Learning By Doing
简化版单片机目标板使用说明书

第 1 章 系统简介

LBD3 简化版单片机目标板是由南京邮电大学计算机学院自主设计的配合单片机教学的实验系统。使用该系统共可十分方便地配合上位机集成开发环境进行本课程的课内实验，是提高单片机课程教学质量及学生动手能力的得力教学实验设备。

1.1 电路板外形

该系统电路板外形如图 1.1 所示。

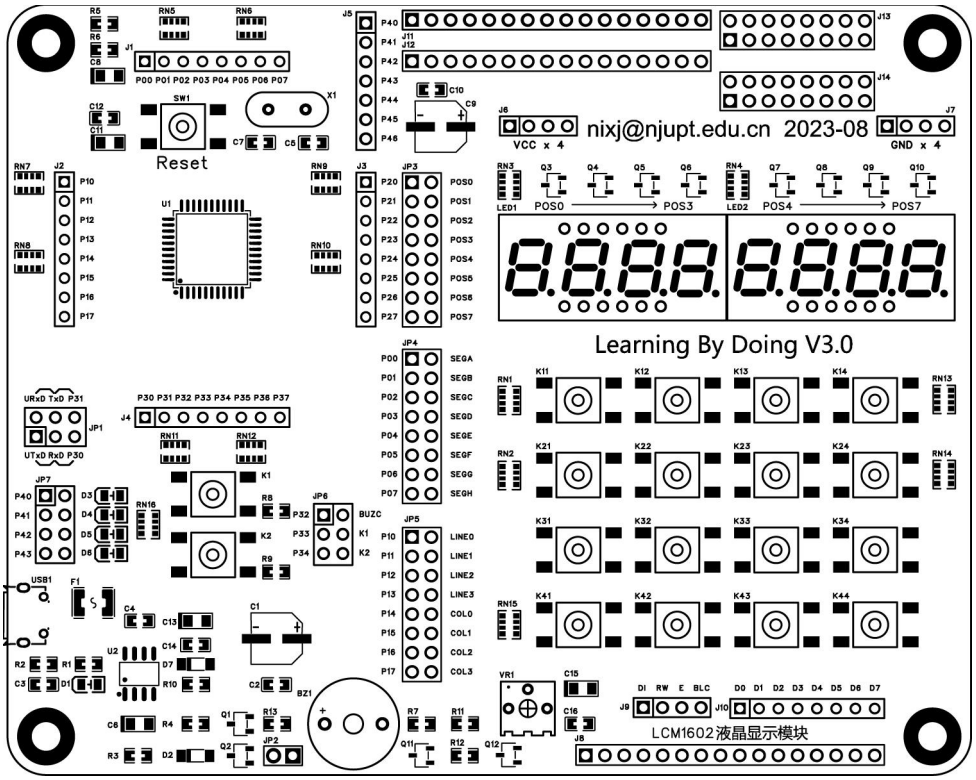


图 1.1 LBD3 简化版单片机目标板电路板外形图

最终加工完成的该系统电路板外形如图 1.2 所示。

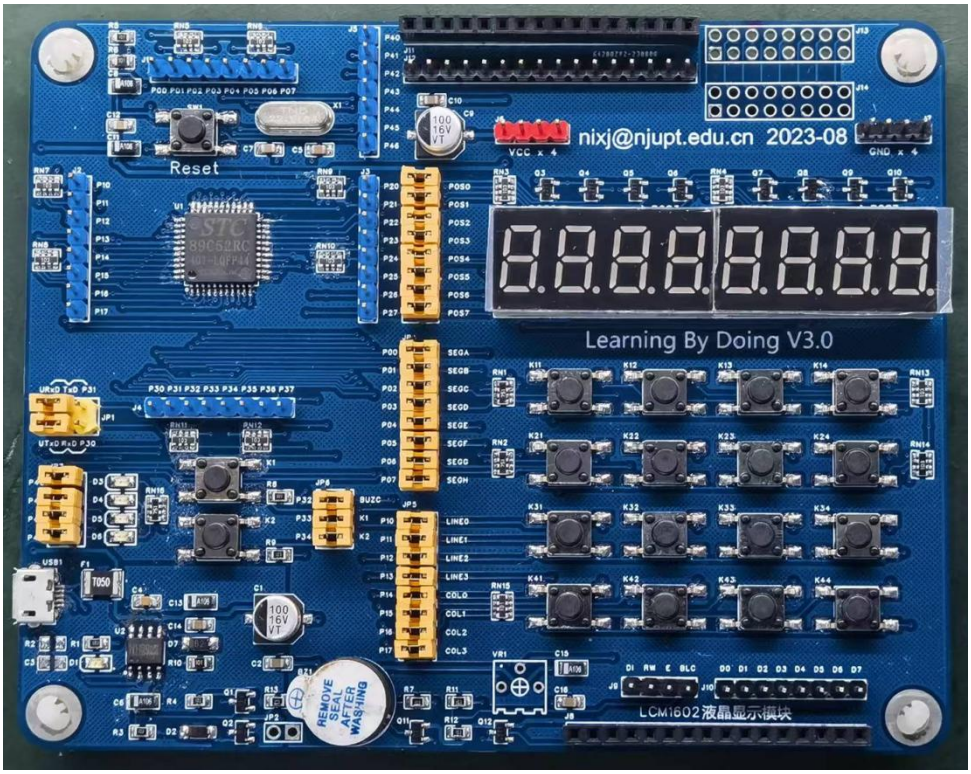


图 1.2 LBD3 简化版单片机目标板实物图

1.2 系统功能简介

1.2.1 电源和 USB 转串口电路

LBD3 简化版单片机目标板（以下简称为 LBD 系统）使用上位计算机的 USB 端口供电。LBD 系统使用非常简单，只要使用 Micro USB 线将上位机的 USB 口和 LBD 系统的 USB 插座（电路板左下角 USB1）即可，此时 USB1 右下方的发光二极管 D1 点亮，表示 USB 系统供电正常，如果是第一次连接 LBD 系统，Windows 操作系统会提示找到 USB 转串口设备，安装驱动程序（CH340 芯片）后，Windows 系统将在“端口(COM 和 LPT)”中显示新安装的串行口的端口号（如 COM3）；若以前已经安装过驱动程序，则插上 USB 电缆后将自动识别并分配好 COM 端口。

在 USB 线连接到上位机的情况下，LBD 系统的 USB 转串口电路和单片机 CPU 及外围设备都将一直保持供电。当使用 STC 单片机上位机编程软件给单片机下载/更新代码时，LBD 系统会自动短暂断开单片机系统的电源并恢复，待代码下载/更新完成后，单片机将自动执行新代码，极大地方便了程序的调试和更新。

为了保护计算机的 USB 端口、单片机及其外围电路，LBD 系统设计了一个自恢复保险丝 F1，若电路板上出现短路的情况，F1 将暂时熔断；排除短路问题后，F1 将自动恢复到导通状态。

为了方便用户引出电源进行实验，LBD 系统从电路板上引出了+5V 电源和 GND，引出接口为电路板右上侧的 J6 和 J7。本系统中，+5V 电源可承受的总电流不大于 500mA。

1.2.2 CPU

1. CPU 及晶振频率选择

LBD 系统中，选用宏晶公司的国产单片机，型号为 STC89C52RC，晶振频率为 22.1184MHz。

2. CPU 工作模式

为简化电路，方便实验，LBD 系统中单片机仅设计了最小模式，即没有扩展总线，P0~P3 均作为普通 IO 端口使用，各端口的每一位引脚都可以作为独立的端口引脚使用。

STC 89C52RC 兼容标准 51 内核，自带 8K 字节的 Flash 作为程序存储器，支持通过串行口进行在系统编程（ISP）。当使用 STC 单片机上位机编程软件给单片机下载/更新代码时，LBD 系统会自动短暂断开单片机系统的电源并恢复，待代码下载/更新完成后，单片机将自动执行新代码，极大地方便了程序的调试和更新。

1.2.3 系统复位

LBD 系统采用了阻容上电复位及按键复位两种复位方式。用户使用 USB 线将目标板连接到上位机的 USB 口时，系统自动进行上电复位；此后任何时候用户按下位于电路板左上角的按钮 SW1，系统进行将进行一次复位操作。

第 2 章 系统资源

LBD3 简化版单片机目标板集成了基本的实验资源，可以直接进行单片机原理及系统设计课程的课内实验。

在下文对实验系统的描述中，某端口或连接器的第 1 位（或低位）对应字节的低有效位（LSB），第 8 位（或高位）对应字节的高有效位（MSB）。

2.1 CPU 最小系统

LBD 系统的 CPU 位于图 1-1 中的左上区域，芯片为 TQFP44 封装，直接焊接在电路板上，不需要额外接线，提高了实验过程中电路连接的可靠性。

2.1.1 CPU 及最小系统电路

系统中 CPU 的标号为 U1，型号为 STC 89C52RC，位于图 1-1 中左上部分。基本的最小系统电路如图 2.1 所示。CPU 和标准 MCS-52 内核兼容，具体功能请参阅其 Datasheet。

在图 2.1 中，J6 用于向可能的外接电路输出 VCC，J7 用于外接 GND；C9~C12 为电源续流及去耦电容；SW1 为复位按钮，按下 SW1 即可实现 CPU 的一次复位；K1 和 K2 为两个独立按键，将 JP2 的 3-4 及 5-6 引脚短接，即可通过 P3.3 和 P3.4 输入独立按键 K1 和 K2 的状态。

在平时的使用中，还需要注意的是 P3.0 和 P3.1，即单片机的串行口 RxD 和 TxD 的连接方式。为了实现单片机代码的下载，单片机的串行口需要和 USB 转串口芯片 CH340N 扩展的串行口相连，因此平时 JP1 的 1-3 和 2-4 需要用短路块短接，保证单片机串行口的收、发分别和上位机通过 USB 总线扩展的串行口的发、收相连，以实现代码的下载。如果用户程序需要将单片机的 P3.0 和 P3.1 作为普通 IO 口使用，则需要在代码下载后，断开 USB 连接（相当于关闭目标板的电源），将短路块移动到另外一侧，即短路 3-5 和 4-6，再连接 USB 线即可将 P3.0 和 P3.1 作为普通 IO 口使用。

