

TS8900-R128 实验板学习手册

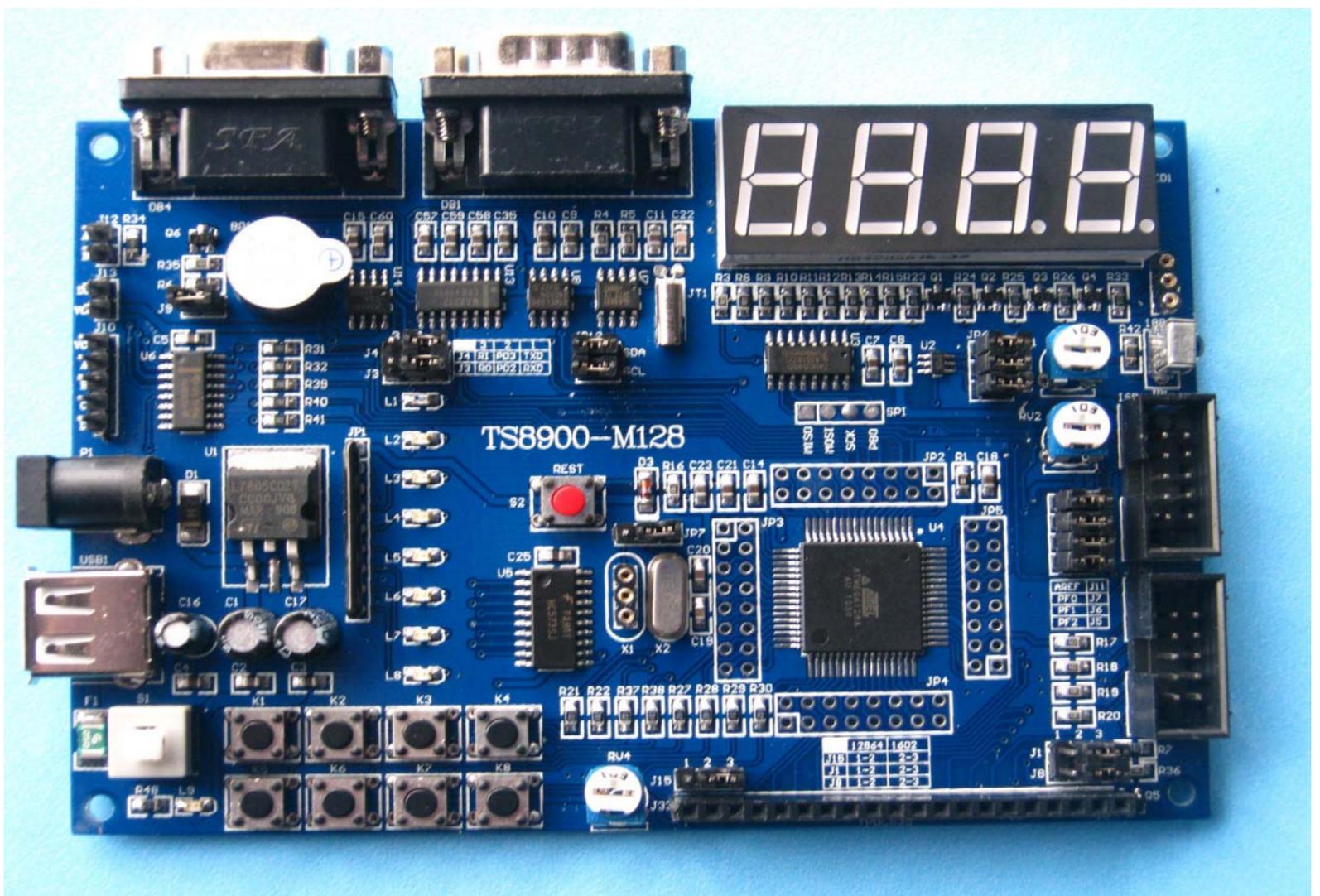
目录

第一章	AVR 单片机的概述.....
第二章	一个 AVR 单片机的最小系统介绍.....
第三章	AVR 单片机的熔丝位介绍 (ATMEGA128)
第四章	TS8900-R128 开发板的硬件介绍.....
第五章	AVR 开发板的开发工具安装及操作介绍.....
第六章	开发板下载器或者仿真器驱动的安装及软件的使用.....
第七章	怎么样用 ICC 编写第一个 AVR 程序.....
第八章	开发板实验指导.....
	1, 流水灯实验.....
	2, 数码管显示实验.....
	3, 蜂鸣器实验.....
	4, 按键实验.....
	5, 中断实验.....
	6, 串口通讯实验.....
	7, AD 转换实验.....
	8, 1602 液晶显示实验.....
	9, 12864 液晶显示实验.....
	10, AT24C02 实验.....
	11, PCF8563T 实验.....
	12, PWM 实验.....
	13, 定时器实验.....
	14, DA 转换实验.....
	15, 步进电机实验.....
	16, 直流电机实验.....
	17, 485 通信实验.....
	18, 红外摇控实验.....

