这充分说明本书始终符合高职创新教学改革方向。

本次修订的重点是更新配套的教学机器人器材图片及其相关案例。相比原有的教学机器人教学板，新的教学板做了如下改进。

(1) 老版教学板需要使用专用的 USB ASP 下载器和 USB 转 232 数据线，使用起来非常不方便。而新的教学板只需要一根 USB 数据线即可实现程序的下载和程序运行时的数据交互，极大地简化了学习开发过程。

(2) 由于教学板所用核心芯片 AT89S52 的端口电流驱动能力弱，老版教学板不能直接驱动部分品牌的标准伺服电机，需要外接上拉电阻。新版教学板直接加入了上拉电阻，使其可以直接连接各种品牌的伺服电机，从而简化了学习开发过程。

(3) 新版教学板采用了 TQFP 封装工艺的 AT89S52 芯片，采用贴片方式安装到教学板上，比老版的 PLCC 芯片封装方式具有更高的可靠性。

(4) 新版教学板只安装了 AT89S52 芯片，无须再兼顾 AVR 单片机教学，使得教学板更加简洁、美观和可靠。

除了教学板的升级，对配套教学机器人的机械本体也进行了升级，由原来的一体化车体升级为标准金属模块搭建的车体，为后续的各种机械结构和电子模块的拓展提供了更多的便利。

此次教材和配套器材的升级，为更多的高职院校采用此套教学方案提供了方便，编著者也希望更多的高职院校能够采用这本教材，实现老师和学生能力的双提升！

本教材的修订由电子工业出版社编辑发起，在此特别感谢编辑的努力与坚持。另外，还要特别感谢全童科教（东莞）有限公司的同事们在搭建模型、拍摄和修改配图等方面所做的工作。

学习本书所需的教學板、机械本体和电子配件可以联系全童科教（东莞）有限公司（微信公众号：全童科教）购买。

学无止境，没有最好，只有更好！书中如有不妥之处，敬请读者批评指正。

编著者

2022 年 1 月

目 录

第 1 讲 从 Arduino 到 C51 单片机	(1)
单片机和微控制器	(1)
学习单片机的意义	(1)
C51 系列单片机	(2)
机器人与 C51 单片机	(3)
任务 1 获得软件	(4)
任务 2 安装软件	(5)
安装 Keil μ Vision4	(5)
任务 3 硬件连接	(5)
连接到计算机上	(5)
电源的连接	(6)
对教学板和单片机进行通电检查	(6)
对教学板和单片机进行通信连接	(6)
任务 4 第一个程序	(7)
创建与编辑你的第一个程序	(7)
第一个 C 语言程序: HelloRobot.c	(9)
编译你的第一个程序	(10)
程序调试	(11)
下载可执行文件到单片机	(12)
用串口调试软件查看单片机输出信息	(13)
HelloRobot.c 是如何工作的	(13)
printf 函数	(14)
“while(1);” 的作用	(15)
HelloRobotYourTurn.c 是如何工作的	(16)
任务 5 做完实验关断电源	(18)
第 2 讲 C51 接口与伺服电机控制	(20)
C51 单片机的输入/输出接口	(20)
任务 1 单灯闪烁控制	(21)
LED 电路元件	(21)
LED 电路搭建	(21)

例程: HighLowLed.c	(22)
HighLowLed.c 是如何工作的	(22)
任务 2 机器人伺服电机控制信号	(26)
例程: BothServoClockwise.c	(28)
任务 3 计数并控制循环次数	(29)
例程: ControlServoRunTimes.c	(31)
例程: BothServosThreeSeconds.c	(33)
任务 4 用计算机来控制机器人的运动	(34)
例程: ControlServoWithComputer.c	(34)
ControlServoWithComputer.c 是如何工作的	(35)
第 3 讲 C 语言函数与机器人运动控制	(38)
任务 1 基本运动动作	(38)
例程: RobotForwardThreeSeconds.c	(39)
RobotForwardThreeSeconds.c 是如何工作的	(39)
向后走、原地转弯和绕轴旋转	(40)
例程: ForwardLeftRightBackward.c	(41)
任务 2 匀加速/减速运动	(44)
编写匀加速运动程序	(44)
例程: StartAndStopWithRamping.c	(45)
任务 3 用函数调用简化运动程序	(47)
例程: MovementsWithFunctions.c	(50)
例程: MovementsWithOneFuntion.c	(52)
任务 4 高级主题——用数组建立复杂运动	(53)
字符型数据	(54)
数组	(55)
例程: NavigationWithSwitch.c	(57)
NavigationWithSwitch.c 是如何工作的	(59)
例程: NavigationWithValues.c	(60)
第 4 讲 C51 接口与触觉导航	(63)
触觉导航与单片机输入接口	(63)
任务 1 安装并测试机器人“胡须”	(63)
安装“胡须”	(64)
测试“胡须”	(65)
例程: TestWhiskers.c	(65)

位操作符	(66)
if 语句	(67)
? 操作符	(67)
任务 2 通过“胡须”导航	(70)
编程使机器人基于“胡须”导航	(70)
关系与逻辑运算符	(71)
例程: RoamingWithWhiskers.c	(72)
“胡须”导航机器人怎样行走	(75)
任务 3 机器人进入死区后的人工智能决策	(75)
编程逃离墙角死区	(75)
例程: EscapingCorners.c	(76)
EscapingCorners.c 是如何工作的	(79)
第 5 讲 C51 接口与红外线导航	(82)
使用红外线发射和接收器件探测道路	(82)
红外前灯	(82)
任务 1 搭建并测试红外发射和检测器对	(83)
元件清单	(83)
搭建红外前灯	(83)
测试红外发射和检测器对	(84)
例程: TestLeftIrPair.c	(85)
排错	(87)
函数延时的不精确性	(87)
任务 2 探测和避开障碍物	(88)
改变“胡须”程序, 使它适用于 IR 检测和机器人躲避	(88)
例程: RoamingWithIr.c	(89)
任务 3 高性能的 IR 导航	(93)
在每个脉冲信号之间采样以避免碰撞	(93)
例程: FastIrRoaming.c	(93)
FastIrRoaming.c 是如何工作的	(95)
任务 4 俯视的红外发射和检测器对	(97)
推荐材料	(97)
用绝缘带模拟桌子的边沿	(98)
编程检测边沿	(98)
例程: AvoidTableEdge.c	(99)