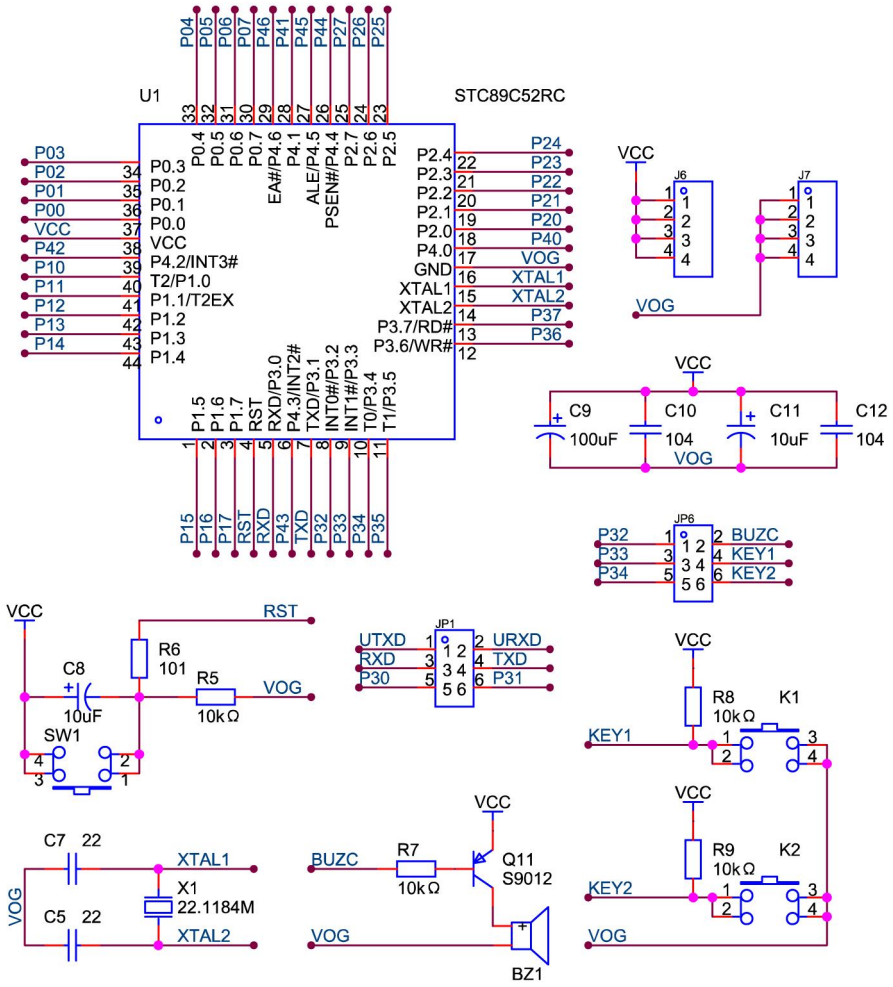


图 2.1 STC89C52RC 最小系统原理图

2.1.2 CPU 各引脚的外围电路

我们知道，MCS-51 系列单片机的并行端口引脚作为普通 IO 口使用时，P0 口必须外接上拉电阻，其它三个端口也建议外接上拉电阻以加强驱动能力。因此，LBD 系统专门为 STC89C52RC 的所有 IO 端口引脚外接了上拉电阻。相关电路原理图如图 2.2 所示，P0~P3 四个 IO 端口的 32 个引脚分别通过 8 个 4 联电阻 RN5~RN12 上拉到 VCC，每个上拉电阻的阻值均为 10KΩ，以提高 IO 端口的驱动能力。同时，为了方便引出各 IO 端口引脚进行实验，系统通过 J1~J4 将 P0~P3 端口的每个引脚都进行了并接，必要时可通过杜邦线将相应的管脚引出进行实验。

STC89C52RC 还提供了一个扩展的 P4 端口，但只有 P4.0~P4.6 共 7 个引脚，通过 J5 并接引出，且没有外扩的上拉电阻，在使用时需加以注意。

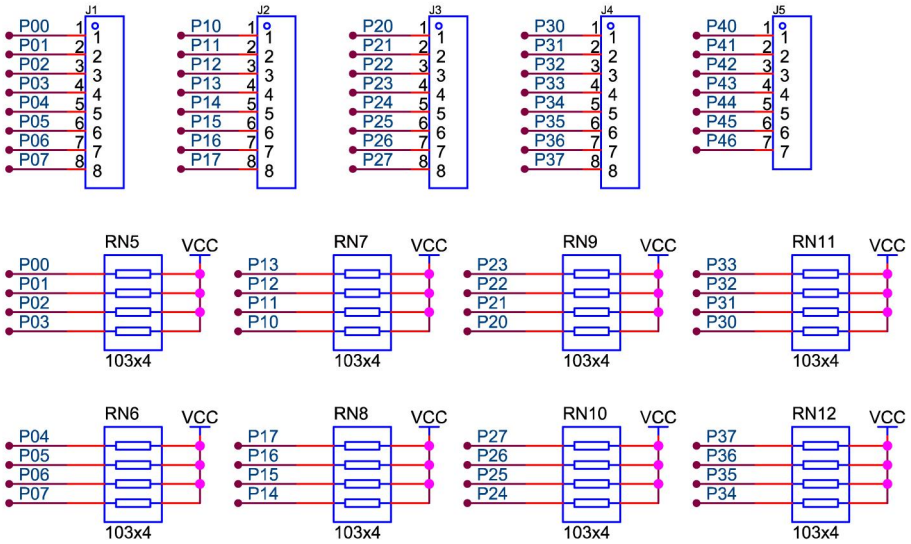


图 2.2 CPU 各端口引脚的上拉电路原理图

2.2 目标板其它器件

2.2.1 蜂鸣器

蜂鸣器控制电路见图 2-3，标号为 BZ1。用跳线将 JP2 的 1-2 脚短接，即可通过 P3.2 引脚控制，P3.2 输出低电平，Q11 导通，蜂鸣器鸣响；P3.2 输出高电平，Q11 截止，蜂鸣器停止鸣响。在电路板上，蜂鸣器位于下方中部。

2.2.2 字符点阵 LCD

系统中提供了一个外接 1602 字符点阵 LCD 模块的接口 J8，位于电路板的右下方，具体的电路连接如图 2-3 所示。

1602 LCD 模块接口电路包括数据线（D0~D7）、控制线（D/$\overline{I}$、R/$\overline{W}$、E）和背光控制（LBLC），分别通过 J10、J9 引出。LBLC 用于控制 LCD 模块的背光开关，当该引脚输入低电平信号时，LCD 模块的背光打开，否则背光关闭。可变电阻 VR1 用于调节 LCD 显示模块的显示对比度。

在本实验系统中，可通过将单片机的 IO 端口通过杜邦线连接到 J9、J10 的方式实现对 1602 字符点阵 LCD 模块的控制。通过程序控制单片机的端口，模拟 LCD 模块的总线操作时序，即可控制 LCD 模块的显示，这种工作模式下 CPU 和显示模块之间的控制命令和操作时序等请参阅 1602 LCD 显示模块的数据手册。

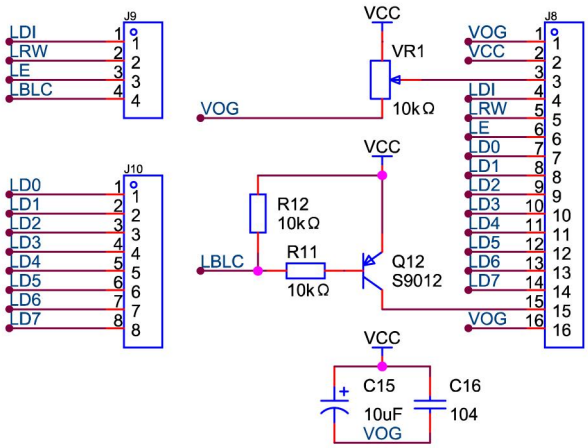


图 2-3 1602 LCD 显示模块电路图

2.2.3 独立 LED

实验系统中提供了 4 个独立的 LED，编号为 D3~D6，位于电路板左侧 USB 插座上方，上下排列。D3~D6 的负极连接到 JP7 的一边，它们的正极分别经过一个 1K 欧姆的电阻连接到 VCC。单片机的 P4.0~P4.3 引脚连接到 JP7 的另一边，当通过 4 个短路块将 JP7 两边的引脚分别短接时，P4.0~P4.3 某位 IO 端口引脚输出低电平，则对应位置的 LED 点亮。

独立 LED 的电路如图 2-4 所示。

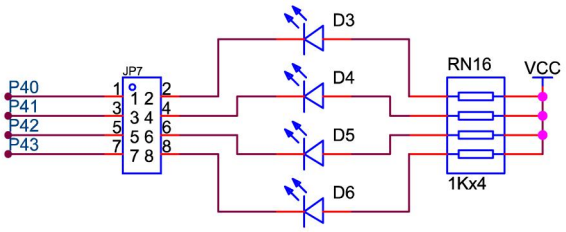


图 2-4 独立 LED 驱动电路

2.2.4 独立按键

本系统提供了 2 个独立按键 K1 和 K2，位于独立 LED 的右侧。当按键没有按下时，输出高电平；当按键按下时，输出低电平。2 个按键分别通过 JP6 的短路块连接到 CPU 的 P3.2 和 P3.3 引脚。

