



第1章 单片机和嵌入式系统基础知识

何宾

2019.03



8051单片机硬件开发平台

本书所介绍的三款8051单片机硬件开发平台均采用了STC公司最新的硬件可仿真的IAP15W4K58S4。

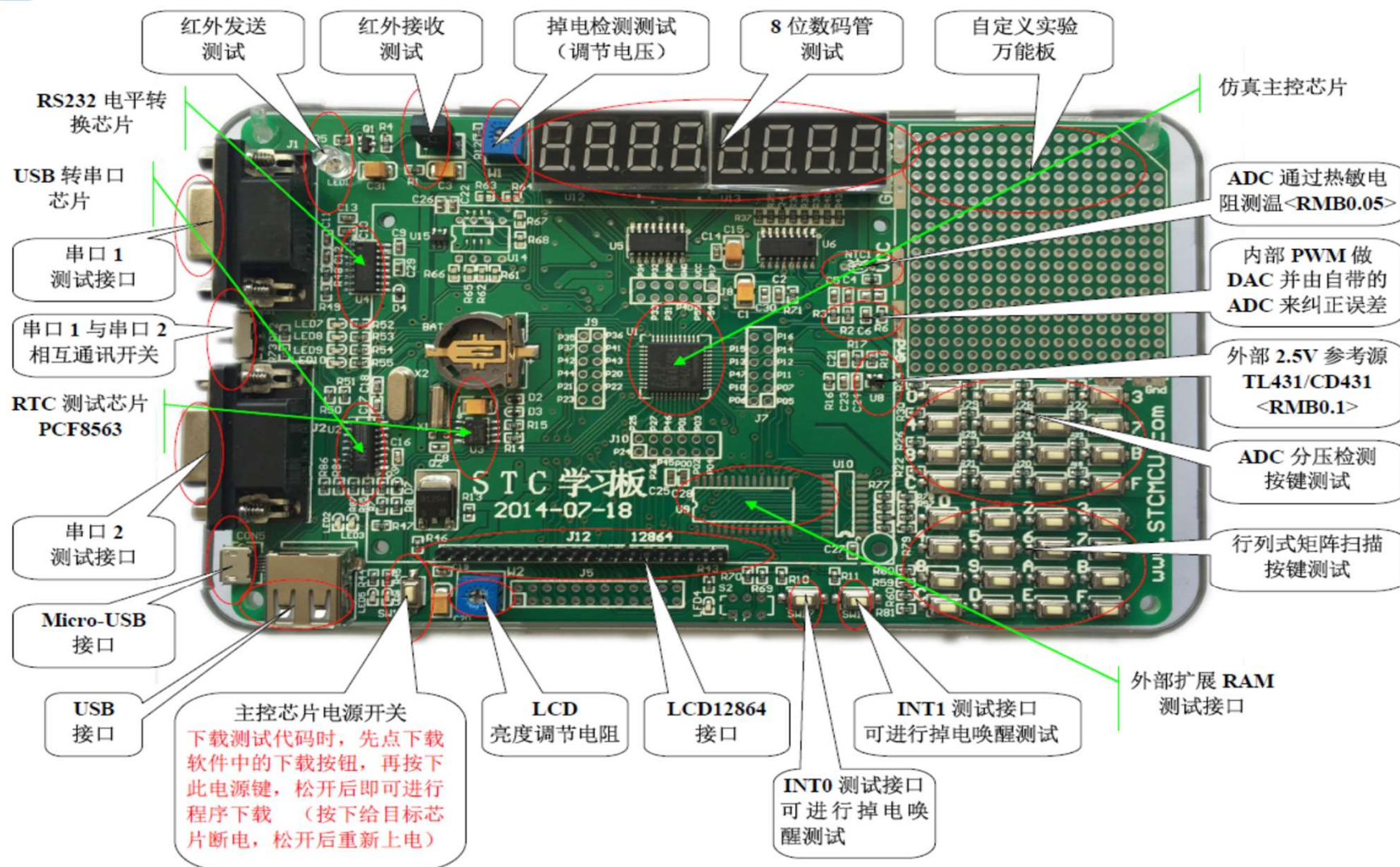
8051单片机硬件开发平台

--stc官方开发板（授课平台）



8051单片机硬件开发平台

--stc官方开发板 (授课平台)