AvoidTableEdge.c 是如何工作的	(101)
第 6 讲 C51 定时器与机器人的距离检测	(104)
用同样的 IR 组电路检测距离	(104)
推荐工具和材料	(104)
任务 1 定时/计数器的运用	(104)
定时/计数器的控制	(105)
工作模式	(105)
定时/计数器初值的计算	(106)
例程: TimeApplication.c	(106)
TimeApplication.c 是如何工作的	(107)
中断	(108)
中断优先级	(108)
任务 2 测试扫描频率	(109)
红外检测器频率检测	(109)
用频率扫描法进行编程, 做距离检测	(110)
例程: TestLeftFrequencySweep.c	(110)
TestLeftFrequencySweep.c 是如何工作的	(112)
例程: DisplayBothDistances.c	(114)
任务 3 尾随小车	(114)
对尾随车编程	(116)
例程: FollowingRobot.c	(117)
FollowingRobot.c 是如何工作的	(119)
任务 4 跟踪条纹带	(121)
搭建和测试路径	(121)
需要的材料	(121)
测试条纹带	(121)
绝缘带路径排错	(123)
编程跟踪条纹带	(123)
例程: StripeFollowingRobot.c	(124)
第 7 讲 C51 单片机的 UART 与机器人串口通信	(126)
串口控制寄存器 SCON	(127)
什么是波特率	(127)
波特率的计算	(128)
RS232 电平与 TTL 电平转换	(128)

任务 1 编写串口通信程序	(129)
例程: uart.h	(129)
uart.h 是如何工作的	(132)
存储器结构	(132)
串口工作流程	(133)
例程: HelloRobot.c——printf("Hello,this is a message from your Robot\n")	(134)
例程: ControlServoWithComputer.c——scanf("%d",&PulseDuration)	(134)
第 8 讲 C51 单片机显示接口编程与机器人应用	(136)
LED 数码管	(136)
任务 1 数码管显示	(136)
元件清单	(138)
电路设计	(138)
共阴极数码管显示的源程序	(140)
电路连接	(141)
任务 2 认识 LCD 显示器	(142)
LCD 显示器	(142)
LCD 显示器与 C51 单片机的连接	(142)
LCD 接口说明	(143)
任务 3 编写 LCD 驱动程序	(145)
元件清单	(146)
电路连接	(146)
例程: LCDdisplay.c	(146)
LCDdisplay.c 是如何工作的	(150)
指针	(151)
任务 4 用 LCD 显示机器人运动状态	(152)
C 语言的编译预处理	(152)
例程: MoveWithLCDDisplay.c	(156)
MoveWithLCDDisplay.c 是如何工作的	(158)
第 9 讲 多传感器智能机器人	(160)
任务 1 多传感器信息与 C 语言结构体的使用和编程	(160)
结构体	(160)
例程: IRRoamingWithWithStructLCDDisplay.c	(164)
IRRoamingWithWithStructLCDDisplay.c 是如何工作的	(165)

任务 2 智能机器人的行为控制策略和编程	(169)
例程: NavigationWithSensors.c	(171)
NavigationWithSensors.c 是如何工作的	(173)
第 10 讲 机器人循线竞赛	(175)
任务 1 QTI 传感器及其通信接口	(176)
任务 2 安装 QTI 传感器到机器人前端	(177)
任务 3 编写 QTI 传感器的测试程序	(178)
测试程序: Test4QTI.c	(179)
Test4QTI.c 是如何工作的	(180)
任务 4 设计算法实现机器人无接触传感器游中国	(182)
例程: RobotTourChina.c	(183)
RobotTourChina.c 是如何工作的	(186)
执行调试	(187)
任务 5 修改算法实现机器人游中国	(188)
任务 6 用数组实现机器人游中国比赛	(193)
任务 7 改进运动执行程序, 提升执行的可靠性	(199)
函数	(203)
附录 A C 语言概要归纳	(208)
C 语言概述	(208)
数据类型、运算符与表达式	(208)
分支结构程序	(210)
循环控制	(212)
数组	(214)
函数	(215)
预处理命令	(215)
指针	(217)
结构体	(218)
位运算	(219)
附录 B 微控制器原理归纳	(221)
引言	(221)
一些概念	(221)
中央处理器 (CPU)	(222)
RAM 和 ROM	(223)
地址总线、数据总线和控制总线	(223)

微处理器和微控制器	(224)
附录 C 无焊料面包板	(225)
附录 D LCD 模块电路	(228)
附录 E 本书所使用的机器人零配件清单	(228)

电子工业出版社版权所有
盗版必究

第 1 讲 从 Arduino 到 C51 单片机



学习情境

通过《Arduino 机器人制作、编程与竞赛》初中级课程的学习，读者已经了解和掌握了采用 Arduino 微控制器制作小型机器人的技巧和编程方法。Arduino 是一个封装了一个 AVR 单片机的 8 位微控制器（所谓封装就是将单片机、只读存储器和晶振集成在一个模块上面），可直接采用 C/C++ 语言编程，编程时不必深入了解单片机的内部构造和工作原理，只要了解其输入/输出接口特性。采用 C/C++ 语言编程，使我们首先不必纠缠于复杂的硬件接口编程和编译过程，而只专注于智能程序的结构、逻辑设计及实现方法。这样做的好处是使学习变得简单方便、开发项目时快速高效；缺点则是成本较高、灵活性不强。

成本高和灵活性不强的原因是 Arduino 已经是一个通用的控制模块，经过了二次封装，自然就多了一道制造和开发成本。但是在许多大批量小型智能产品的开发过程中，往往要求我们直接采用单片机进行开发，因为这样不仅可以大幅降低生产成本，而且可以提高产品的可靠性和效率。本课程将引领大家从已经掌握的微控制器出发，深入微控制器内部，学习和掌握如何直接用单片机和 C 语言来开发智能产品。