独立按键和拨码开关电路如所示。

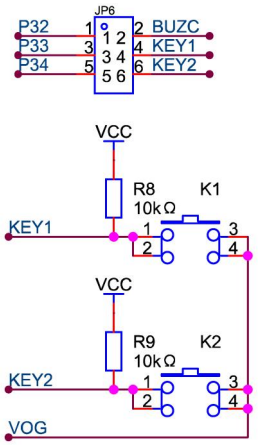


图 2-5 独立按键电路原理图

2.2.5 行列式键盘

行列式键盘位于电路板右下方，共 16 个按键，分为四行四列。键盘的四行连接到 JP5 的低 4 位，键盘的四列则连接到 JP5 的高 4 位。在每个行列的交叉点，分别跨接了一个按键的两端。用户可通过循环对每一行输出低电平，同时读入四列的状态来判断是否有按键按下及其行列值。行列式键盘的电路如图 2-4 所示。当 JP5 的两侧分别通过 8 个短路块短接时，4 根行线和 4 根列线分别连接到单片机 P1 端口的低 4 位和高 4 位。

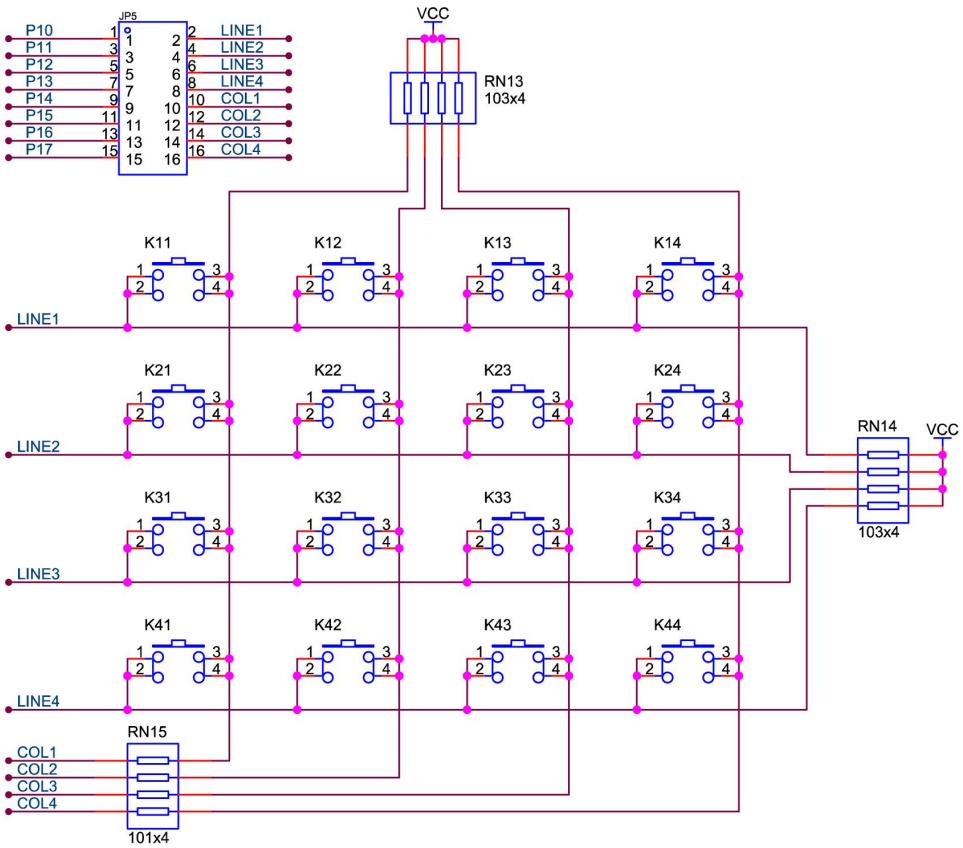


图 2-6 4x4 行列式键盘电路原理图

2.2.6 组合数码管

系统中的组合数码管共有 8 个，由两个 4 位共阳型组合数码管组成，位于电路板右侧中部。每个 4 位数码管的 4 个公共正极端分别引出，定义为 POS0~POS7，为 8 个数码管的位置选择端，由 JP3 引入。当 JP3 的两边通过 8 个短路块短接时，使用单片机的 P2 端口驱动 8 个 PNP 型三极管，作为对每个显示位置数码管的位置选择信号。

要点亮数码管，仅有位置驱动还不够，必须同时提供该数码管的段驱动。在本系统中，8 个数码管的同名段都复接在一起，经过限流电阻后，连接到 JP4。当 JP4 两端短接时，由单片机的 P0 端口驱动。当 P0 端口的某位引脚输出低电平时，则该位对应的数码管的段就被驱动为有效，如果此时某个位置驱动也有效，则该位置的数码管的某段点亮。组合数码管部分的电路如图 2-7 所示。

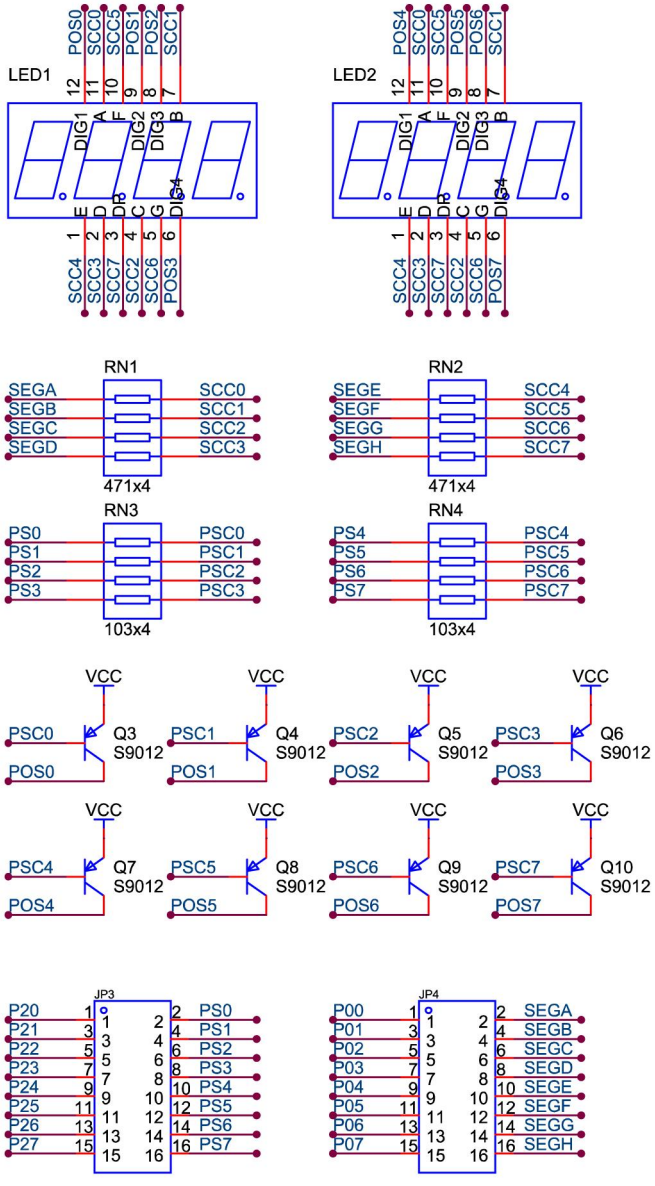


图 2-7 组合数码管显示驱动电路原理图

第 3 章 Keil μVision C51 集成开发环境及 ISP 软件使用简介

汇编语言和 C 语言是 MCS-51 系列单片机系统开发中最常用的程序设计语言。

汇编语言常被称为“低级语言”，这是由于汇编语言的语句通常和 CPU 的指令助记符直接对应的原因。采用汇编语言设计的程序具有程序结构紧凑、目标代码效率高、占用程序存储器空间少和运行速度快等特点，适用于实时测控等应用领域。但由于汇编语言面向具体 CPU 的指令系统，其程序代码的可移植性、通用性较差，编程也比较烦琐，提高了系统设计和维护的难度。

C 语言则基本摆脱了对具体 CPU 指令系统的依赖。C 语言是面向过程的，侧重于如何解决问题，具有统一的语法规范，所以程序代码的可移植性、通用性较好，维护和编程都较为简单。

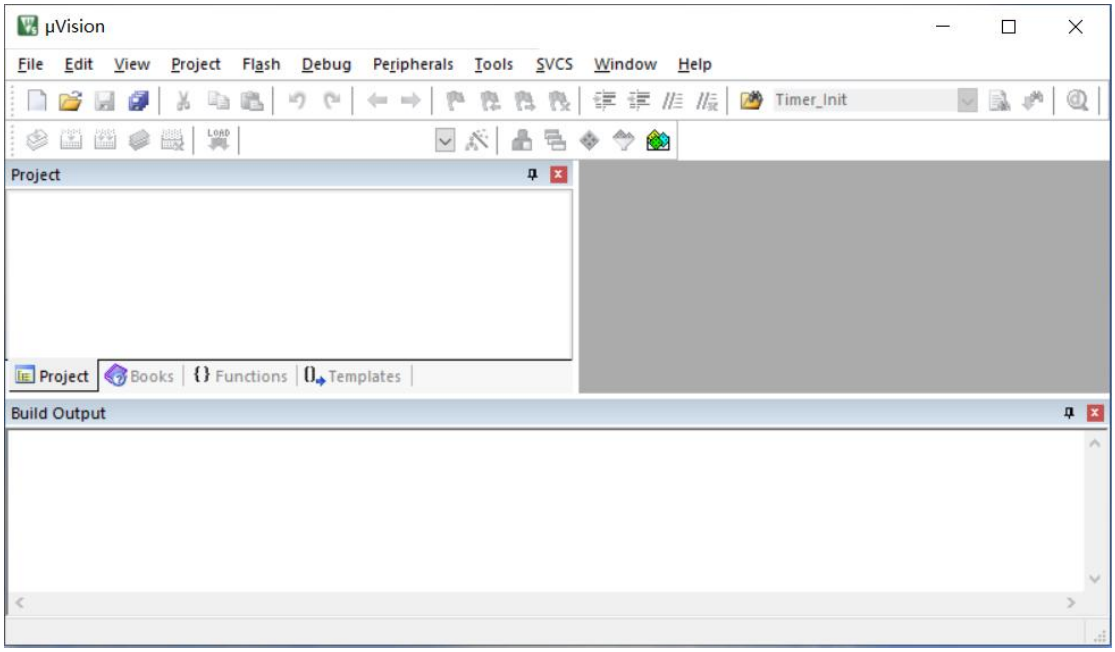
针对 MCS-51 型单片机系统的 C 语言称为 C51。目前 C51 的应用非常广泛，由德国 Keil 公司推出的 Keil μVision 集成开发环境已成为 C51 开发工具领域事实上的标准，由于其出色的编译性能，Keil 现已被 ARM 公司收购。本章对其进行简单的介绍。