运行第一个8051单片机程序

--硬件连接

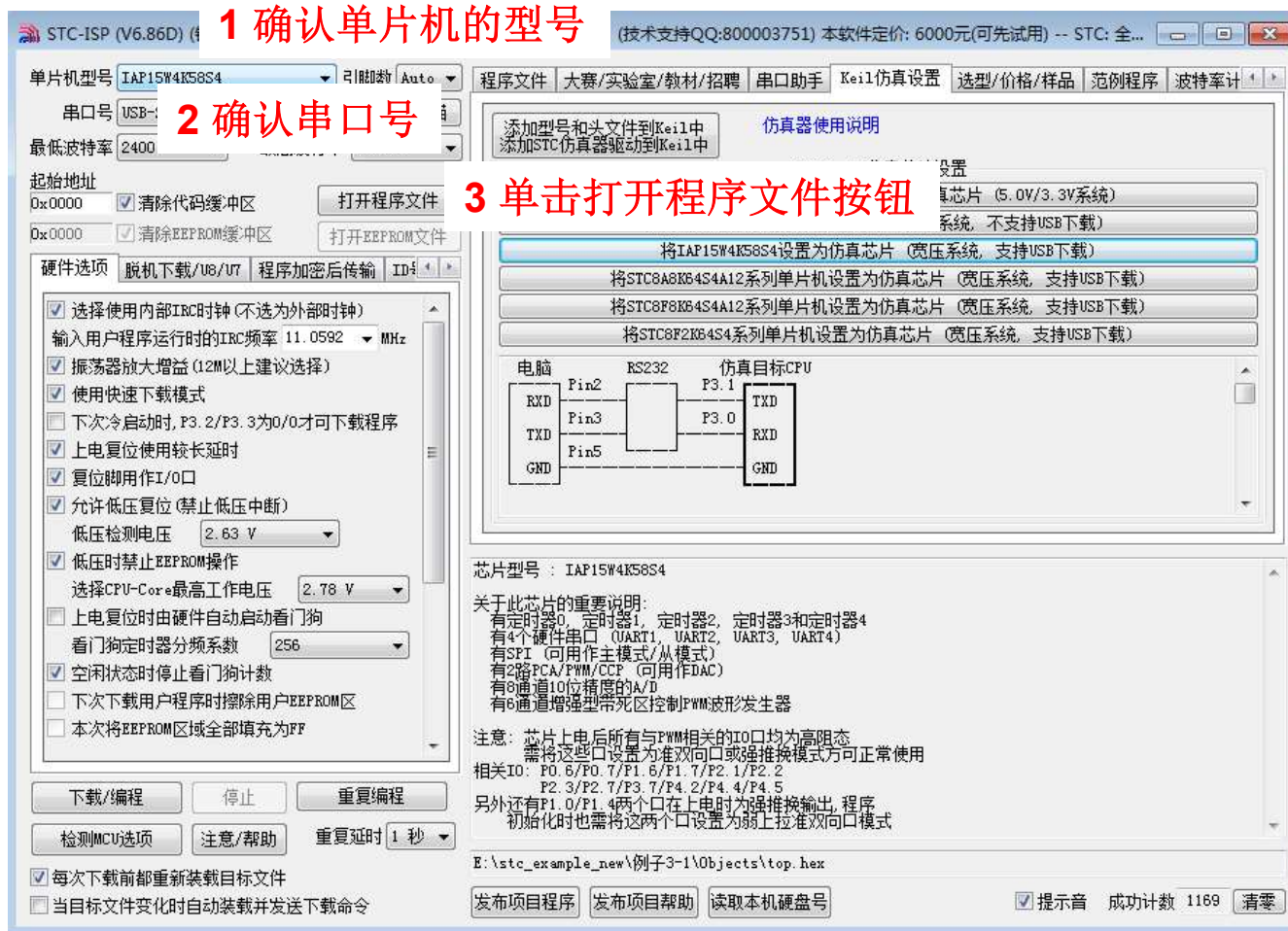


当硬件连接完成后，电脑会自动安装软件驱动程序。

运行第一个8051单片机程序

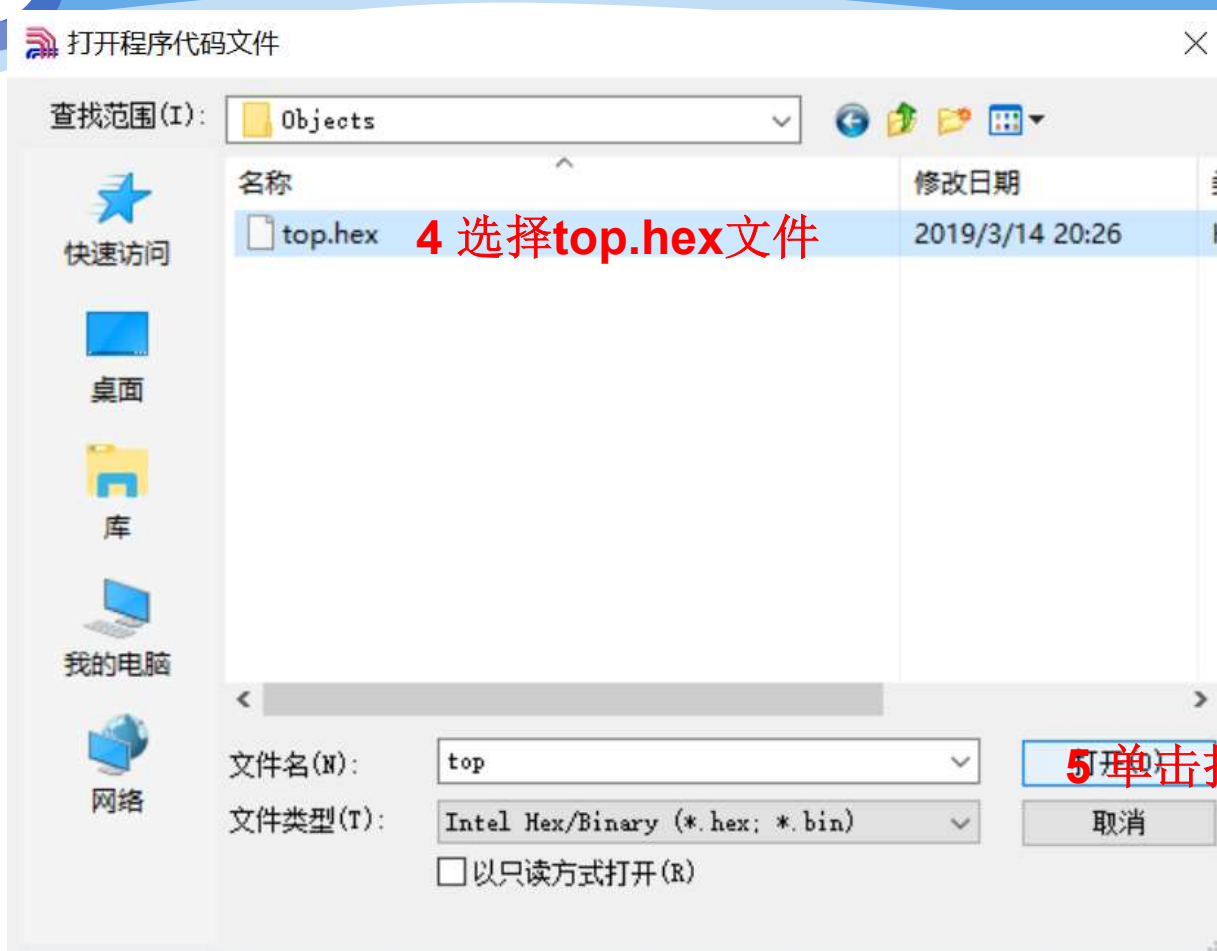
--步骤

打开从STC官网上下载的stc-isp-15xx-v6.86软件



运行第一个8051单片机程序

--步骤



单片机型号 IAP15W4K58S4 引脚数 Auto

串口号 USB-SERIAL CH340 (COM3) 扫描

最低波特率 2400 最高波特率 115200

起始地址 0x0000 清除代码缓冲区 打开程序文件

0x0000 清除EEPROM缓冲区 打开EEPROM文件

硬件选项 脱机下载/U8/U7 程序加密后传输 ID#

☒ 选择使用内部IRC时钟 (不选为外部时钟)

输入用户程序运行时的IRC频率 11.0592 MHz

☒ 振荡器放大增益 (12M以上建议选择)