单片机和微控制器

一台能够工作的计算机包括 CPU（Central Processing Unit，中央处理单元，用于运算、控制）、RAM（Random Access Memory，随机存储器，用于数据存储）、ROM（Read Only Memory，只读存储器，用于程序存储）、输入/输出设备（串口、并口等）。在个人计算机（PC）上，这些部分被分成若干块芯片或者插卡，安装在一个被称为主板的印制电路板上。而在单片机中，这些部分全部被做在一块集成电路芯片中，因此被称为单片机。单片机真正工作，还需要稳定的电源、晶振、外部存储器和编程调试接口，就像计算机工作也需要电源、晶振、硬盘或其他大容量外部存储器和操作系统一样。微控制器就是将单片机真正能够独立工作所需的电源适配器、晶振、外部存储器和串口转换电路等部分封装到一个模块上，这样微控制器就能够直接与 PC 连接进行编程开发，教学板上几乎没有任何其他的芯片和电路。

学习单片机的意义

与个人计算机、笔记本电脑相比，单片机的功能是很少的。实际生活中并不是任何需



要计算机的场合都要求计算机有很高的性能，如空调温度的控制、冰箱温度的控制等都不需要很复杂、很高级的计算机。关键看是否够用、是否有很高的性能价格比。

单片机具有体积小、质量轻、价格便宜等优势，已经渗透到我们生活的各个领域：导弹的导航装置、飞机上各种仪表的控制、工业自动化过程的实时控制和数据处理、广泛使用的各种智能 IC 卡、民用豪华轿车的安全保障系统、录像机、摄像机、全自动洗衣机、程控玩具、电子宠物等，更不用说自动控制领域的机器人、智能仪表和医疗器械了。

因此，单片机的学习、开发与应用将造就一批计算机应用、嵌入式系统设计与智能化控制的科学家和工程师。同时，学习使用单片机是了解通用计算机原理与结构的最佳选择。



嵌入式系统

嵌入式系统是指嵌入到工程对象中能够完成某些相对简单或者某些特定功能的计算机系统。与从 8 位单片机迅速向 16 位、32 位、64 位过渡的通用计算机系统相比，嵌入式系统有其功能的特殊要求和成本的特殊考虑，从而决定了嵌入式系统在高、中、低端系统三个层次共存的局面。在低端嵌入式系统中，8 位单片机从 20 世纪 70 年代初期诞生至今还一直在工业生产和日常生活中被广泛使用。

嵌入式系统嵌入到对象系统中，并在对象环境下运行。对象领域相关的操作主要是对外界物理参数进行采集、处理，对对象实现控制，并与操作者进行人机交互等。

鉴于嵌入式低端应用对象的有限响应要求、嵌入式系统低端应用的巨大市场，以及 8 位单片机具有的计算能力，可以预测在未来相当长的时间内，8 位单片机仍然会在嵌入式应用中扮演重要角色。

C51 系列单片机

一提到单片机，就会经常听到这样一些名词：MCS51、8051、C51 等，它们之间究竟是什么关系呢？

MCS51 是指由美国 INTEL 公司生产的一系列单片机的总称。这一系列单片机包括很多品种，如 8031、8051、8751 等，其中 8051 是最典型的产品。该系列单片机都是在 8051 的基础上进行功能的增减和改变而来的，所以人们习惯于用 8051 来称呼 MCS51 系列单片机。

INTEL 公司将 MCS51 的核心技术授权给了很多公司，所以许多公司都在做以 8051 为核心的单片机，当然，功能或多或少有些改变，以满足不同的需求。其中，较典型的一款单片机 AT89C51（简称 C51）是由美国 ATMEL 公司以 8051 为内核开发生产的。本书使用的 AT89S52 单片机就是在此基础上改进而来的。

AT89S52 是一种高性能、低功耗的 8 位单片机，内含 8KB ISP（In-system Programmable，系统在线编程）的可反复擦写 1000 次的 Flash 只读程序存储器，器件采用 ATMEL 公司的高密度、非易失性存储技术制造，兼容标准 MCS51 指令系统及引脚结构。在实际工程应用中，



功能强大的 AT89S52 已成为许多高性价比嵌入式控制应用系统的解决方案。

早期的单片机应用程序开发通常需要仿真机、编程机等配套工具，要配置这些工具需要一笔不小的投资。本书采用的 AT89S52 不需要仿真机和编程机，只要运用 ISP 电缆就可以对单片机的 Flash 反复擦写 1000 次以上，因此使用起来特别方便简单，尤其适合初学者使用，而且配置十分灵活，可扩展性特别强。

➤ In-system Programmable (ISP, 系统在线编程)

In-system Programmable 是指用户可以把已编译好的程序代码通过一条“下载线”直接写入器件的编程（烧录）方法，已经编程的器件也可以用 ISP 方式擦除或再编程。ISP 所用的“下载线”并非不需要成本，但相对于传统的“编程器”，其成本已经大大降低了。通常 Flash 型芯片会具备 ISP 下载能力。

本书将引导你运用 AT89S52 作为机器人的大脑来制作一款教育机器人，并采用 C 语言对 AT89S52 进行编程，使机器人完成下述 4 个基本智能任务。

- (1) 安装传感器以探测周边环境。
- (2) 基于传感器信息做出决策。
- (3) 控制机器人运动（通过控制电机带动轮子旋转）。
- (4) 与用户交换信息。

通过完成这些任务，你将在不知不觉中掌握 C51 单片机的原理与应用开发技术，以及 C 语言程序设计技术，轻松走上嵌入式系统开发之路。

为了方便单片机与电源、ISP 下载电缆、串口线，以及各种传感器和电机的连接，须制作一个电路板，将单片机贴装在该电路板上，并安装一块小的面包板，方便给机器人搭接各种传感器电路。本书将此电路板称为教学板，如图 1-1 所示。

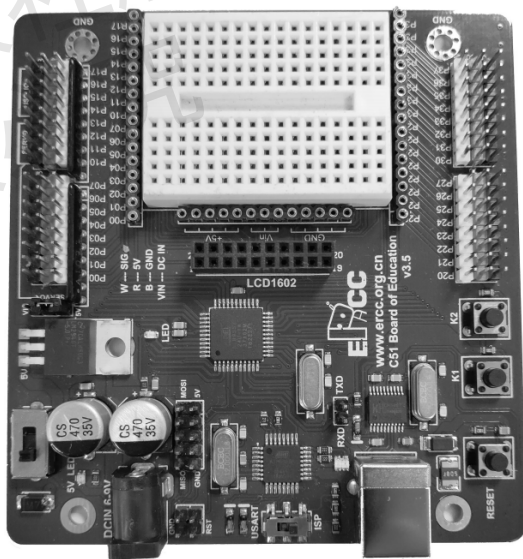


图 1-1 C51 单片机教学板

机器人与 C51 单片机

如图 1-2 所示是本书需要制作的小型机器人，它采用 AT89S52 单片机作为大脑，通过教学板安装在机器人底盘上。本书将以此机器人作为平台，完成上面提到的机器人所需具备的 4 种基本能力，使机器人具有基本的智能。