3.1 Keil 软件的使用

3.1.1 启动 Keil

安装好 Keil 后，双击桌面图标启动 Keil，将看到如下图所示的启动界面。注意安装的 Keil 版本不同，启动时的界面也会有所不同，但开发单片机程序具体的操作过程大致相同。

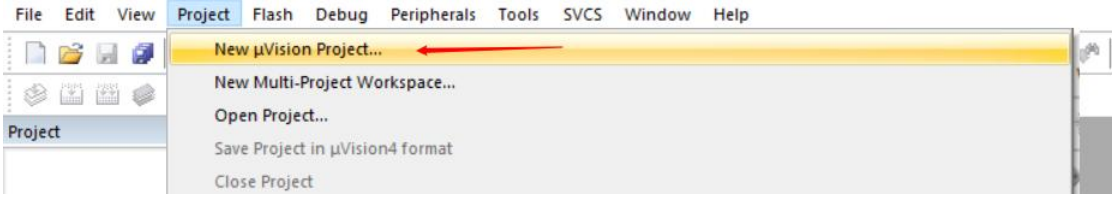
启动完毕的 Keil 集成开发环境的界面如下图所示。



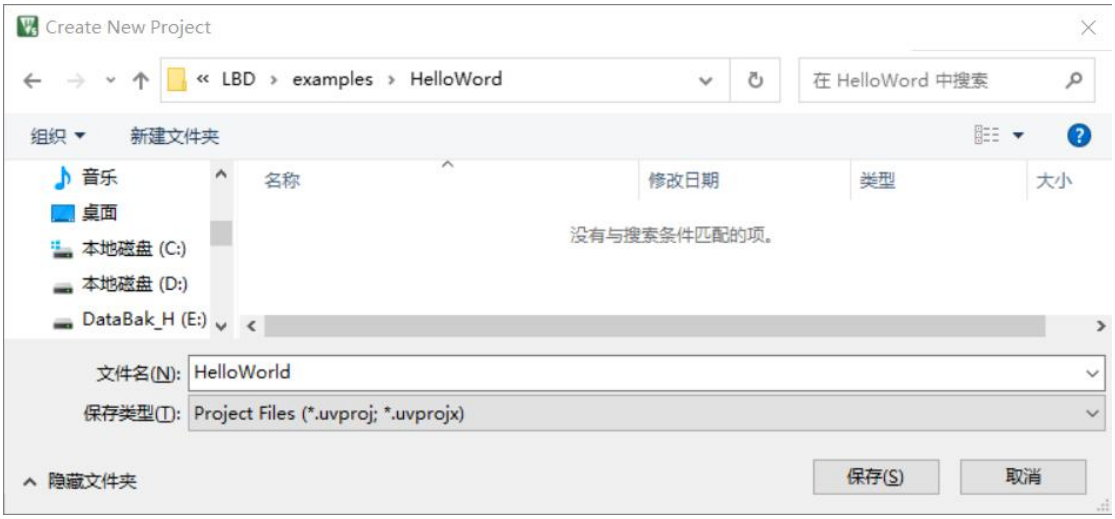
3.1.2 新建工程

在用 Keil 进行单片机程序设计之前，首先要建立工程。所谓的单片机程序，不但包括 C 语言或汇编语言的源程序，还包括和整个应用系统相关的一些设置和参数，例如该应用系统中单片机的晶振频率、外部程序存储器及数据存储器的地址空间等，建立工程的目的就是将所有的这些设置和参数一并保存，以方便系统的维护和管理。在 Keil 中建立工程的步骤如下：