第一章 AVR 单片机开发板概述

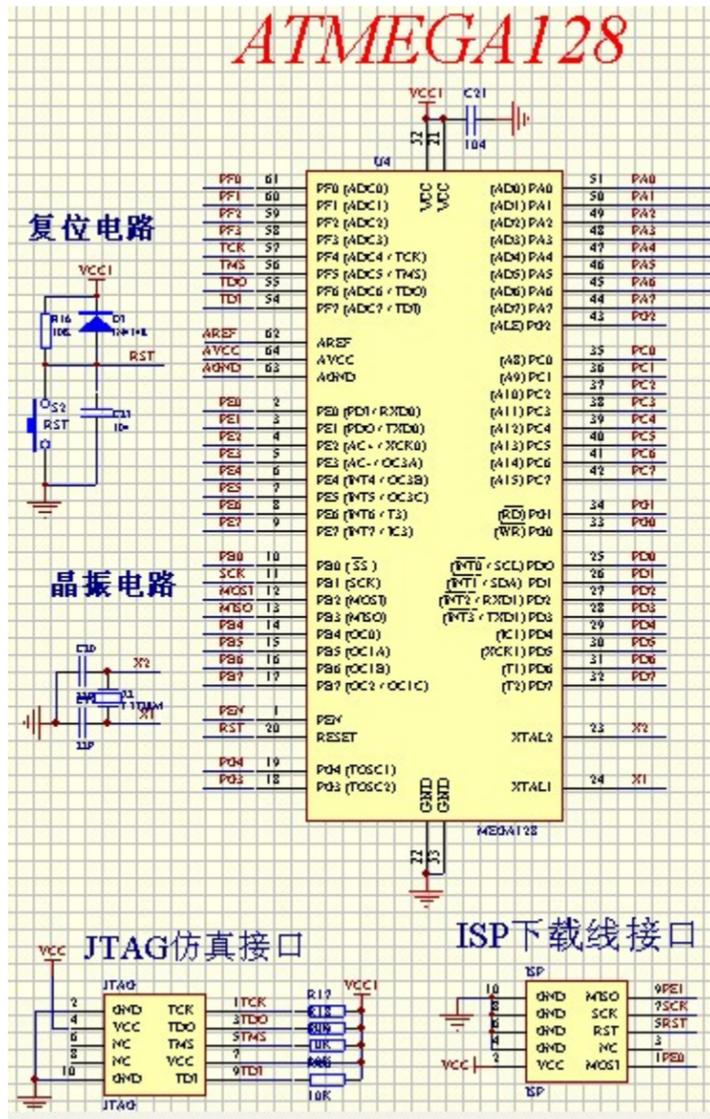
作为电子爱好者或者电子行业硬件工程的开发者，单片机技术的掌握是必须的，但是目前很多的初学者选择入门的单片机都是基于 MCS-51 内核的单片机，虽然此类单片机学习简单，使用方便，但是其性能在很多场合却是大打折扣，要么速度欠缺，要么存储空间欠缺或者耗电量大，因此，由于 MCS-51 自身的结构所限制，与目前的新技术有明显的脱节。AVR 单片机是近 10 年来发展起来的新型的、基于增强型 RISC 结构的单片机。AVR 在运行速度，低功耗，存储器空间，内部功能模块的集成性成，高保密度，多种外围接口（I2C, PWM, ISP, IAP, AD）等，以及在开发技术和仿真，下载调试方面都比 MCS-51 内核的单片机有明显的优势。

TS8900-M128 型单片机学习开发板采用 ATMEGA128 芯片作为主控芯片，是专门针对 AVR 单片机初中高级开发者和电子工程师开发设计的一套多功能系统实验板，它工作稳定，模块化设计，开发资料齐全，可为学习者提供一个较好的实验开发平台。开发板整体图片（如下图：）



第二章 一个 AVR 单片机的最小系统电路

学习 AVR 单片机，首先给大家先介绍一个 AVR 单片机最低要求而且必需的硬件工作图，也就是大家所说的最小系统：



第一，保证有一个给单片机工作的正确的电源（3.3V 或者 5V）；如上图 VCC 电压

第二，时钟振荡电路也就是通常说的晶振电路；(AVR 单片机总体分为内部时钟和外部时钟)；时钟涉及到单片机一个非常重要的问题，熔丝位的配置问题，这在后章会有详细介绍。

第三，一个正确的连接下载口或者 JTAG 口，用来为单片机下载程序或者更新程序，当然，如果是在产品中使用编程器事先给单片机写好程序的话除外；

第四，一个正确的复位电路；（在实际应用中也可以直接接一个 10K 的上拉电阻。）

有了以上四点，那么一个 AVR 单片机最小系统就可以正常工作了。

第三章 AVR 单片机的熔丝位介绍（ATMEGA128 为例）

在 AVR 单片机内部有多组与器件配置和运行环境相关的熔丝位, 这些熔丝位非常重要, 用户可以通过设定和配置熔丝位, 使 AVR 具备不同的特性, 以更加适合实际的应用. 往往用户忽视熔丝位配置的重要性, 随意修改熔丝位, 造成一些意想不到的后果, 如芯片无法正常运行, 芯片死锁, 无法再次进入 ISP 编程模式等; 或者对自己的系统不知如何配置熔丝位是最合适的. 下面将介绍在开始学习和使用 ATMEGA128 时需要特别注意和关心的重要熔丝位的使用配置。

1, 熔丝位 M103C。M103C 的配置将设定 ATMEGA128 是以 ATMEGA103 兼容方式工作运行还是 ATMEGA128 本身的方式工作运行。ATMEGA128 在出厂时 M103C 默认状态为“0”，即默认以 ATMEGA103 兼容方式工作。当用户系统设计使芯片以 ATMEGA128 方式工作时，应首先将 M103C 的状态配置为“1”，即禁止，不然使用者会发现以 ATMEGA128 写的 C 程序在系统上没法正常运行。

2, 存储器加密锁定位，Atmega128 有 2 个加密锁定位 LB1 和 LB2，用于设定对片内存储器的加密方式，用户可在编程方式下，对 LB1 和 LB2 不编程（1）或编程（0），从而获得对片内存储器不同的加密保护方式。需要进一步说明是：

表 2-3 加密锁定位保护方式

加密锁定位			保护方式
模式	LB2	LB1	
1	1	1	无锁定方式（无加密），出厂状态
2	1	0	禁止对 FLASH, EEPROM, 熔丝位的再编程
3	0	0	禁止对 FLASH, EEPROM, 加密锁定位, 熔丝位的再编程和校验