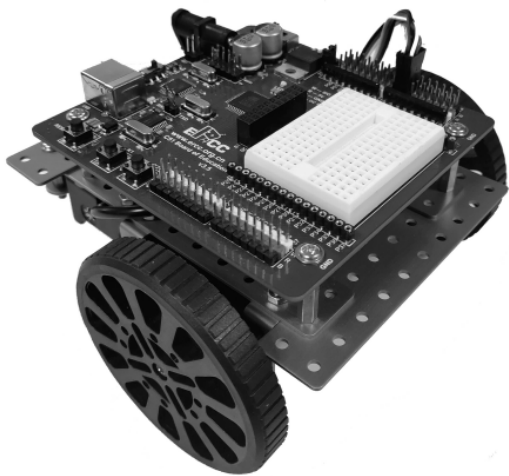


图 1-2 采用 C51 单片机制作的机器人

本讲首先通过以下步骤来介绍如何安装和使用 C51 单片机的 C 语言编程开发环境，如何用 C 语言开发第一个简单的机器人程序，并在机器人上运行编写的程序。本讲的具体任务包括：

- 寻找并安装开发编程软件；
- 连接机器人到电池或者供电的电源；
- 连接单片机教学板 ISP 接口到计算机，以便编程；
- 连接单片机教学板串行接口到计算机，以便调试和交互；
- 运用 C 语言初次编写少量的程序，运用编译器编译生成可执行文件，然后下载到单片机上，通过串口观察机器人上的单片机教学板的执行结果；
- 完成后，断开电源。

任务 1 获得软件

在本书的学习实践过程中，将反复用到 3 款软件：Keil μ Vision IDE 集成开发环境、Progisp 单片机 ISP 下载编程软件和串口调试软件。

1. Keil μ Vision IDE 集成开发环境

Keil C 语言是美国 Keil Software 公司出品的 51 系列兼容单片机的 C 语言软件开发系统。与汇编相比，C 语言在功能、结构性、可读性、可维护性上有明显的优势，因而易学易用。Keil 提供了包括 C 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（ μ Vision4）将这些部分组合在一起。运行 Keil 软件需要 WIN98、NT、WIN2000、WINXP、WIN2007、WIN2010 等操作系统。如果你使用 C 语言编程，那么 Keil 几乎就是你的不二之选。即使不使用 C 语言而仅用汇编语言编程，其方便易用的集成环境、强大的软件仿真调试工具也会令你事半功倍。

利用该开发环境，可以快捷、方便地建立面向各种单片机的 C 语言编程项目，编写 C 语言源程序，并将 C 程序编译和生成可下载到目标单片机的执行程序。

2. Progisp 单片机 ISP 下载编程软件

该软件是一款免费下载的 ISP 下载编程软件，不需要专门的安装即可使用，非常方便。



使用该软件，读者可以将C语言程序生成的可执行文件下载到机器人单片机上。使用时需要1个USB A转B信号线。

3. 串口调试软件

SerialDebugTool.exe 是本书使用的串口调试软件。该软件提供单片机与计算机的交互信息窗口，包括显示单片机发给计算机的信息窗口和计算机发给单片机的数据输入窗口。在硬件上，计算机至少要有串行接口或USB接口来与单片机教学板的串口连接。

任务2 安装软件

现在，如果读者已经从网站上获得了上述3个软件安装包，就可以开始安装软件了。软件的安装很简单，与安装其他软件过程一样。

安装 Keil μ Vision4

(1) 执行 Keil μ Vision4 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单击【Next】按钮，将程序默认安装在 C:\Program Files\Keil 文件目录下。

(3) 将安装包中“头文件”文件夹中的文件复制到 C:\Program Files\Keil\C51\INC 文件夹中。

Keil μ Vision4 IDE 软件安装到计算机上的同时，会在计算机桌面上建立一个快捷方式。

Proisp 单片机 ISP 下载编程软件与 SerialDebugTool.exe 串口调试软件都不需要安装，只需要将安装包中的这些软件复制到计算机上即可。

为了方便实用，建议建立桌面子目录将这三个工具软件全部放到里面。

任务3 硬件连接

C语言教学板需要连接电源来运行，同时需要连接到计算机上以便编程和交互。

连接到计算机上

C语言教学板通过USB A转B信号线连接到计算机上，程序的下载和信息的交互都通过该信号线完成。图1-3为本书使用的USB A转B信号线。



图 1-3 USB A 转 B 信号线

电源的连接

为了方便和节约电池，在一般的编程和调试时，建议使用一个 6V/2A 的电源适配器给 C 语言教学板供电。当需要机器人进行自主运动或者进行比赛时，使用 3.7V 锂电池给机器人供电。将锂电池装入专门的电池盒时，注意按照里面标记的电池极性（“+”和“-”）方向装入。

对教学板和单片机进行通电检查

教学底板上有一个三位开关（如图 1-4 所示），开关拨到“OFF”位时断开教学板电源。无论是否将电池组或者其他电源连接到教学底板上，只要三位开关位于“OFF”位，那么设备就处于关闭状态。

现在将三位开关由“OFF”位拨至“1”位，打开教学板电源，如图 1-5 所示。检查教学底板上绿色 LED 电源指示灯是否变亮。如果没有，检查电源适配器或者电池盒里的电池和电池盒的接头是否已经插到教学板的电源插座上。

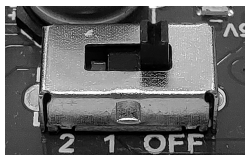


图 1-4 处于关闭状态的三位开关



图 1-5 处于“1”位状态的三位开关

开关“2”将会在后续的学习中用到。将开关拨至“2”位后，电源不仅给教学板供电，同时会给机器人的执行机构——伺服电机供电，同样，此时绿色 LED 电源指示灯仍然会变亮。

对教学板和单片机进行通信连接

教学板上有一个二位开关（如图 1-6 所示），当需要给单片机烧录程序时，应将开关拨到“ISP”位，接通单片机下载通道。如需使用教学板串口通信功能，则将二位开关由“ISP”位拨至“USART”位。