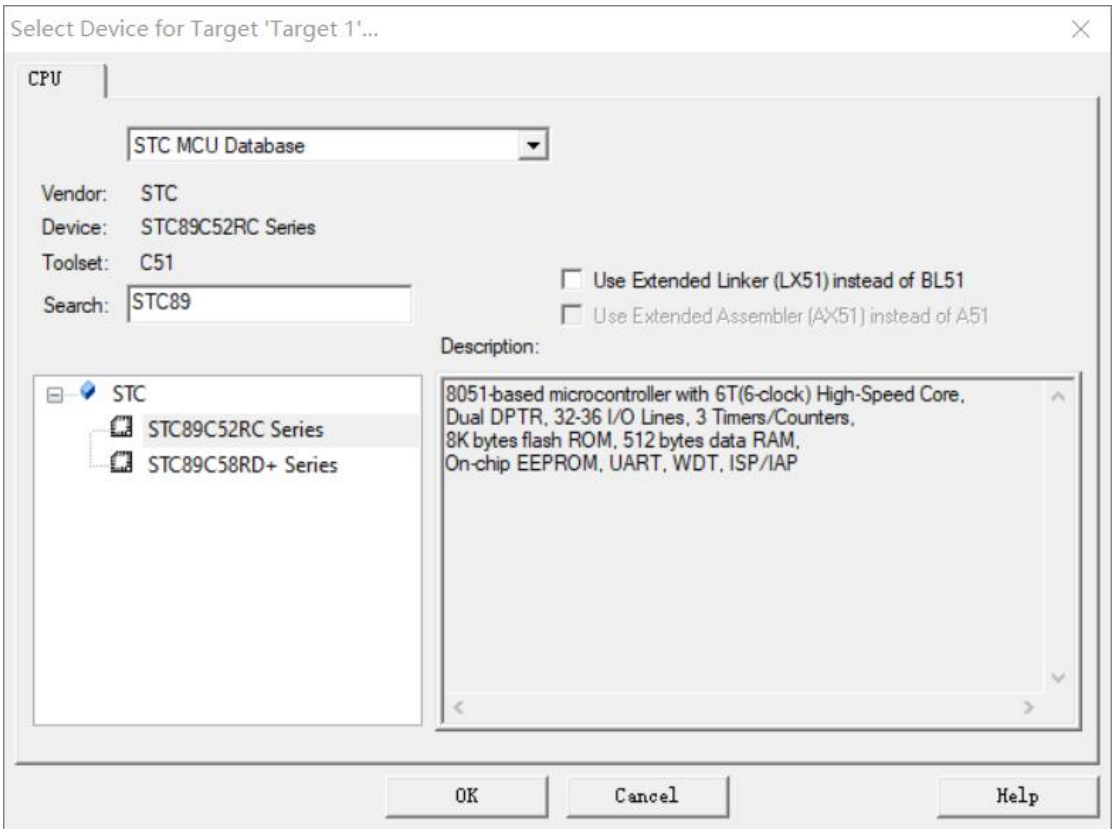
(1) 选择菜单 Project→ New Project



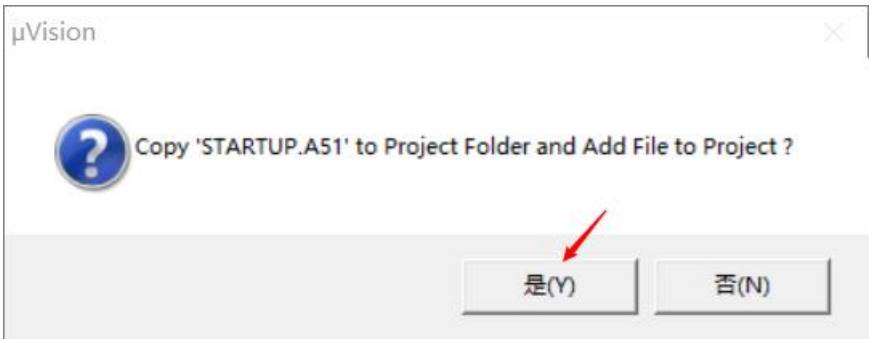
(2) 选择新工程的存储路径，并输入工程的文件名



(3) 根据 Keil 的提示选择该工程使用的 CPU 类型

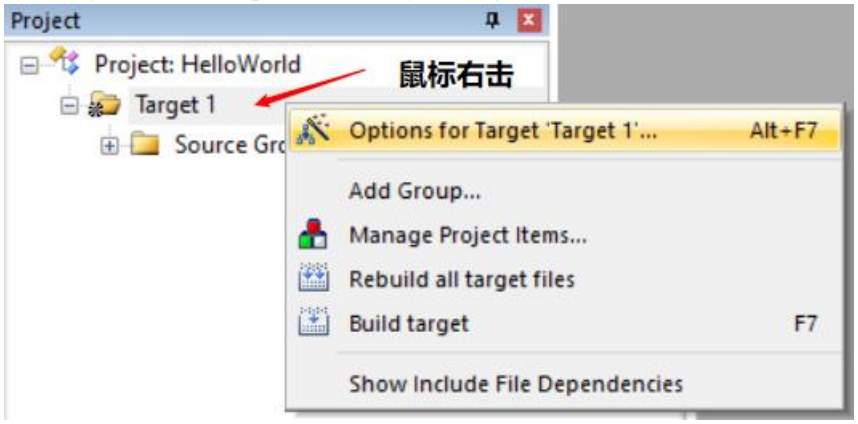


(4) 选择 Yes，将 Keil 提供的启动代码加入新建的工程



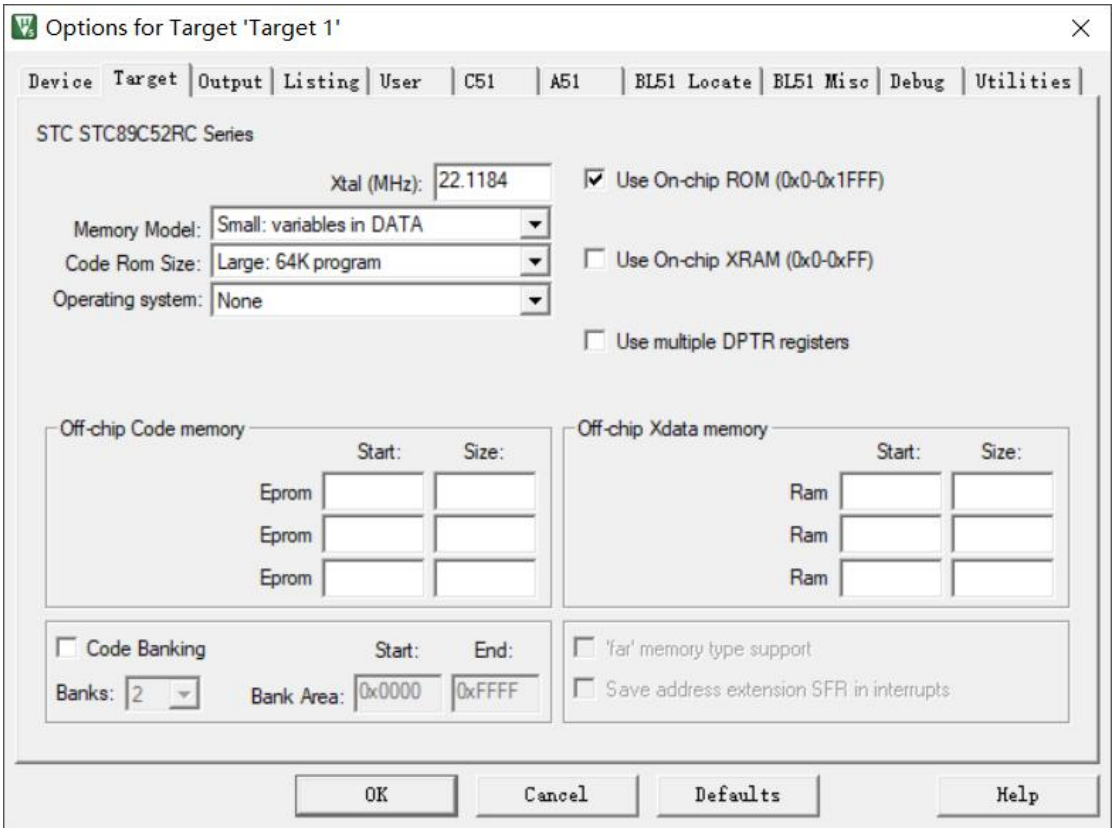
3.1.3 设置工程的配置及参数

(1) 右击 Target 1，选择 Options for Target ‘Target 1’进行参数设置

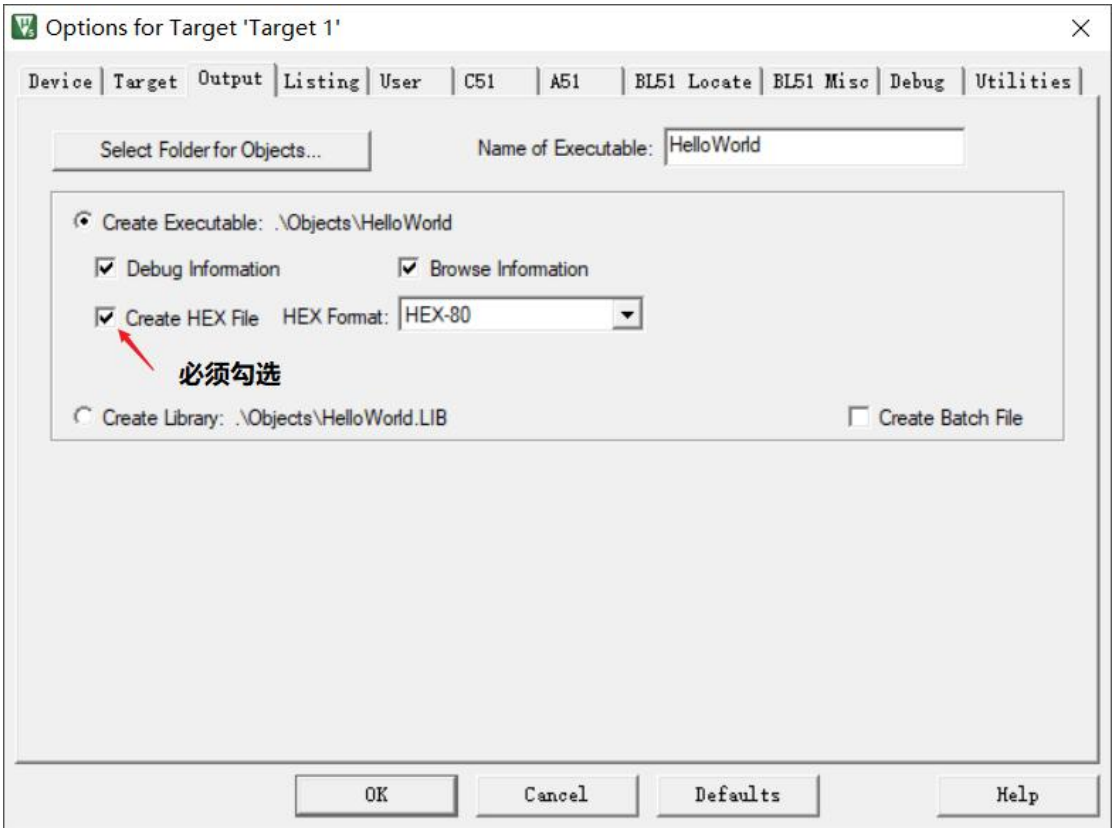


(2) 设置晶振频率，选择使用内部 ROM 存储程序。如果有外部 ROM 或 RAM，应在

Off-Chip Code memory 和 Off-chip Xdata memory 栏中设置其起始地址及存储容量。



(3) 选择 Output 选项卡，选中 Create HEX File 项，要求编译后生成 HEX 文件

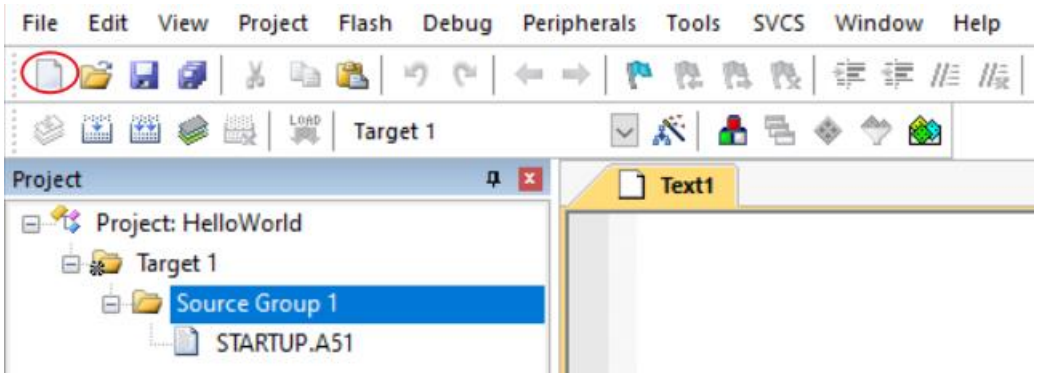


至此 Project 的参数配置完毕。

3.2 Keil 单片机程序开发实例

3.2.1 使用 Keil 建立一个实际的工程

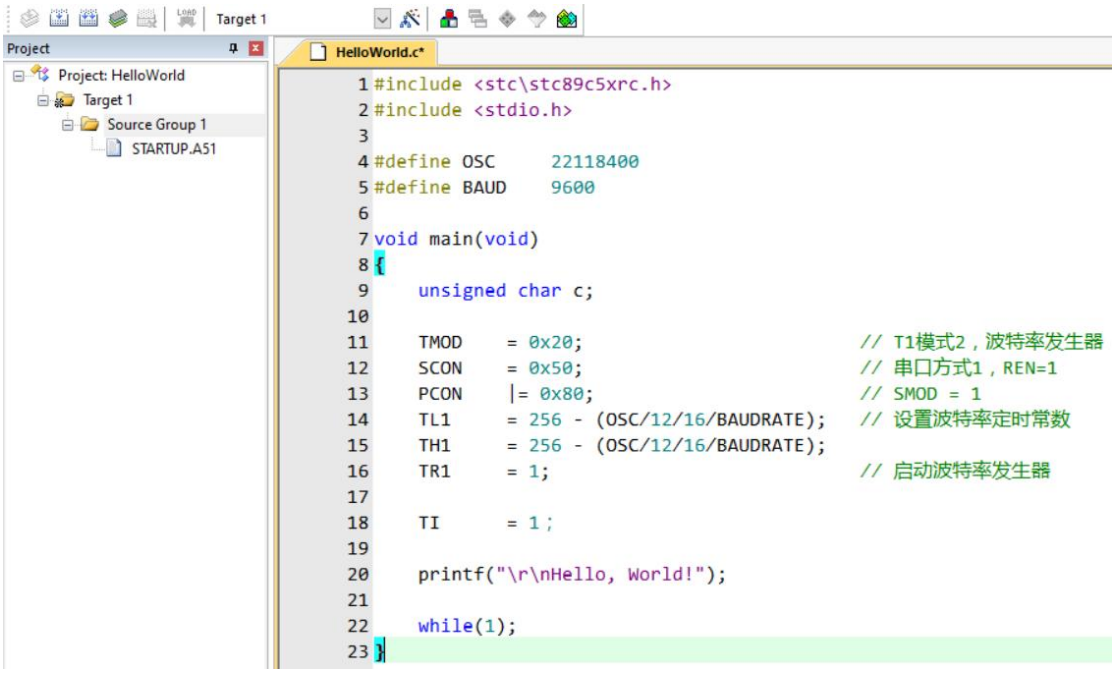
(1) 在 Keil 的菜单中，选择 File→New 或点击 NEW 图标，可以建立一个新文件