☒ 使用快速下载模式

☐ 下次冷启动时, P3.2/P3.3为0/0才可下载程序

☒ 上电复位使用较长延时

☒ 复位脚用作I/O口

☒ 允许低压复位 (禁止低压中断)

低压检测电压 2.63 V

☒ 低压时禁止EEPROM操作

选择CPU-Core最高工作电压 2.78 V

☐ 上电复位时由硬件自动启动看门狗

看门狗定时器分频系数 256

☒ 空闲状态时停止看门狗计数

☐ 下次下载用户程序时擦除用户EEPROM区

☐ 本...

8 单击下载/编程按钮

下载/编程

停止

重复编程

检测MCU选项

注意/帮助

重复延时 1 秒

☒ 每次下载前都重新装载目标文件

☐ 当目标文件变化时自动装载并发送下载命令

程序文件 大赛/实验室/教材/招聘 串口助手 Keil仿真设置 选型/价格/样品 范例程序 波特率计

添加型号和头文件到Keil中
添加STC仿真器驱动到Keil中

仿真器使用说明

STC Mon51仿真芯片设置

将IAP15F2K61S2/IAP15L2K61S2设置为仿真芯片 (5.0V/3.3V系统)

将IAP15W4K61S4设置为仿真芯片 (宽压系统, 不支持USB下载)

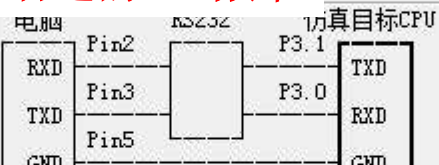
将IAP15W4K58S4设置为仿真芯片 (宽压系统, 支持USB下载)

将STC8A8K64S4A12系列单片机设置为仿真芯片 (宽压系统, 支持USB下载)

将STC8F8K64S4A12系列单片机设置为仿真芯片 (宽压系统, 支持USB下载)

系列单片机设置为仿真芯片 (宽压系统, 支持USB下载)

6 选择合适的IRC频率



7 通过按下单片机开发系统上的电源开关, 给单片机断电

芯片型号: IAP15W4K58S4

关于此芯片的重要说明:

有定时器0, 定时器1, 定时器2, 定时器3和定时器4
有4个硬件串口 (UART1, UART2, UART3, UART4)
有SPI (可用作主模式/从模式)
有2路PCA/PWM/CCP (可用作DAC)
有8通道10位精度的A/D
有6通道增强型带死区控制PWM波形发生器