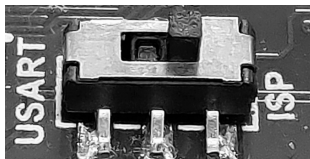


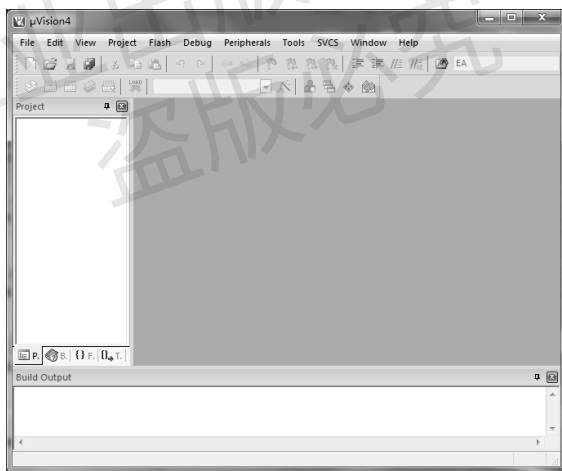
图 1-6 处于 ISP 下载状态的二位开关

任务 4 第一个程序

第一个 C 语言程序将告诉 AT89S52 单片机控制器，让它在执行程序时通过串口发送一条信息给计算机，在计算机的串口调试窗口中显示出来。

创建与编辑你的第一个程序

双击 Keil μ Vision4 IDE 的图标，启动 Keil μ Vision4 IDE 程序，打开如图 1-7 所示的 Keil μ Vision4 IDE 的主界面。通过“Project”菜单中的“New μ Vision Project...”命令建立项目文件，过程如下。

图 1-7 Keil μ Vision4 IDE 的主界面

(1) 单击“Project”菜单，会出现如图 1-8 所示的菜单画面，选择“New μ Vision Project...”命令，将出现如图 1-9 所示的对话框。在对话框目录栏中选择项目文件的存储目录，双击选中的目录或者单击“打开”按钮，或者直接在文件名中输入文件名，“打开”按钮会自动变为“保存”按钮。

(2) 在文件名中输入“HelloRobot”，保存至你想保存的位置（如 E:\C 语言程序设计\程序），可不用加后缀名。单击“保存”按钮，会出现如图 1-10 所示的窗口。

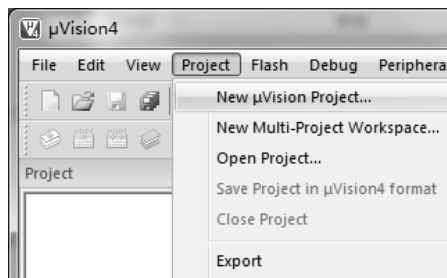


图 1-8 “Project” 菜单

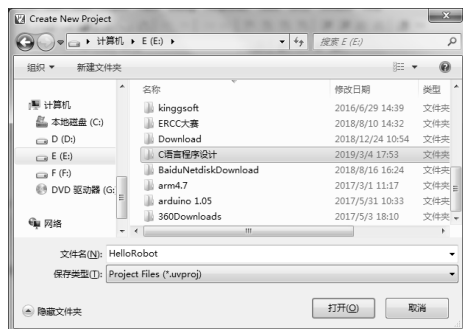


图 1-9 “Create New Project” 对话框

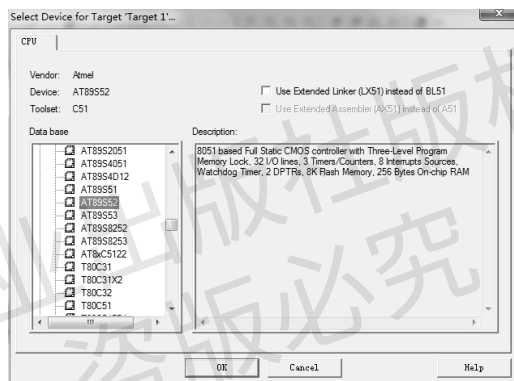


图 1-10 单片机型号选择窗口

(3) 这里要求选择项目芯片的类型。Keil μ Vision4 IDE 几乎支持所有的 51 核心单片机，并以列表的形式给出。本书使用的是 ATMEL 公司的 AT89S52，在 Keil μ Vision4 IDE 提供的数据库（Data base）列表中找到此款芯片，然后单击“OK”按钮，会出现如图 1-11 所示的窗口，询问是否加载 8051 启动代码，在这里我们选择“否”，不加载。（如果选择“是”，对你的程序没有任何影响。若你感兴趣，可选择“是”，看看编译器加载了哪些代码。）之后在界面左侧会出现如图 1-12 所示的目标工程窗口，此时就得到了目标项目文件 Target 1。

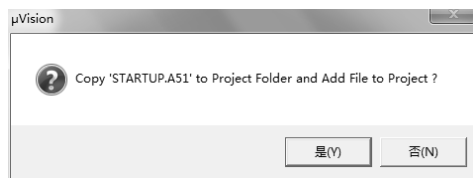


图 1-11 是否加载 8051 启动代码提示窗口

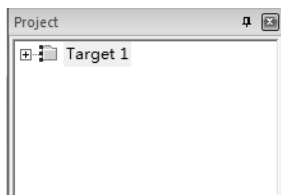


图 1-12 目标工程窗口



Target 1 项目文件创建后, 还只是一个框架, 紧接着需要向项目文件中添加源程序。Keil μ Vision4 支持编写 C 语言程序, 可以是已经建立好的 C 语言程序文件, 也可以是新建的 C 语言程序文件。如果是添加已建立好的 C 语言程序文件, 则直接用后面的方法添加到项目中; 如果是新建的 C 语言程序文件, 则先将程序文件存盘后再添加。

单击  按钮 (或通过 “File” $\rightarrow$ “New” 操作), 为该项目新建一个 C 语言程序文件, 保存后弹出如图 1-13 所示的对话框, 将文件保存在项目文件夹中 (保存的源文件名称可以和项目名称一样, 这样便于分辨哪个源文件属于哪个项目, 只是它们的扩展文件名不同), 在文件类型中填写 .c (这里 .c 为文件扩展名, 表示此文件类型为 C 语言源文件), 因为下面将采用 C 语言编写第一个程序。