选择建立新文件后，Keil 会打开一个临时窗口，名称为 Text1，供开发人员编辑程序。由于 Keil 的编辑器针对汇编语言和 C 语言源程序提供了关键字识别、自动缩进、语法检查等专业化的处理功能，因此建议先将这个 Text1 文件另存为合适的汇编或 C 源文件后再输入程序。

选择 File→Save AS 菜单，将文件存储为有合适扩展名的源程序文件，然后再输入程序。例如，我们设计的第一个例子是输出 Hello world 信息，故将这个 Text1 另存为 HelloWrold.C，然后输入程序代码。

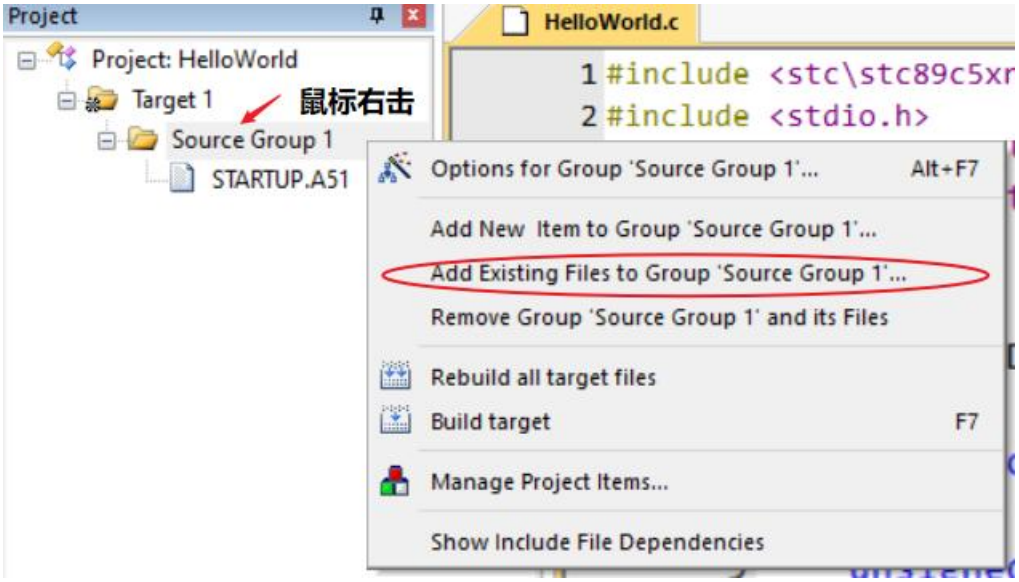
(2) 输入程序代码后的 C 语言源程序



输入程序的过程我们将发现，Keil 提供的编辑器会将 C 语言的关键字按不同颜色显示，还能根据代码的逻辑关系提供源程序的自动缩进、括号配对等处理。这样的功能将减少用户在源代码输入时可能犯错误的机会，更有利于提高编程效率。

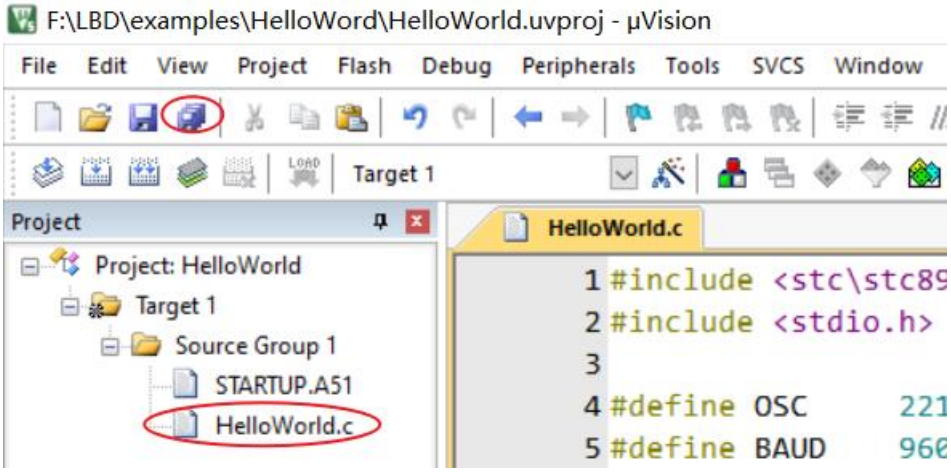
(3) 将源代码加入工程

在上一个步骤中虽然已将源代码保存在一个文件中，但是该文件还必须加入到工程中才能够进行后续的编译、链接等处理。



鼠标右击 Source Group 1，选择 Add Existing Files to Group ‘Source Group 1’，然后选择文件即可将源代码文件加入工程。加入源代码后，窗口左侧的 Project Workspace 显示了工程中各文件之间的关系。

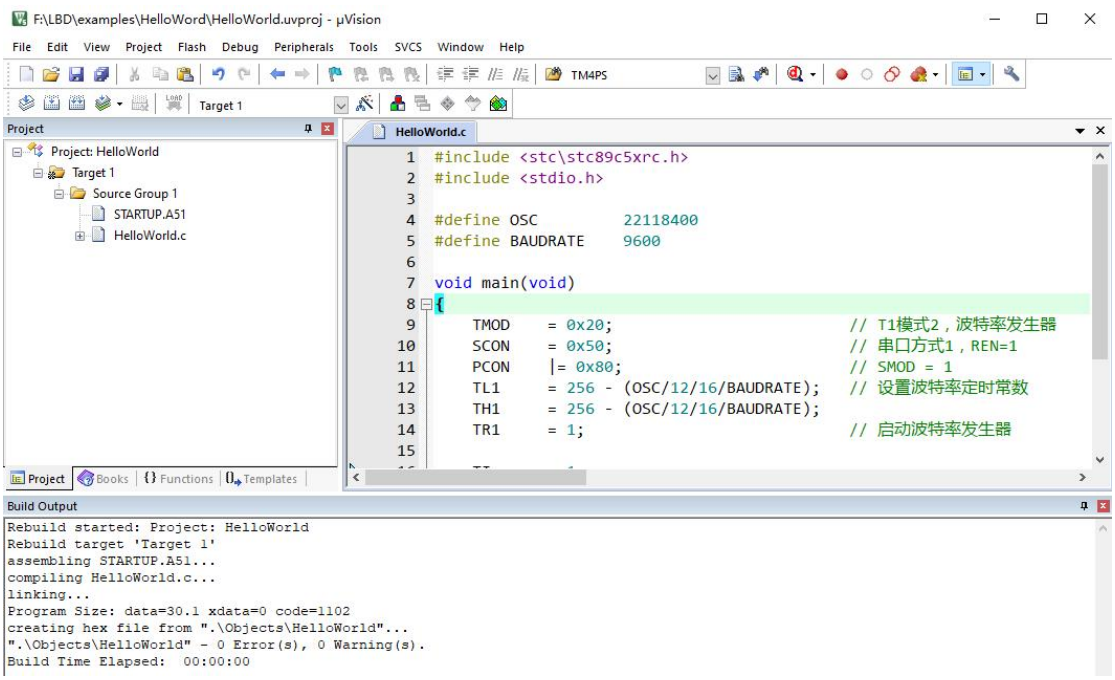
由于可以从外部选择程序加载，因此设计程序时我们也可以使用其它任意一种编辑器（例如 Notepad、UltraEdit 等）来编写源代码。



选择 File→Save All 或点击“保存所有文件”按钮即可保存工程中的所有文件。至此，工程建立完毕。

3.2.2 程序的编译、链接及运行

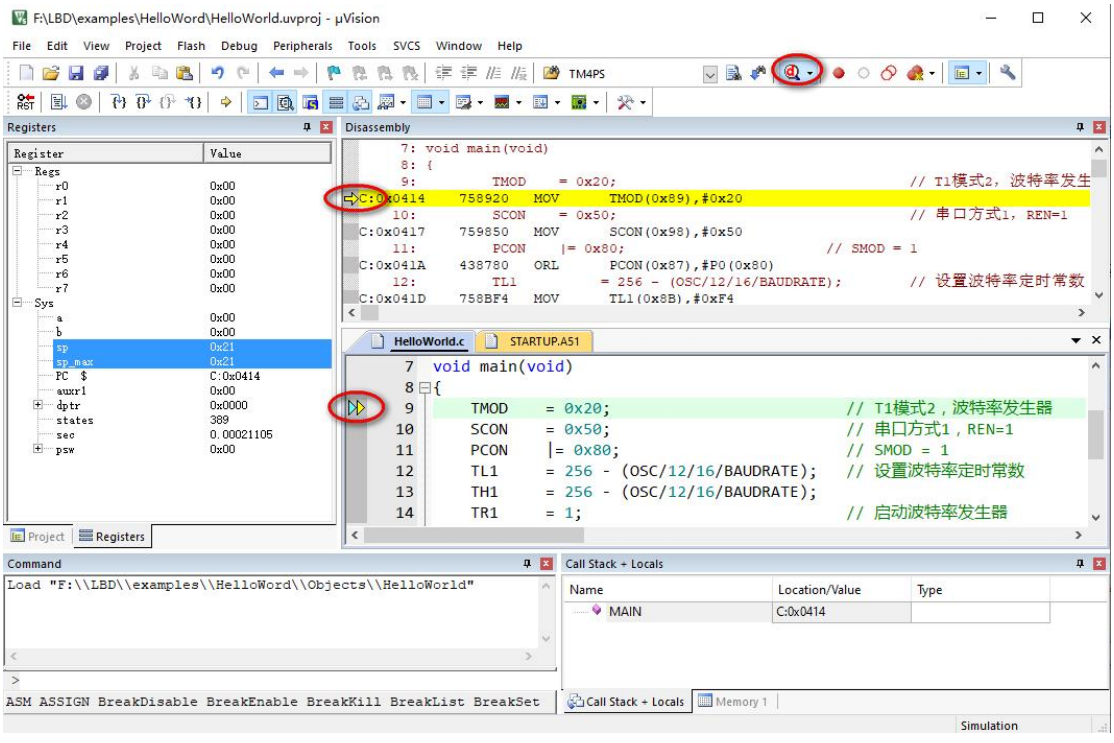
(1) 编译、链接代码



点击 Rebuild all target files 按钮后，Keil 首先保存工程中所有的文件，然后对源程序进行编译、链接，如果在编译或链接的过程中发现错误，Keil 会在 Build Output 窗口给出提示；当源程序没有任何错误后，编译链接完毕会一并生成最终的 HEX 文件。Keil 同样会在 Build Output 窗口中输出编译、链接过程的结果信息。