注意: 芯片上电后所有与PWM相关的IO口均为高阻态

需将这些口设置为准双向口或强推挽模式方可正常使用

相关IO: P0.6/P0.7/P1.6/P1.7/P2.1/P2.2

P2.3/P2.7/P3.7/P4.2/P4.4/P4.5

另外还有P1.0/P1.4两个口在上电时为强推挽输出, 程序初始化时也将这两个口设置为弱上拉准双向口模式

E:\stc_example_new\例子3-1\Objects\top.hex

发布项目程序

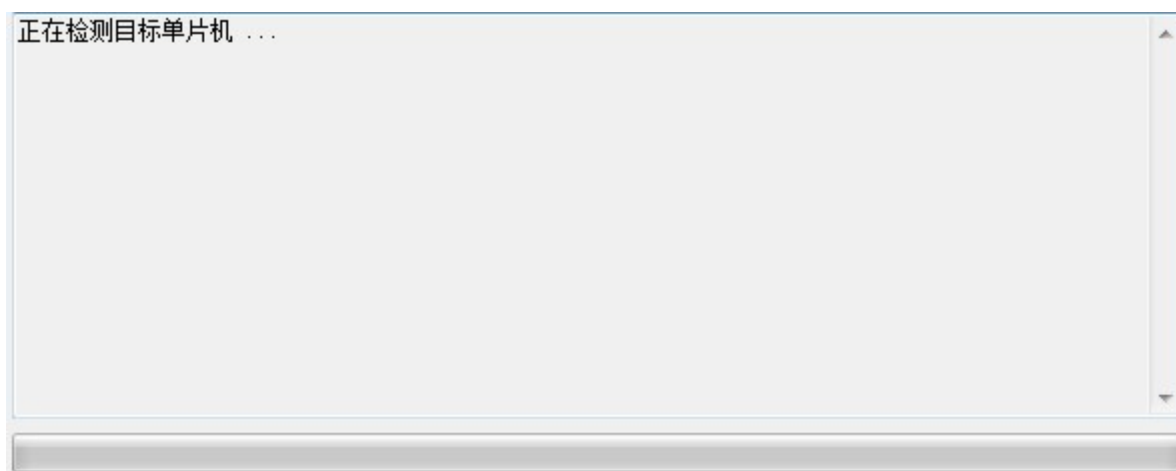
发布项目帮助

读取本机硬盘号

☒ 提示音 成功计数 1169 清零

运行第一个8051单片机程序 --步骤

9 在**STC-ISP**软件右下方窗口中出现“正在检测目标单片机”的提示信息



10. 在通过再次按下单片机开发板上的电源开关，给单片机系统上电，此时**STC-ISP**软件将设计代码自动下载到**IAP15W4K58S4**单片机中

运行第一个8051单片机程序

--步骤

11. 在STC-ISP（V6.85D）软件右下方的界面上出现编程过程中的信息。

```
更新后的硬件选项为：
. 下次冷启动后系统时钟源为内部IRC振荡器
. 当前的振荡器的频率：11.065MHz
. 振荡器放大增益使能
. P3.2和P3.3与下次下载无关
. 上电复位时增加额外的复位延时
. 复位引脚用作普通I/O口
. 检测到低压时复位
. 低压检测门槛电压：2.63 V
. 低压时不能进行EEPROM操作
. CPU-Core最高工作电压：2.78 V
. 上电复位时，硬件不启动内部看门狗
. 上电自动启动内部看门狗时的预分频数为：256
. 空闲状态时看门狗定时器停止计数
. 启动看门狗后，软件可以修改分频数，但不能关闭看门狗
. 下次下载用户程序时，不擦除用户EEPROM区