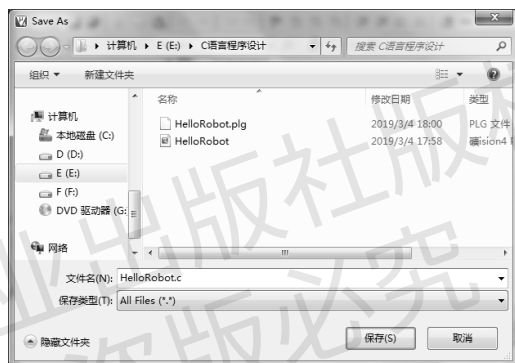


图 1-13 C 语言源文件保存对话框

第一个 C 语言程序: HelloRobot.c

```
#include<uart.h>
int main(void)
{
    uart_Init();                      //串口初始化
    printf("Hello,this is a message from your Robot\n");
    while(1);
}
```

将该例程输入 Keil μ Vision IDE 的编辑器, 并以文件名 HelloRobot.c 保存。按照下面的步骤将该文件添加到目标工程中。

(1) 单击图 1-12 中的 “+” 号, 将打开如图 1-14 所示的列表。

(2) 右键单击 “Source Group 1”, 在出现的菜单中

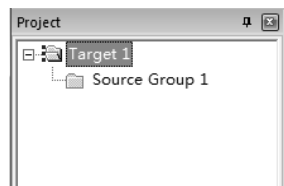


图 1-14 添加 C 语言文件到目标工程



选择“Add File To Group”→“Source Group 1”命令，出现“Add Existing Files to Group Source ‘Group1’”对话框。从中选择需要添加的程序文件，如刚才建立的 HelloRobot.c，单击“Add”按钮，把所选文件添加到项目文件中。

（3）程序文件添加到项目文件后，这时图 1-14 中“Source Group 1”的前面将出现一个“+”号；单击它，将出现刚才添加的源文件名，如图 1-15 所示（注意，图中显示的文件名是刚才输入的文件名）。

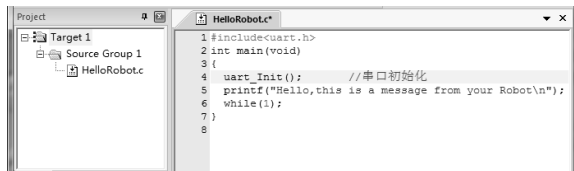


图 1-15 添加了 C 语言源文件的目标工程窗口和 C 程序源文件窗口

双击源文件名，即可显示和编辑源文件。

编译你的第一个程序

下面来生成下载需要的可执行文件。要生成可执行的.hex 文件，需要对目标工程“Target 1”进行编译设置。右键单击“Target 1”，选择快捷菜单中的“Option for target ‘Target 1’”命令。选择“Output”选项卡，勾选“Create HEX File”复选框，如图 1-16 所示，单击“OK”按钮，关闭设置窗口。单击 Keil μVision IDE 快捷工具栏中的按钮，Keil 的 C 编译器开始根据要生成的目标文件类型对目标工程项目中的 C 语言源文件进行编译。在编译过程中，我们可以观察到源文件有没有错误产生，如果没有错误产生，在 IDE 主窗口的下面的输出窗口（Output Window）中将出现如图 1-17 所示的提示信息，表明已成功生成了可下载的执行文件，并存储在 C 语言源程序存储的目录中，文件名就是 HelloRobot.hex。

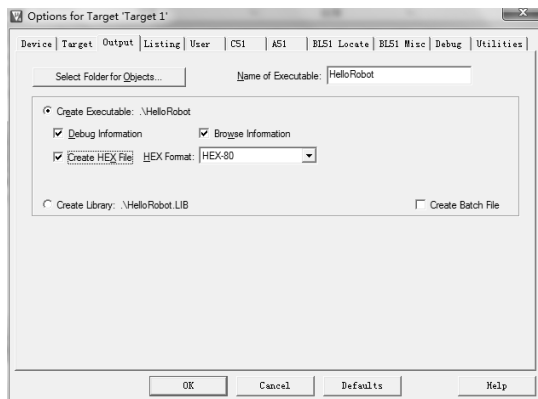


图 1-16 设置目标工程的编译输出文件类型

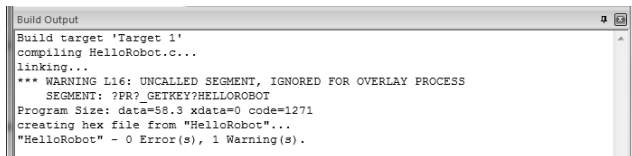


图 1-17 编译过程的输出提示信息

程序调试

如果程序在编译过程中出现了错误，就不能生成可下载的十六进制执行文件。C 语言的编写必须严格按照规定的规范，否则在编译过程中就会出现语法错误。例如，如果在录入程序时忘了在串口初始化语句的后面加分号：

```
uart_Init()
```

```
//串口初始化
```

编译时就会出现如图 1-18 所示的语法错误提示信息。

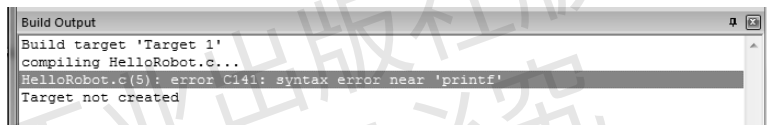


图 1-18 语法错误提示信息

错误信息提示首先给出发生错误的文件名称，随后括号中的数字表示错误发生的行数，这里是指第 5 行。提示信息“error C141:syntax error near ‘printf’”给出错误的编号（C141），这是一种语法错误，并告知错误发生在‘printf’附近。双击错误信息，编辑窗口中的光标会直接定位到错误位置。

C 语言对函数名称的大小写是敏感的，也就是同一个名字不同的大小写表示的是两个函数，标准函数大小写写错也会提示语法错误。例如，如果将 printf 写成了 Printf，那么编译时会出现如下警告和错误提示信息：

```
HELLOROBOT.C(5): warning C206: 'Printf': missing function-prototype  
HELLOROBOT.C(5): error C267: 'Printf': requires ANSI-style prototype
```