- 在 AVR 的器件手册中，使用已编程（Programmed）和未编程（Unprogrammed）定义加密位和熔丝位的状态。Unprogrammed 表示熔丝位状态为“1”（禁止），Programmed 表示熔丝位状态为“0”（使能）即 1：未编程；0：编程。
- AVR 的加密位和熔丝位可多次编程，其熔丝位不是 OTP 熔丝。
- AVR 芯片加密锁定后（LB2/LB1=1/0, 0/0），在外部不能通过任何方式读取芯片内部 Flash 和 EEPROM 中的数据，但熔丝位的状态仍然可以读取，不能修改配置。
- 需要重新下载程序时，或芯片被加密锁定后，或发现熔丝位配置不对，都必须先在编程状态使用芯片擦除命令，清除芯片内部存储器中的数据，同时解除加密锁定；然后重新下载运行代码和数据，修改和配置相关的熔丝位；最后再次配置芯片的加密锁定位。
- 编程状态的芯片擦除命令是将 Flash 和 EEPROM 中的数据清除，并同时两位锁定位状态配置成无锁定状态（LB2/LB1=1/1）。但芯片擦除命令并不改变其他熔丝位的状态。
- 下载编程的正确操作程序是：在芯片无锁定状态下，下载运行代码和数据，配置相关的熔丝位，最后配置芯片的加密锁定位。

3, 系统时钟类型的配置，（Atmega128 可以使用多种类型的系统时钟源，最常用的为 2 种：使用内部的 RC 振荡源（1/2/4/8MHz）和外接晶体（晶体可在 0~16MHz 之间选择）配合内部振荡放大器构成的振荡源。具体系统时钟类型的配置由 CKOPT 和 CKSEL [3: 0] 共 5 个熔丝设定，表 2—4 和 2—5 给出了具体的配置值。在使用中，用户首先要根据实际使用情况进行正确设置，而且千万注意不要对这些熔丝位误操作。

(AVR 单片机在售出时, 片内的 FLASH 存储器和 EEPROM 存储器阵列是处在擦除的状态(即内容=\$FF), 且可被编程. 同时其器件配置熔丝位的默认值为使用内部 1MHz 的 RC 振荡源作为系统时钟)

AVR 单片机为用户提供了更多灵活选择系统时钟的可能性, 以满足和适应实际产品的需要. Atmega128 在片内集成有内部可校准的 RC 振荡器, 能提供固定的确 1/2/4/8MHz 的系统时钟, 这些频率是在 5V、25° C 时的标称数值. CKOPT 和 CKSEL 熔丝按表 2-4 编程配置时, 可以选择 4 种内部 RC 振荡源之一作为系统时钟使用, 此时将不需要外部元件。

当产品对系统时钟的精度要求比较高, 或需要使用一些特殊频率的系统时钟场合时, 例如使用了 USART 通信接口, 系统时钟频率需要使用 4.6080/7.3728/11.092MHz 时, 就要使用外接晶体(晶体可在 0~16MHz 之间选择)配合内部振荡放大器构成振荡源, 具体连接电路如图 2-6 (a) 所示. 此时需要将 CKOPT 和 CKSEL 熔丝按表 2-5 编程配置。

表 2-4 系统时钟类型为使用内部 RC 振荡源

CKOPT	CKSEL [3:0]	工作频率范围/MHz
1	0001	1.0 (出厂设置)
1	0010	2.0
1	0011	4.0
1	0100	8.0

表 2-5 使用外部晶体与片内振荡放大器构成的振荡器

熔丝位		工作频率范围/MHz	C1, C2 容量/Pf (仅适用石英晶振)
CKOPT	CKSEL [3: 0]		
1	101X	0.4~0.9	仅适用陶瓷振荡器
1	110X	0.9~3.0	12~22 (应与使用晶体配合)
1	111X	3.0~8.0	12~22 (应与使用晶体配合)
0	101X, 110X, 111X	≥1.0	12~22 (应与使用晶体配合)

在表 2-5 中, 当 CKOPT=0 时, 振荡器的输出振幅较大, 容易起振, 适合在干扰大的场合以及使用的晶体超过 8MHz 时的情况下使用; 而当 CKOPT=1 时, 振荡器的输出振幅较小, 这样可以减少对电源的消耗, 对外的电磁辐射也较小。

4, JTAGEN。如果不使用 JTAG 接口, 应将 JTAGEN 的状态设置为“1”, 即禁止 JTAG, JTAG 引脚可以用于 I/O 口。如果要使用 JTAG 时, 应将 JTAGEN 的状态设置为“0”, 即使能 JTAG, JTAG 的四个引脚将不能用于 I/O。

5, SPIEN。SPI 方式下载数据和程序允许, 默认状态为允许“0”, 一般保留其状态。建议大家一定不要把 SPIEN 熔丝位设定为 1, 即关闭 SPI 下载功能, 此点非常重要, 在你关掉 SPI 下载功能时以后对你弄错熔丝位解锁就非常不方便了。

6, WDTON。看门狗的定时器始终开启。WDTON 默认为“1”, 即禁止看门狗的定时器始终开启。如果该位设置为“0”后, 看门狗的定时器就会始终打开, 不能被内部程序控制了, 这是为了防止当程序跑飞时, 未知代码通过写寄存器将看门狗定时器关断而设计的(尽管关断看门狗定时器需要特殊的方式, 但它保证了更高的可靠性)。

7, EESAVE。执行擦除命令时是否保留 EEPROM 中的内容, 默认状态为“1”, 表示 EEPROM 中的内容 FLASH 中的内容一同擦除。如果该位设置为“0”, 对程序进行下载前的擦除命令只会对 FLASH 代码区有效, 而对 EEPROM 区无效。这对于希望在系统更新程序时, 需要保留 EEPROM 中的数据的情况下是十分有用的。

8, BOOTRST。决定芯片上电启动时, 第一条执行指令的地址。默认状态为“1”, 表示启动时从 0X0000 开始执行; 如果 BOOTRST 设置为“0”, 则启动时从 BOOTLOADER 区的起始地址处开始执行程序。BOOT LOADER 区的大小由 BOOTSZ1 和 BOOTSZ0 决定, 因此其首地址也随之变化。BOOT 复位熔丝位配置如下表:

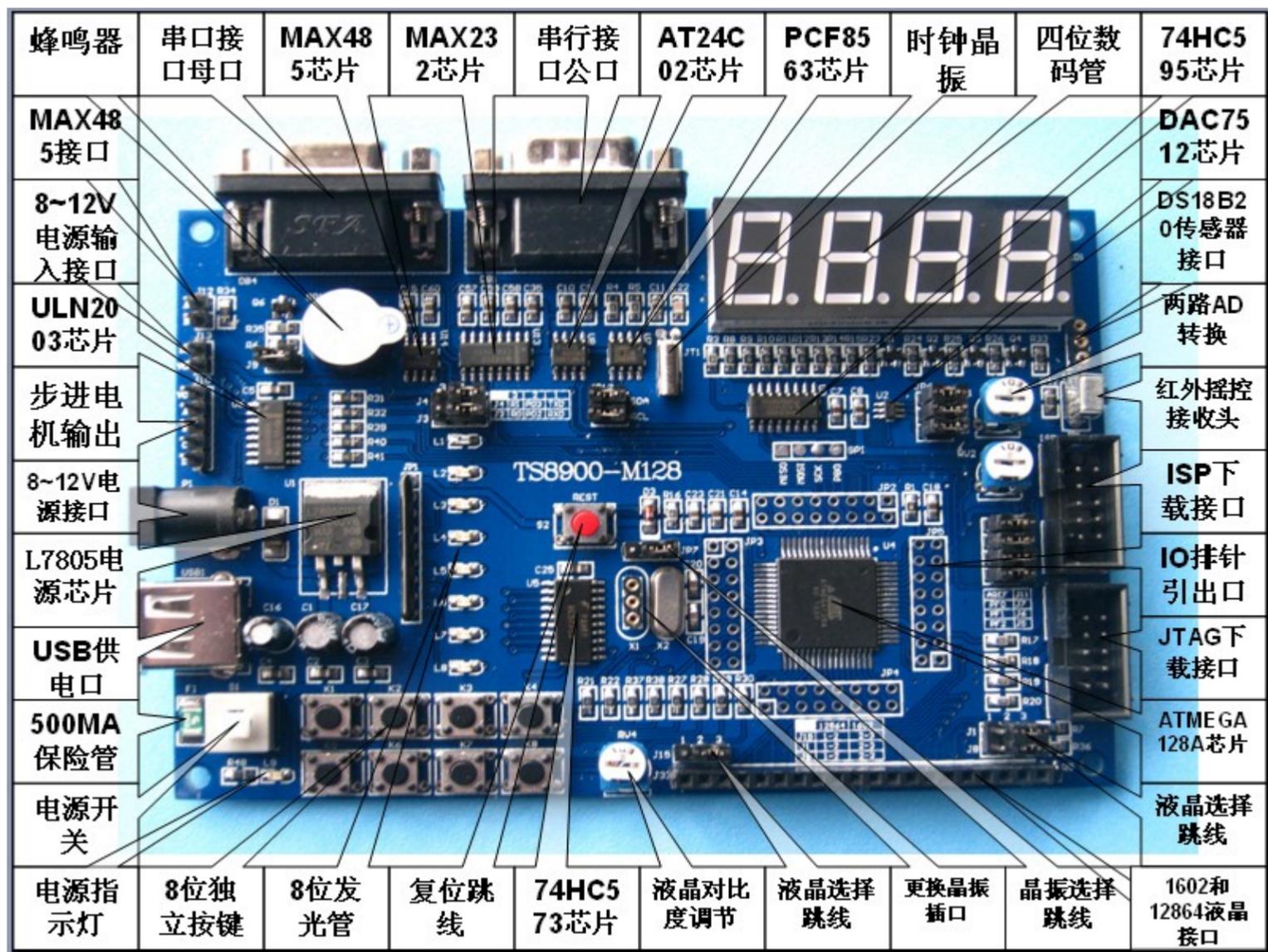
BOOTRST	复位地址
1	应用区复位 (地址\$0000)
0	BOOTLOADER 复位

9, BOOTSZ1 和 BOOTSZ0。这两位确定了 BOOTLOADER 区的大小以及其超始的首地址。默认的状态为“00”，表示 BOOTLOADER 区为 4096 字，超始首地址为 0XF000。BOOT 区大小配置如下表：

BOOTSZ1	BOOTSZ0	BOOT 区大小	页数	应用 FLASH 区	BOOTLOADER FLASH 区	应用区结束地址	BOOT 复位地址 (BOOTLOADER 起始地址)
1	1	512 字	4	\$0000~\$FDFE	\$FE00~\$FFFF	\$FDFE	\$FE00
1	0	1024 字	8	\$0000~\$FBFF	\$FC00~\$FFFF	\$FBFF	\$FC00
0	1	2048 字	16	\$0000~\$F7FF	\$FB00~\$FFFF	\$F7FF	\$FB00
0	0	4096 字	32	\$0000~\$EFFF	\$F000~\$FFFF	\$EFFF	\$F000