. 下次下载用户程序时，没有相关的端口控制485
. 下次下载时不需要校验下载口令
. TXD与RXD为相互独立的IO
. 芯片复位后，TXD脚为弱上拉双向口
. 芯片复位后，P2.0输出高电平

. 芯片出厂序列号：F569008C033354
单片机型号：IAP15W4K58S4
固件版本号：7.3.4T

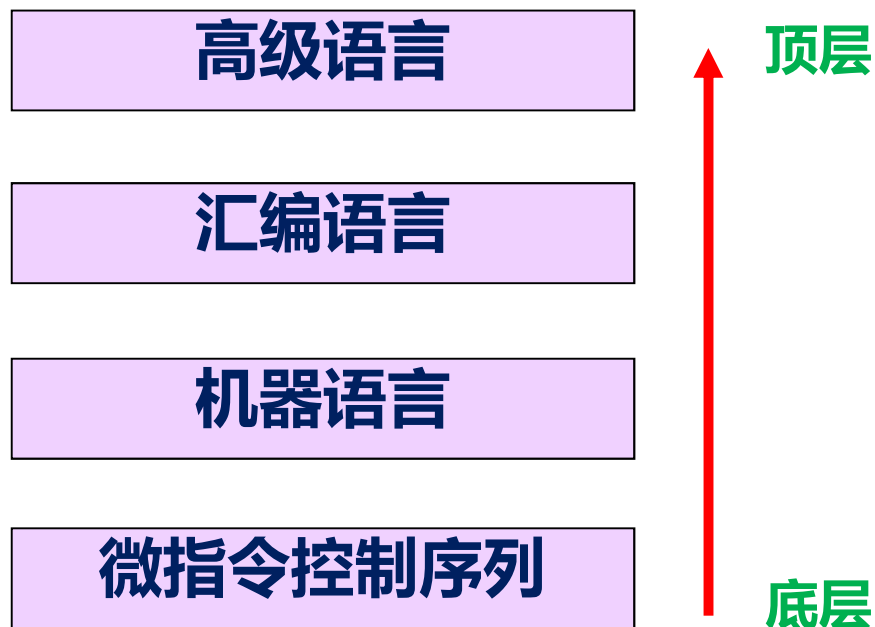
. 用户设定频率：11.059MHz
. 调节后的频率：11.065MHz
. 频率调节误差：0.054%
```

操作成功！(2018-04-06 18:12:08)

12. 观察STC开发板上外设的工作情况。

8051单片机编程语言

从系统结构来说，单片机语言分为四个层次



8051单片机编程语言

--微指令控制序列

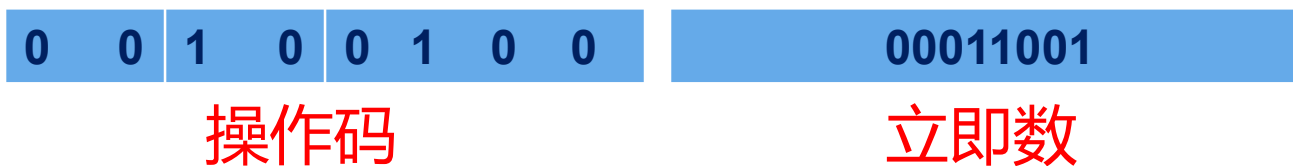
通过由有限自动状态机所构成的微指令控制器对CPU内部的寄存器、存储器和ALU等参与具体数据处理的功能部件“发号施令”。

8051单片机编程语言

--机器语言

输入给CPU，通知其执行运算的称为机器语言。

- 机器语言是由0和1构成的序列，序列中包含操作码和操作数两部分。
- ◆ 8051单片机中，ACC累加器和一个常数（立即数）相加的机器语言的格式如下：



8051单片机编程语言

--机器语言

■ 操作码

- 操作码告诉CPU所需要执行的操作。操作码部分包含了操作的类型编码，同时也包含了一部分的操作数内容，指明了参与加法运算的一个数来自ACC累加寄存器中。

■ 操作数

- 操作数是执行操作所针对的对象。这些对象包括：立即数（常数），寄存器和存储器等。

纯粹意义上的机器语言对程序员太难理解了，为什么？

8051单片机编程语言

--汇编语言

用汇编助记符描述机器指令完整的形式为：

[标号:] 助记符 [操作数] [; 注释]