首先是警告程序中的 Printf 没有函数原型，随后就是错误信息，这个函数需要 ANSI 型函数原型。

如果字符串少加了一个双引号，例如，在要显示的字符串前缺少了双引号，即

```
printf(Hello,this is a message from your Robot\n");
```

则系统会给出 4 条错误信息：



```
HELLOROBOT.C(5): error C202: 'Hello': undefined identifier  
HELLOROBOT.C(5): error C141: syntax error near 'is'  
HELLOROBOT.C(5): error C103: '<string>': unclosed string  
HelloRobot.c(5): error C305: unterminated string/char const
```

由此可见，一个小小的语法错误可以导致编译时出现很多错误信息。因此，在编写和录入程序时一定要认真仔细，标点符号、大小写、名字都不能错。

总的说来，语法错误比较容易调试和修改，只要认真检查，就可以很快排除。特别是根据错误信息提示进行排除，速度会更快。

当程序没有任何编译和连接错误时，就能够顺利生成可以下载到单片机的十六进制文件了。

下载可执行文件到单片机

将教学板上的二位开关拨至“ISP”位，单击 Progisp 下载编程软件图标，打开下载编程软件窗口，如图 1-19 所示。通常不需要你更改任何选项，只需要在第一个列表框中选择正确的单片机型号，然后单击右上角的“调入 Flash”按钮，选择要下载的可执行 hex 文件——HelloRobot.hex，再单击“自动”按钮，即可开始下载。如果下载成功，则在窗口下部会显示“芯片编程”过程，最后显示“擦除，写 Flash，校验 Flash，成功”。程序在下载前会先自动擦除芯片中的原有程序。



图 1-19 Progisp 下载编程软件窗口

举一反三

如果读者学习过《Arduino 机器人制作、编程与竞赛》初中级课程，并已经掌握了采用



Arduino 开源硬件的 C++语言开发技能，请与前面介绍的 C 语言编程过程进行比较，看看有何不同。并思考一下，这些不同对于初学者而言各有何优缺点？是不是复杂很多？

用串口调试软件查看单片机输出信息

将教学板上的二位开关拨至“USART”位，双击串口调试软件 SerialDebugTool.exe，出现串口调试窗口，如图 1-20（a）所示。在左侧“通信设置”栏的串口号列表框中选择串口“COM××”后，单击下面的“连接”按钮。具体是哪一个串口，可以通过计算机的设备管理器来查看：打开计算机的设备管理器，通过插拔与 C 语言教学板连接的 USB 线找到相应的串口号。如果成功连接，“连接”按钮将变成“断开”按钮。

在“接收区”内你看到了什么？什么也没有！为什么呢？

注意：单片机的特点是只要里面有程序，一开机就开始执行。因此，从执行文件成功下载到单片机的那个时刻开始，程序就开始运行了。当你给单片机接上串口电缆通电时，单片机已经向计算机发送了信息。你错过了接收。怎么办呢？

教学板提供了“Reset”按钮，可以让下载到单片机内的程序重新运行一次。单击“Reset”按钮两次，是不是出现了如图 1-20（b）所示的画面呢？



（a）对串口调试软件进行通信设置



（b）查看输出信息

图 1-20 串口调试窗口

HelloRobot.c 是如何工作的

要讲清楚 C 语言的第一个程序是如何工作的，要比其他高级语言复杂很多。因为 C 语言是一个非常庞大的系统，是为开发大型程序而准备的。即使是最小的一个程序，其框架结构也很复杂。

例程中第一行代码是 HelloRobot.c 所包含的头文件。该头文件在编译过程中用来将下面程序中需要用到的标准数据类型和由 C 语言编译器提供的一些标准输入/输出函数、中断服务



函数等包括进来，生成可执行代码。头文件中可以嵌套头文件，同时也可以直接定义一些常用的功能函数。本例程中的头文件 `uart.h` 在本书的后续任务中都要用到，它包含了本例程中及后面的例程中都要用到的 `uart_Init()` 函数的定义和实现代码。当然，它也将 C 语言的标准输入/输出函数定义和实现包含了进来，如本例程中的 `printf` 函数。

下面先介绍函数的概念。一个较大的 C 语言程序一般分成若干个模块，每个模块实现一定的功能，被称为函数。任何一个 C 语言程序本身就是一个大的函数，该函数以 `main` 函数作为程序的起点，通常称之为主函数。主函数可以调用任何子函数，子函数之间也可以相互调用（但是不可以调用主函数）。函数定义的一般格式为：

函数返回值的类型 函数名（形式参数 1，形式参数 2，……）

第二行就是程序的入口 `main` 函数。`main` 前面的 `int` 用于指定 `main` 的函数返回值类型为整数类型，括号中 `void` 或无内容表示没有形式参数。每个函数的主体都要用“{}”括起来。

函数的具体应用将在第 3 讲中详细介绍。

`main` 函数主体中有两行语句：第一行是串口初始化函数 `uart_Init()`，用来规定单片机串口是如何与计算机通信的。有兴趣的读者可以打开 `uart.h` 头文件，看看该函数是如何实现的。如果其中有很多内容看不懂，不要紧，记住这个函数的功能就行，以后再逐步学习和理解。这行语句中“//”后的是注释。注释是一行会被编译器忽视的文字，不会被编译，仅为了让自己或者别人阅读程序时理解起来更方便。函数体中的第二行语句 `printf` 命令是要单片机通过串口向计算机发送一条信息。

printf 函数

`printf` 函数被称为格式输出函数，其功能是按用户指定的格式，把指定的数据输出显示。该函数是 C 语言提供的标准输出函数，定义在 C 语言的标准函数库中，要使用它，必须包括定义标准函数库的头文件 `stdio.h`。由于在 `uart.h` 头文件中包括了 `stdio.h`，因此本例程无须另外包括该头文件。`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，它给出了每个输出项，要求与格式字符串在数量和类型上一一对应。

非格式字符串在输出时原样输出，在显示中起提示作用。例程中用到的就是非格式字符串。

“\n”是一个向调试终端发送回车命令的控制符。也就是说，当单击“Reset”按钮再次运行程序时，将在下一行显示“Hello, this is a message from your Robot”；如果没有“\n”，则会在上一语句的结尾，即“Robot”后面接着显示。