(2) 程序的调试运行

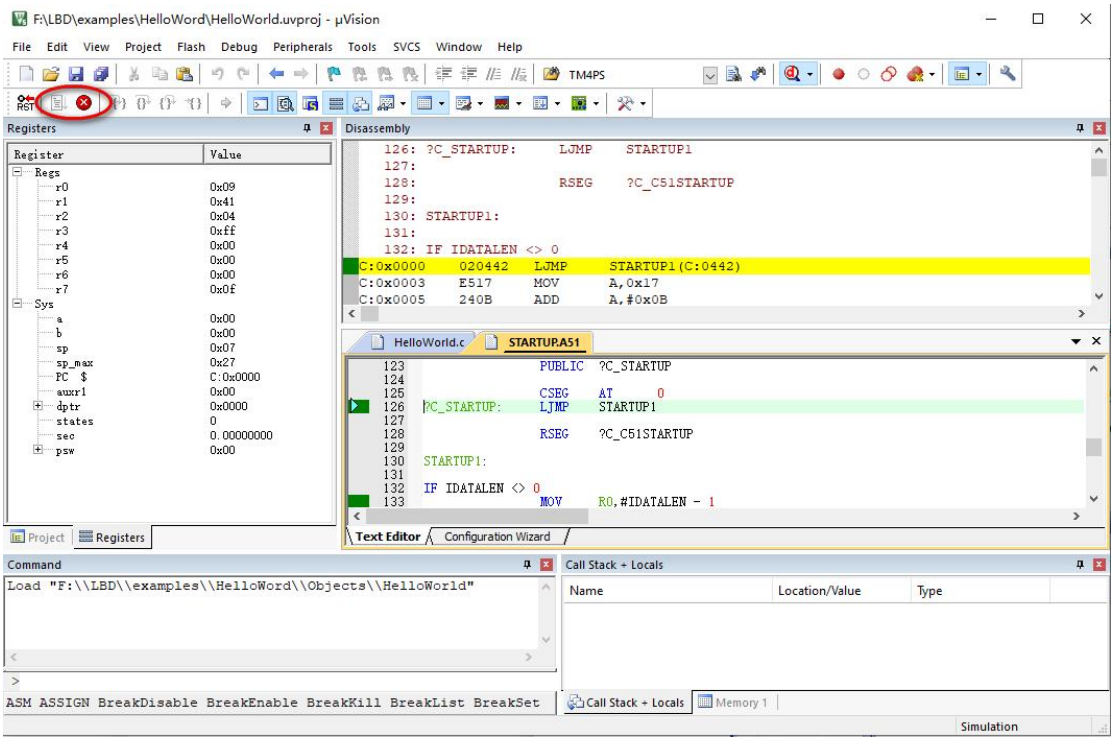
程序设计完成并编译链接成功后，点击 Debug 按钮可以在计算机环境下进行程序的调试运行工作。进入调试模式后，Keil 的界面如下：



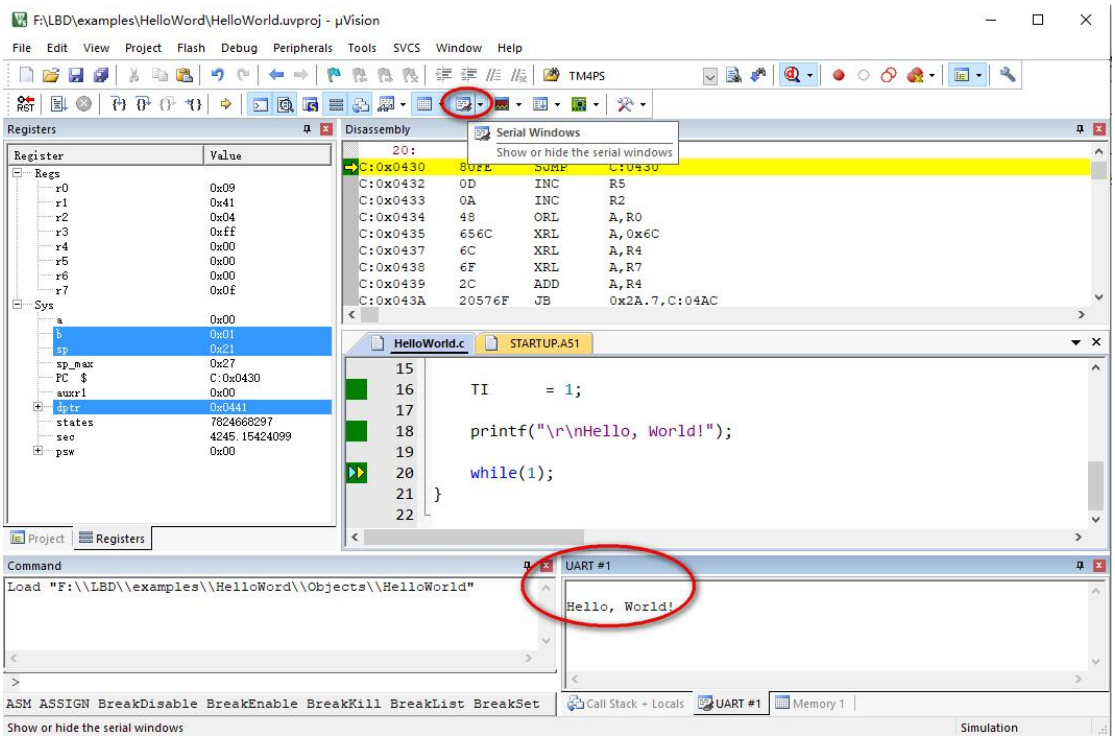
图中反汇编窗口和源程序清单窗口左侧的小箭头表示当前单片机的程序指针所处的位置，即下一条将执行的程序代码的位置。

通过反汇编窗口和源程序窗口的对比，我们可以直观地看到 C 语言源程序和对应的由编译器生成的汇编代码之间的关系，这对我们深刻理解高级语言和汇编语言之间的关系以及计算机程序的本质都是非常有帮助的。

点击工具条中 RST 按钮右侧的 Run 按钮，Keil 将全速运行编译后的程序。



单片机程序在 Keil 中运行起来后，其右侧的 Stop 按钮使能，此时点击 Stop 按钮可停止程序运行。如果需要观察串行口的输出，可点击工具条中的 Serial Windows 按钮，Keil 将打开串行口窗口，并将单片机运行过程中串行口的输出显示在这个窗口中。同时，在该窗口中通过键盘输入的数据将被送往单片机的串行口输入端。



Keil 所提供的运行调试环境外观和 VC 非常相似，功能也很全面。单片机程序在 Keil 中除了可以进行上述的全速运行外，还可以进行单步或多步的跟踪调试，只要在 Debug 状态下点击 Step into（跟踪到函数内部执行）、Step over（直接执行函数）、Step out（直接运行并返回上层函数）、Run to cursor（运行到光标处）等按钮即可，对程序的调试和排错都有极大的帮助。

上述内容对使用 Keil 进行单片机程序开发做了一个简单的介绍，更详细的内容请自行参阅相关资料。

第 4 章 STC ISP 软件使用简介

4.1 背景知识

4.1.1 什么是 ISP

在 Keil 中虽然可以运行和调试程序，但是毕竟只是在计算机环境下的模拟调试，当需要在实际的单片机系统中运行程序的时候，就需要将 keil 编译链接后生成的代码固化到单片机的程序存储器中来执行。以前的单片机没有内部程序存储器，一般都外扩一定容量的程序存储器，需要通过一种称为编程器的专门的设备才能将代码固化到程序存储器中。

现在随着集成电路技术的发展，多数单片机都在内部设置了一定容量的 Flash 存储器作为程序存储器，并设计了启动代码，控制单片机和计算机通信，获取并固化程序代码。通信和固化代码的工作可以在最终的应用系统中完成，因此这种固化代码的方式被称为 ISP（In System Programmable，在系统可编程）。更有一些单片机，可以在应用程序工作的过程中和计算机通信，并分块更新和固化代码，这种固化方式被称为 IAP（In Application Programmable，在应用可编程），极大地方便了系统程序的维护和更新，也为单片机实验提供了一个成本低廉、简单易用的硬件开发环境。

4.1.2 什么是 HEX 文件

Keil 将单片机 C51 源程序通过编译链接，生成最终可被单片机识别、执行的代码。上一章中关于 Keil 工程设置中要求生成的 HEX 文件，就是一种最常见的代码文件。但 HEX 文件中保存的内容却并非是二进制字节，而是用一连串有特定格式约定的文本来表示二进制数据信息。目前最常用的是 HEX 文件格式是 Intel 格式，在 Intel HEX 文件中，每一行是一个 HEX 记录，是由多个文本表示的十六进制数组成的机器码或者数据常量，记录格式如下：