第四章 TS8900-R128 开发板的硬件介绍

一，硬件资源介绍(如下图)：

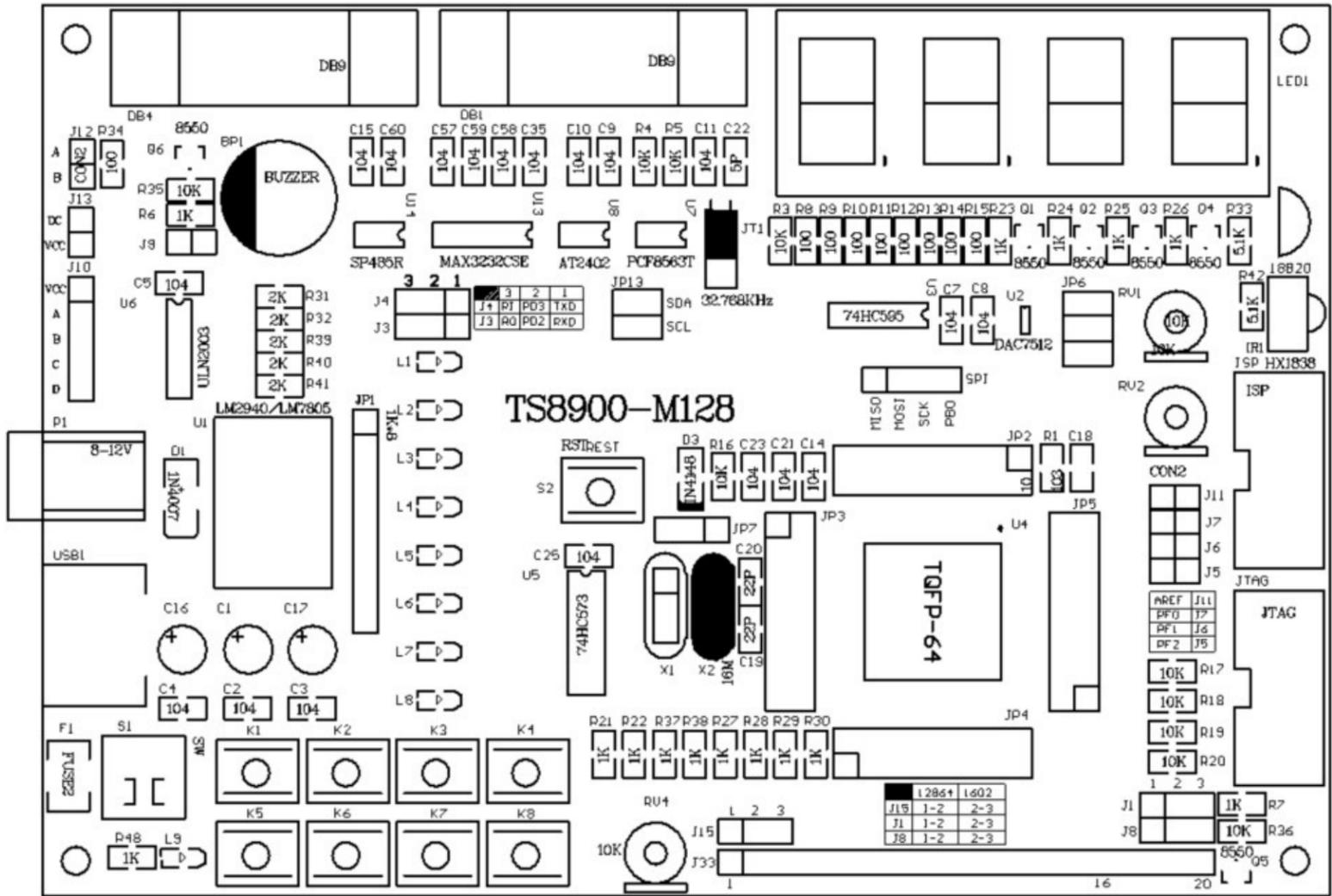


1. CPU 核心板 mega128, AVR 单片机主芯片, 内包含有: 128K Flash、4K SRAM、4KByte EEPROM, 芯片可与 AT mega64 互换。

2. AT24C02, I²C 总线器件, EEPROM 数据存储, 256Byte EEPROM。
3. 74HC595, SPI 总线器件, SPI 总线并口扩展器件, 8 位移位寄存器 (串行输入, 3S 并行锁存输出)。
4. PCF8563T, 时钟芯片。
5. ULN2003 步进电机和直流电机驱动芯片。
6. HX1838 红外一体化接收头。
7. RS232, RS232 接口芯片, 两组接口 (和下面 485 接口共用 CPU 两组 UART 接口)。
8. MAX487, 485 网络驱动芯片, 产品组网中最常用最方便的一种方式, 大部分室外控制网络采用 485 通信。
9. 8 个输入按键、1 个复位按键。
10. 4 位动态扫描 LED 数码管、8 位 LED 发光二极管。
11. 1 个无源蜂鸣器。
12. 2 路 A/D 输入可调电位器。
13. 1 个三线扩展接口, 可用来扩展热敏电阻、DS18B20 数字温度计等实验。
14. LCD 接口, 液晶显示屏接口, 可接 LCD1602 和 LCD12864 两种液晶屏。
15. AVR JTAG 仿真接口, AVR ISP 下载接口。
16. 所有 IO 口全部引出, 方便扩展实验及测试。
17. CPU 采用独立 mega128 核心板, 方便升级及更换芯片。
18. 在 1 路 RS232 外提供了 1 路 RS485 网络驱动芯片, 实用的多点远程通信网络。
19. 带有高速光耦, PWM 隔离输出, 中断隔离输入;
20. 支持外接电源也支持 USB 电源供电;

二, 开发板跳线说明:

TS8900-M128 开发板丝印图如下:



J12 跳线——MAX485 通信接口；

J13 跳线—— 直流电机接口；

J10 跳线——步进电机接口；

J3, J4 跳线——MAX485 和串口实验跳线选择端；J3, J4 同时选择 1-2 脚则选择了串口通讯，同时选择 2-3 脚则选择了 485 通信。

JP13 跳线——I2C 实验跳线选择端；

JP6 跳线——进行 DA 实验时跳线选择端；

JP7 跳线——外加晶振时钟选择端；跳到右边两个则选择 16M 晶振，跳到左边两个则选择更换晶振；默认跳到右边；

J5 跳线——DA 输口连到主芯片的 (PF2) AD 输入端，可进行 DA 检测实验。

J6, J7 跳线——两路 AD 输入选择端。

J11 跳线——AD 模数转换参考电压选择端。

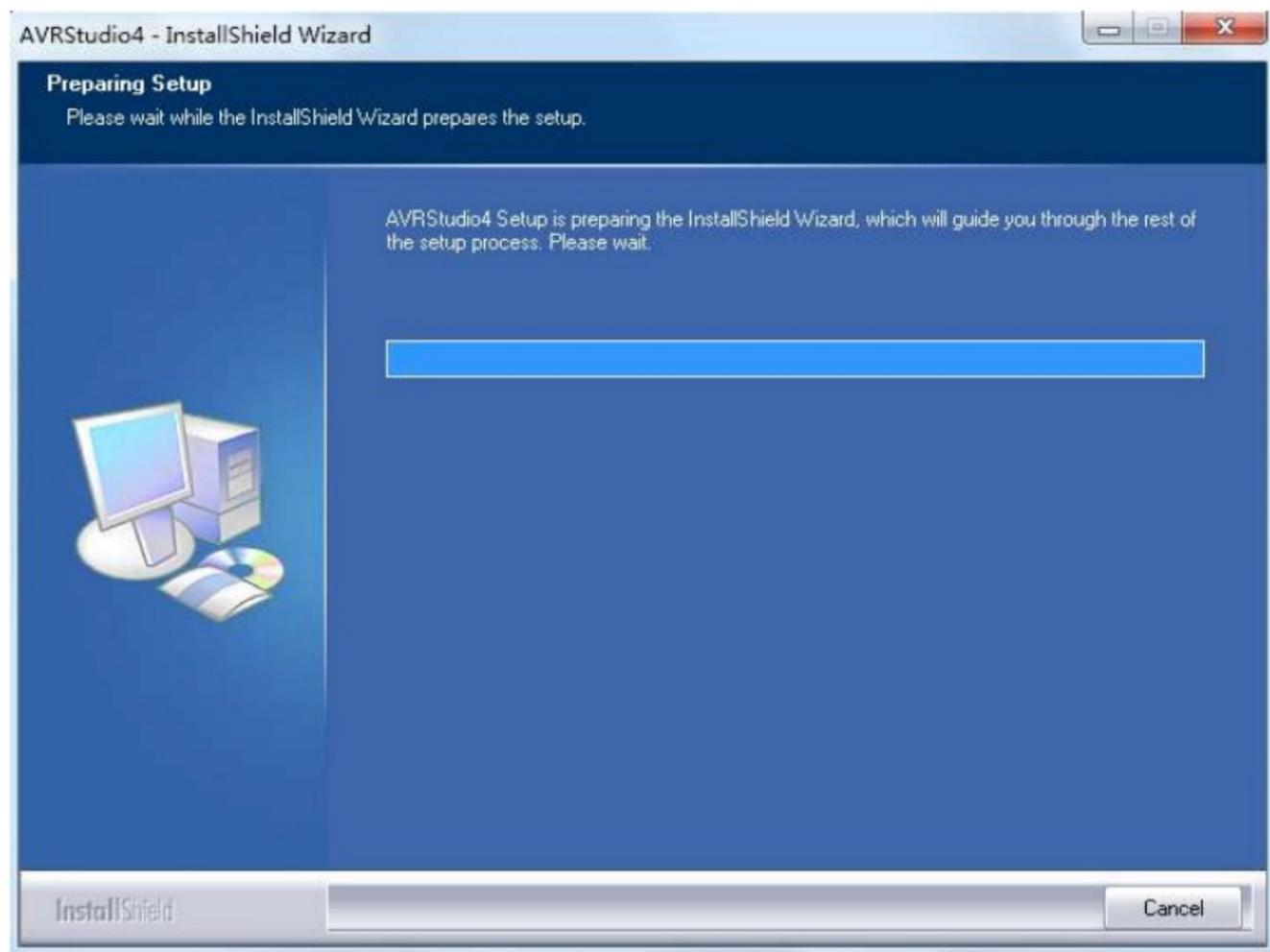
J15, J1, J8 跳线——12864 液晶和 1602 液晶跳线选择端口；三根跳线同时选择 1-2 则可进行 12864 实验，同时选择 2-3 则可进行 1602 液晶时验；