- 标号用来表示一行指令，
- 助记符表示所要执行的逻辑操作行为，
- 操作数为逻辑操作行为所操作具体对象
- 现在用汇编语言来描述下面的机器指令：

ADD A, #25

8051单片机编程语言

--汇编语言

- 汇编语言下面是机器语言

- ◆ 对于使用汇编语言编程的程序员来说，他必须很清楚CPU的指令集，寄存器单元和存储器映射等繁琐的硬件规则。

- 虽然其执行效率基本上和机器语言一样，但是使用汇编语言编程效率很低。

- ◆ 因为很多程序员根本不了解CPU的具体内部结构，对它们来说，使用汇编语言编程并不比使用机器指令编程好到哪里去，这也是一件令他们非常痛苦的事情。

8051单片机编程语言

--汇编语言

汇编语言仍然非常重要。重要性体现在：

- 对理解CPU内部的结构和运行的原理非常重要；
- 很多与CPU打交道的软件驱动程序，尤其是程序的引导代码必须用汇编语言开发。 **(C语言不能直接映射到底层硬件)**
- 此外，在一些对程序执行时间比较苛刻的场合也需要使用汇编语言进行开发。

8051单片机编程语言

--高级语言

MCU的软件集成开发工具（如Keil μ Vision）支持使用C语言实现对单片机的程序开发。

- **C语言不能直接在CPU上，它必须通过编译器和链接器（软件工具）的处理，最终生成可执行文件，也就是转换成机器语言，才能在CPU上运行。**

8051单片机编程语言

--高级语言

使用以C语言为代表的高级语言所编写的代码其运行的效率不可能比用汇编语言编程的运行效率高。

- 要让C语言所编写的代码和汇编语言编写的代码有一样高的代码执行效率，需要C程序员使用各种程序设计技巧提高C语言代码的设计效率
- 调整编译工具的优先级设置选项，以满足代码长度和运行时间的双重要求。

高级语言

--代码长度和运行时间

- 尽可能的要求程序代码尽可能的短，这样可以大大节省所占用的程序存储器的空间，减少了对程序存储器空间的要求。
- 程序代码的运行尽量满足实时性的要求，这样在程序的执行过程中可以实时响应不同外设的要求。

高级语言

--代码长度和运行时间

越来越多的厂商提供了硬件的应用程序接口
(**Application Program Interface, API**) 函数。

- 程序员可以不用知道更多的硬件实现细节，只需关心如何编写代码来使硬件工作，显著提高了程序设计效率。

C语言、汇编语言和机器语言之间的对应关系

两个8位数相加的C语言描述的例子

```
void main()
```

```
{
```

```
    char a=10,b=80,c;
```

```
    c=a+b;
```

```
}
```

2:	{	
3:	char a=10,b=80,c;	
4:	c=a+b;	
C:0x000F	740A	MOV A,#0x0A
C:0x0011	2450	ADD A,#0x50
C:0x0013	F508	MOV 0x08,A
5:	1	

机器语言 反汇编代码 (汇编语言)

小结

对于8051单片机而言，应该从电子系统的高层次来认知，而不应该仅仅只从单片机本身来认知。

- **这是因为由8051单片机所构成的电子系统，包括了软件和硬件两大部分。**
 - **软件包括了汇编语言设计、C语言程序设计、操作系统、数据结构、算法（数字信号处理和控制理论）的知识；**
 - **硬件包含了模拟电子技术、数字逻辑、处理器、接口、ADC和DAC、电路设计的知识。**

小结

教材就是基于8051单片机系统这个层次，虽然单片机所涉及的知识点较多，但是仍然有一条主线。

- **即正确认识处理器架构、处理器和指令集之间的关系、汇编语言和C语言之间的关系、接口与外设之间的关系。**



小结

在学习单片机这门课程的时候应该紧密围绕着三大主题。

- 软件和硬件的协同设计
- 软件和硬件的协同仿真
- 软件和硬件的协同调试