一个 Intel HEX 文件可以包含任意多条的十六进制记录，每条记录有五个域，例如：

:LLAAAATT[DD...]CC

: 冒号，表示一条 Intel HEX 记录的开始；

LL 记录的长度域，表示本条记录包含多少字节的数据(DD…)；

AAAA 地址域，表示本条记录中数据的起始存储地址；

TT 表示本条记录的类型的代码，代码包括：

00：数据记录；

01：文件结束记录；

02：扩展段地址记录；

04：扩展线性地址记录；

对于 MCS-51 单片机，因其程序存储器空间最多为 64K 字节，因此不会用到扩展地址字段；

DD…　数据域，每两个连续的字符表示一个十六进制的字节数据，一条记录可能包含多个数据字节，字节数目由本条记录的 LL 域指定；

CC　校验和域，表示本条记录中除冒号和校验和字节以外所有字节的效验和，计算方法是将本条记录冒号之后，校验和之前的所有字符，以两个字符为一个单位，转化为实际的字节后，全部累加后对 256 取模所得到的余数，再求出余数的补码即是效验和。

对于上一节中 Keil 生成的 HEX 文件，其第一行为：

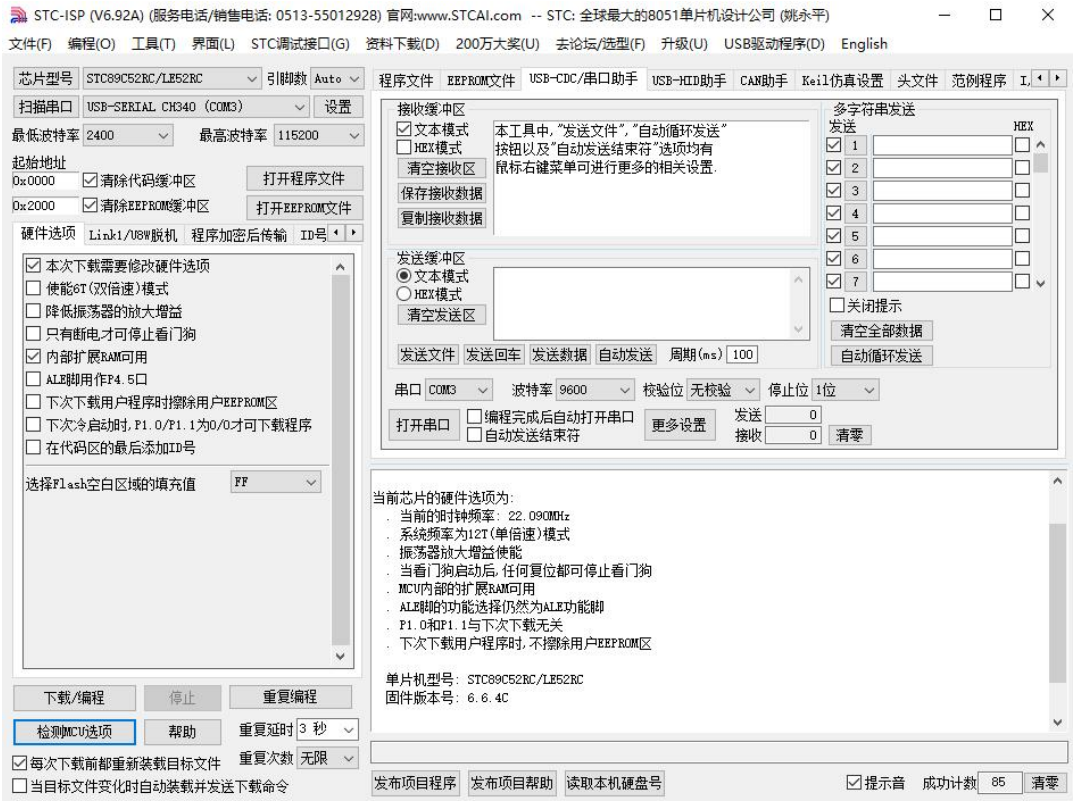
:0300000002047087

对照上述的 HEX 文件格式，可以分析出，本行包含 3 个字节的有效数据，起始存储地址为 0000，类型为 00，表示数据记录，3 个字节的数据为 0x02、0x04、0x70，将冒号后、校验和前的数据字节相加，即 0x03+3 个 0x00+0x02+0x04+0x70 = 0x79，求 0x79 的补码（数据按位取反再加 1）即为 0x87。

4.2 STC ISP 程序

4.2.1 STC-ISP 程序的获取和安装

LBD3 系统使用的 CPU 是 STC 公司的 STC89C52RC，配套的 ISP 软件是 STC-ISP，该软件可访问 STC 官网（http://www.stcmcudata.com）下载，程序是免安装版的，直接解压安装包把可执行文件放在 Windows 桌面即可。双击启动后 STC-ISP 程序的界面如下：



4.2.2 STC-ISP 程序的使用

当我们启动 STC-ISP 程序后，将 LBD3 目标板通过 Micro USB 线接入 PC 机，系统会提示找到新的 USB 设备，并自动安装 USB 转串口驱动程序。如果系统提示找不到驱动程序，请自行访问南京沁恒公司的官网（http://www.wch.cn），搜索 CH340N 芯片的驱动程序并下载安装。安装完毕后再接入 LBD3 目标板，系统将会自动加载 USB 转串口驱动，并生成一个临时的 COM 口设备，例如 COM3。注意，USB 转串口的串口号随不同系统可能会有不同的编号。

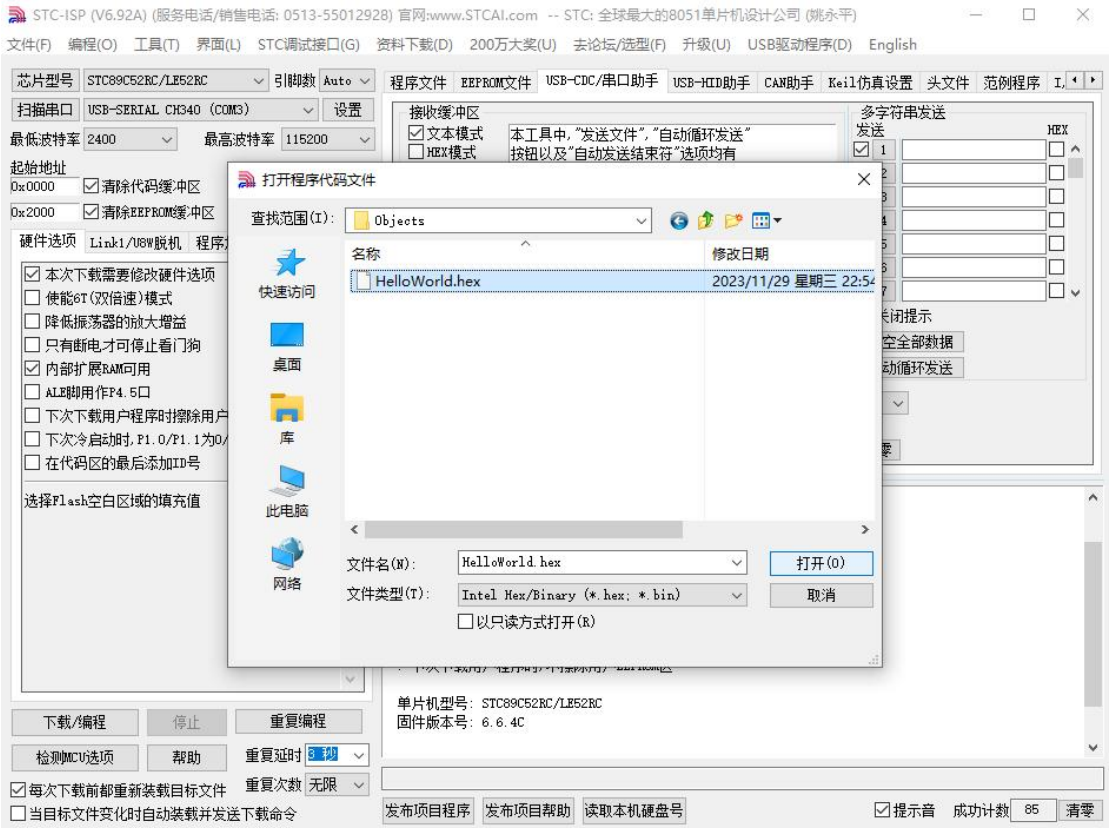
当我们准备使用 STC-ISP 下载编译链接好的单片机程序代码时，请按下列步骤进行：

第一步是选择 ISP 使用的串口号、波特率和单片机型号。波特率可以就是用程序给出的最大、最小缺省值，下载时程序会自动选择。单片机型号一定要选对，之后点击“检测 MCU 选项”按钮，可以在输出窗口看到检测的结果；

第二步是选择要下载的 HEX 文件的路径，点击“打开程序文件”按钮即可进行选择；

第三步点击“下载/编程”按钮即可在输出窗口看到整个在线编程的过程，编程完毕后，单片机系统自动复位，执行新下载的代码。

下图为在 STC-ISP 中选择 HEX 文件的界面，根据 Keil 版本的不同，HEX 文件一般会保存在工程当前目录或者 Listings、Objects 目录下。找到并选中 HelloWorld.hex 文件，点击打开按钮或双击即可将其载入 STC-ISP 中。



打开 HEX 文件后，点击“下载/编程”按钮，单片机将自动复位重启，并在 STC-ISP 程序的控制下开始下载代码，具体的下载过程和结果都会在输出窗口中显示。

HelloWorld 实验代码所完成的工作是通过单片机的串行口送出“Hello World！”字符串。为了方便在计算机上和单片机进行串行通信，STC-ISP 还提供了串口调试助手的功能。选择 USB-CDC/串口助手标签，选择串口号，点击“打开串口”即可进行调试，在下载完成后单片机已经通过串行口输出了 Hello World!字符串，此后我们手动复位一下 LBD 目标板程序就会重新执行，再输出一条该字符串。

当然我们也可以使用其它的串口调试工具，例如 SSCOM、超级终端、CTerm 或者 SecureCRT 等。

从显示结果可以看出，单片机是按照我们程序设计的要求运行的。