第五章 AVR 开发板的开发工具安装操作介绍

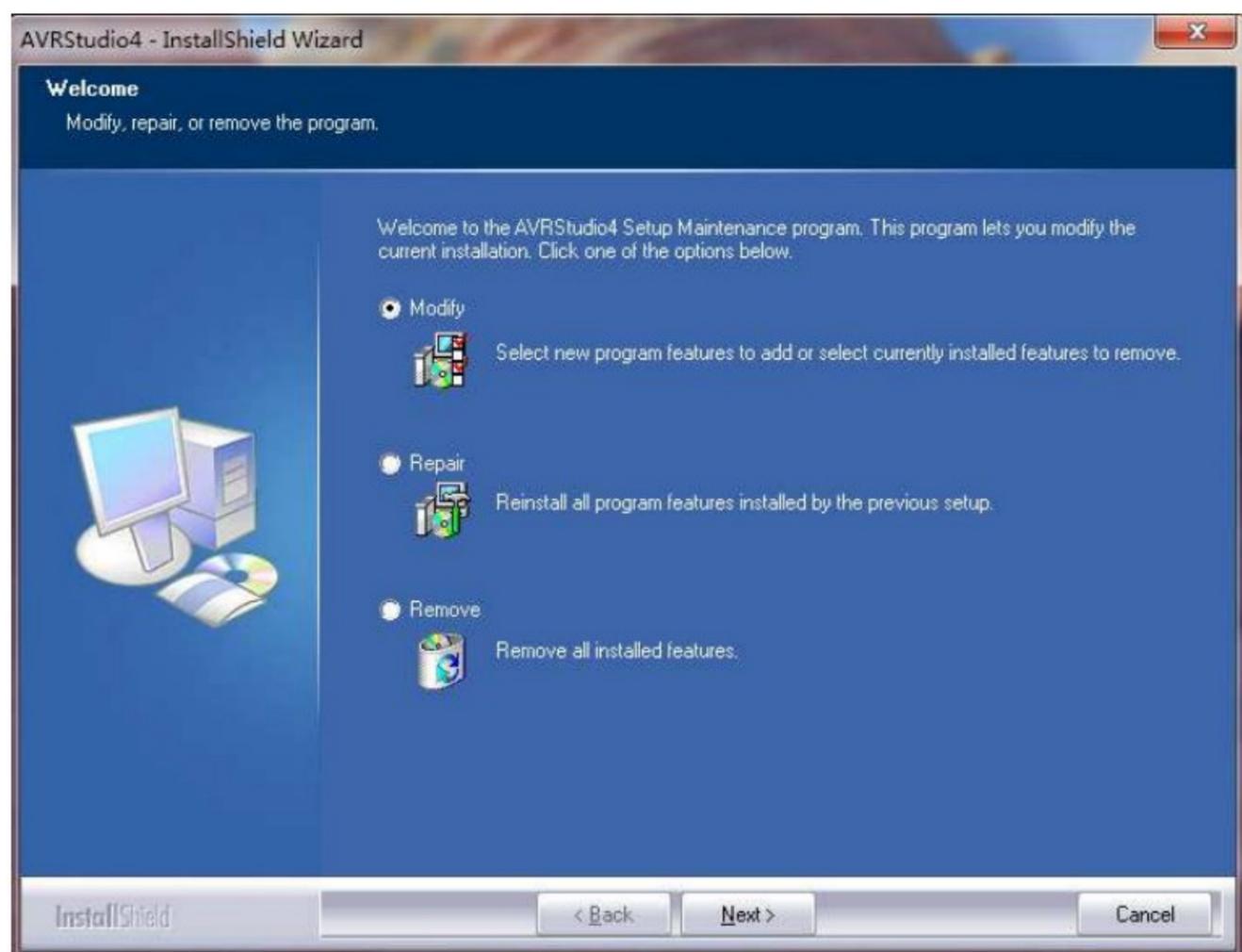
下面介绍 AVR GCC (AVR Studio & WinAVR) 软件开发平台和 ICCAVR 平台的安装，其他平台的安装请参照相关资料自行选择

一. AVR STUDIO 软件的安装

将提供的光盘放入光驱，打开“AVR 系统平台软件”文件夹，里面提供各种常用的 AVR 单片机开发平台软件环境，找到“AVR Studio 开发环境”文件夹，里面有 AVR Studio 4.17 安装包 exe 文件，双击安装文件即可开始安装，如下图