“while(1);”的作用

while 是 C 语言里的循环控制语句，它的具体语法规则将在第 3 讲里介绍，这里仅解释为何要加上这个循环。

while(1)实际上是一个死循环。当 hex 文件被加载到单片机 Flash 存储器上时，是从头开始往下加载的。那么，当你把 hex 文件加载上去时，填满了整个 Flash 空间吗？当然没有！那么，当程序执行完 printf 函数之后，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，也就是发生了跑飞现象。加上 while(1);这个死循环语句，让程序一直停止在这里，就是为了防止程序跑飞。



该你了——例程：HelloRobotYourTurn.c

```
#include<uart.h>
int main(void)
{
    int i;
    uart_Init();
    i=7*11;
    printf("What's 7 X 11?\n");
    printf("The answer is :%d\n",i);
    while(1);
}
```

按照上述方法建立新的项目，输入程序 HelloRobotYourTurn.c 并运行，查看输出结果，是否与图 1-21 一样？



图 1-21 例程 HelloRobotYourTurn.c 输出结果



HelloRobotYourTurn.c 是如何工作的

在介绍 main 函数内容之前，先讲解一下 C 语言的一些基本知识。

C 语言数据类型

C 语言有 5 种基本数据类型：字符型、整型、单精度实型、双精度实型和空类型。这些数据类型的长度和范围会因处理器的类型和 C 语言编译程序的实现而有所不同，对于 Keil 51 产生的目标文件，表 1-1 给出了 3 种常用数据类型在书中用到的长度和范围。

表 1-1 常用数据类型的长度和范围

类 型	长 度（单位 bit）	范 围
char	8	-128~+127 即 $-2^7 \sim (2^7-1)$
int	16	-32768~+32767 即 $-2^{15} \sim (2^{15}-1)$
float	32	$-3.4 \times 10^{-38} \sim 3.4 \times 10^{38}$

标志符

在 C 语言中，标志符是对变量、函数名和其他各种用户定义对象的命名。标志符的长度可以是一个或多个字符。绝大多数情况下，标志符的第一个字符必须是字母或下画线，随后的字符必须是字母、数字或下画线（某些 C 语言编译器可能不允许以下画线作为标志符的起始字符）。表 1-2 是一些正确或错误标志符命名的实例。

表 1-2 正确或错误标志符命名的实例

正确形式	错误形式
count	2count
test23	hi!there
high_balance	high..balance

常量

C 语言中的常量是不接受程序修改的固定值，常量可以为任意数据类型，如下所示：

```
char 'a'、'9'
int 21、-234
```

变量

在程序中可以改变的量称为变量。一个变量应该有一个名字（标志符），在内存中占据



一定的存储单元，在该存储单元中存放变量的值。请注意区分变量名和变量值这两个不同的概念。所有C语言变量必须在使用之前定义。定义变量的一般形式是：

```
type variable_list;
```

这里的 **type** 必须是有效的C语言数据类型，**variable_list**（变量表）可以由一个或多个由逗号分隔的标志符名构成。下面给出一些定义的范例：

```
int i,j,k;  
char 'x', 'y', 'z';
```



注意：C语言中变量名与其类型无关。

运算符

C语言有三大运算符：算术、关系与逻辑、位操作。另外，C语言还有一些特殊的运算符，用于完成一些特殊的任务。

算术运算符

表1-3给出了C语言允许的算术运算符。在C语言中，运算符“+”、“-”、“*”和“/”的用法与大多数计算机语言相同，几乎可以用于C语言内定义的任何数据类型。

表 1-3 算术运算符

算术运算符	用 处
+	加法
-	减法
*	乘法
/	除法

表达式

表达式由运算符、常量及变量构成。C语言的表达式遵循一般代数规则。

C语言规定，任何表达式在其末尾加上分号就构成了语句。

赋值运算符

赋值运算符记为“=”。由“=”连接的式子称为赋值表达式，其后加分号构成赋值语句，其一般形式为：

```
变量=表达式;
```



现在来看看 main 函数是如何工作的。

```
int i;
```

定义了一个整型变量 i，i 即是变量的标志符，分号表示结束。

```
uart_Init();
```

与上一个例程一样，规定单片机串口如何与 PC 通信。

```
i=7*11;
```

将表达式“7*11”的值赋给变量 i，也就是说变量 i 的值为 77。

```
printf("What's 7 * 11?\n");
```

输出“‘What’s 7 * 11?’”，这里 printf 的用法与上一个例程一样。

```
printf("The answer is :%d\n",i);
```

这里用到了 printf 函数的格式字符串输出。%d 指定输出数据的类型为十进制整数。printf 函数首先输出“The answer is :”；然后它遇到了“%d”，表示将后面输出列表中的变量以十进制的形式输出，即将变量 i 以“77”的形式输出；最后的输出结果为：

```
The answer is :77
```

最后一条语句 while(1);也起到在上例中同样的作用——防止程序跑飞。

任务 5 做完实验关断电源

把电源从教学板上断开很重要，原因有几点：第一，在系统不使用时及时断开电源，有利于节约电能，电池可以用得更久；第二，在以后的实验中，你将在教学板的面包板上搭建电路，搭建电路时，应使面包板断电。如果是在教室做实验，老师可能会有额外的要求，如断开串口电缆、把教学板存放到安全的地方等。总之，做完实验后最重要的一步是断开电源。

断开电源比较容易，只要将三位开关拨到“OFF”位即可。



工程素质和技能归纳

1. C51 系列单片机 Keil μ Vision IDE（集成开发环境）软件和 Progisp 单片机 ISP 下载编程软件的下载和安装。
2. 机器人用 C51 教学板与计算机或者笔记本电脑的连接。



3. 如何在集成开发环境中创建目标工程文件，并添加和编辑C语言源程序。
4. C语言程序的编译和下载。
5. 串口调试终端的使用。
6. C语言基本知识：基本数据类型、常量、变量、运算符、表达式。
7. printf格式输出函数的使用。



科学精神的培养

1. 比较 Keil μ Vision IDE 与 Arduino IDE 开发环境的优缺点，找出它们的共同特点。
2. 比较第一个C语言程序与第一个Arduino程序的异同，找出它们的共同点。
3. 比较 Arduino 的 printf 函数和 Keil C 的 printf 函数的异同。
4. 查找 C 语言的标准输入/输出库函数，了解 printf 的总体功能。本讲中用到了它的两个格式符和控制符。
5. 查阅参考书，了解其他数据类型、算术运算符知识。