选择“modify”进行安装，如下图所示：

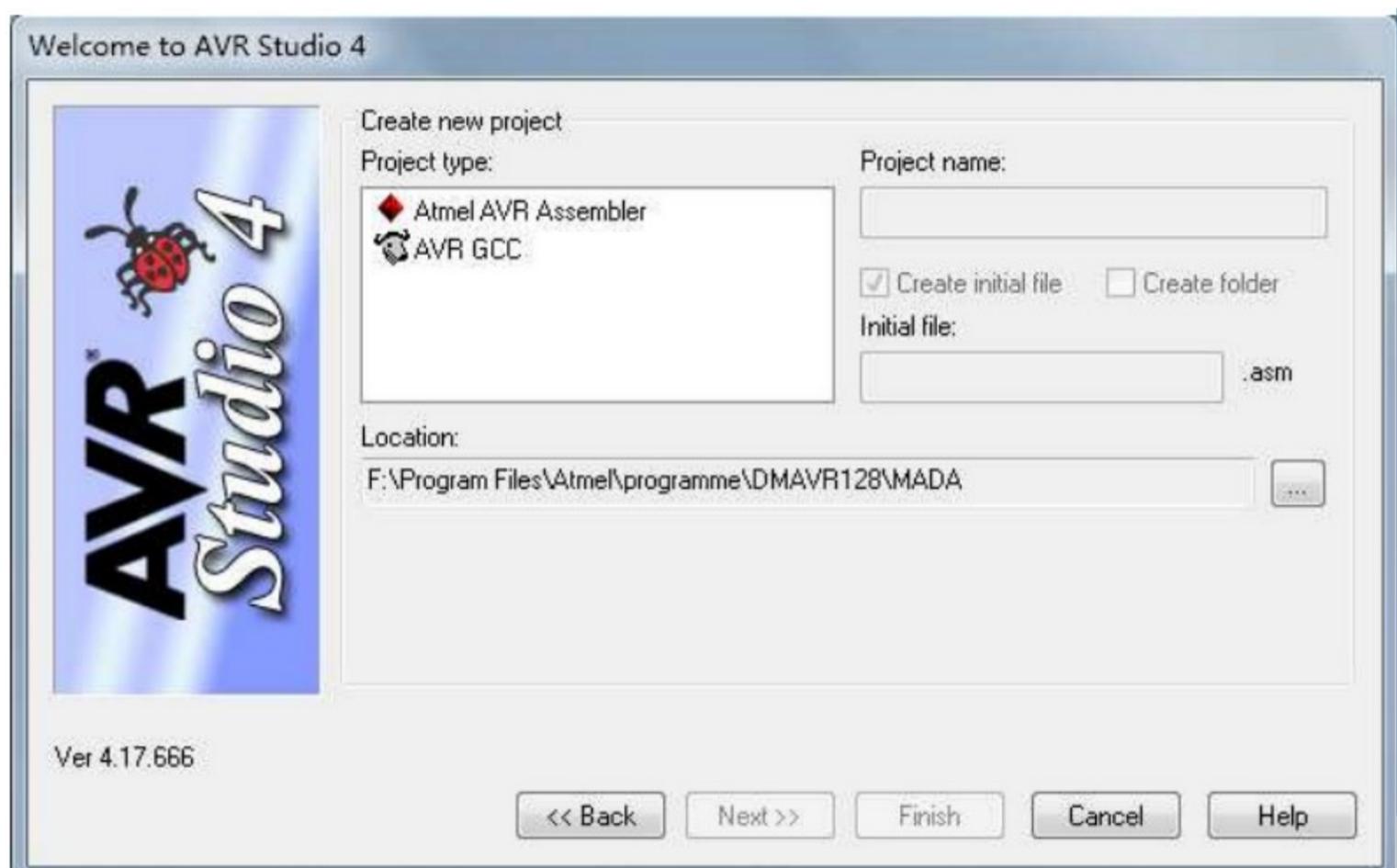


按照提示依次进行安装，选择合适的路径，安装完毕，会给出提示。 点击“finish”完成安装。到此，AVR Studio 平台安装完毕

二、WinAVR 编译器的安装

AVR Studio 平台只是提供汇编语言开发平台以及系统软件仿真平台，在实际应用中，几乎很少用完全的汇编语言完成系统软件的开发，基本都选择高级言开发软件。AVR Studio 平台提供 AVR GCC 接口，因此，要在 AVR Studio 环境下使用高级语言，必须安装高级语言编译器 WinAVR。

在光盘中，打开“AVR Studio 4.17&WinAVR”文件夹下，运行 WinAVR 安装包 WinAVR-20090313-install.exe，选择合适的安装路径，完成高级语言编译器的安装(WinAVR 安装对路径有要求，不能出现汉语路径，为了不在后续编译过程中出现错误，建议用户安装在 建议用户安装在 建议用户安装在 AVR Studio 安装路径下，如果用户没有改动，在 AVR Studio 路径下应该有个 Atmel 文件夹，可以在该文件夹下新建一个文件夹如 WinAVR，将 WinAVR 安装在新建的 WinAVR 文件夹下)。 安装完成后，打开 AVR Studio 软件，可以看到 AVR GCC 选项，在新建项目时可以根据需要选择是使用汇编语言还是高级语言进行程序的开发，打开后的软件提示界面如下：



至此，一套完整的 AVR GCC 软件平台(可使用汇编和高级语言)就完成安装了，如果熟悉的用户就可以应用进行开发了。

三. ICCAVR 平台的安装

在光盘“AVR 系统平台软件”文件夹下有“ICCAVR 开发环境文件夹”，里面包含 ICCAVR 平台的评估版本和两个破解版本 V7.16A 和 V7.22，用户可以任意使用这两个版本，注意使用 V7.16A 版本的用户请参考使用的说明文档自行进行相关操作，我们在这里使用 V7.22 来演示操作，在刚才我们打开的文件夹下双击安装文件 iccv7avrV7.22Setup.msi 即可，出现安装向导，如下图：



根据安装向导，选择安装路径，如：



按照提示继续进行，安装完成后，点击“完成”即可。

四 PDF 文档阅读器的安装

系统配套光盘提供的绝大部分文档资料都是 PDF 的格式，要正常阅读资料，需要先安装 PDF 阅读软件，配套光盘提供了 PDF 阅读软件，打开“PDF 阅读软件”文件夹，双击安装包即可安装。

五 其他软件

系统板配套光盘提供了一些其他的软件，如 IAR for AVR、CVAVR 等软件等，这些软件的安装也不是必须的，是作为附带软件提供给用户，方便部分用户的选择使用，串口通信时，使用串口调试助手作为客户端，方便快速开发程序。

第六章 开发板下载器或者仿真器驱动的安装及软件的使用

安装完成软件驱动后，如果用户购买的套餐包含了 USBASP 下载器或者仿真器，那么必须得安装相应的硬件驱动，以方便使用 USBASP 下载程序和仿真器下载或者仿真调试程序。如果用户使用自己的开发工具，请参考相应的使用方法，这里介绍我们的开发工具的驱动安装方便及软件的使用。

一. USB 型的 ASP 下载器的驱动安装及软件使用：

支持 XP 和 WIN7 及 64 位系统重要提示：请在连接目标板、学习板前，仔细阅读本说明，尤其是“连接电路”部分的内容，以确保本下载器的接口标准与您的目标板、学习板接口标准相同！

USB 接口：直接插电脑 USB 插座。

IDC-10 接口：ATMEL 标准接口定义：

Pin1: MOSI 信号

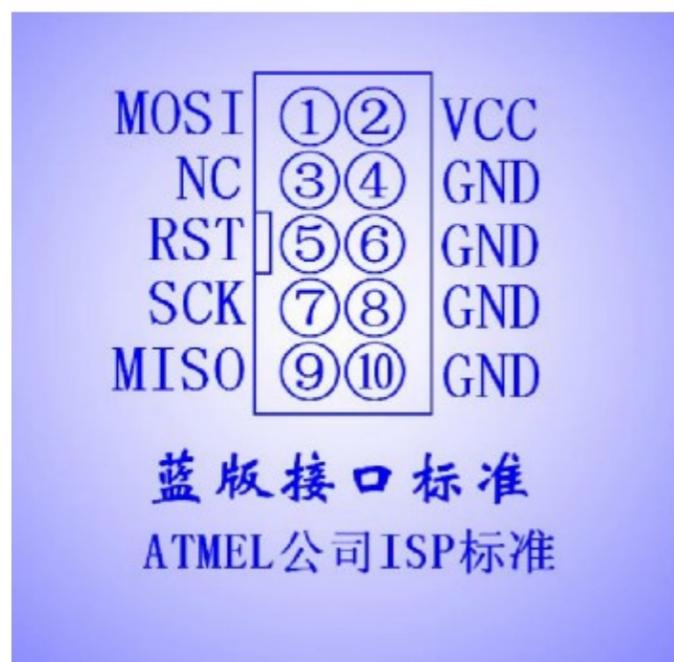
Pin2: 向外供电 5.0V（可选，取决于 J1 是否连接）可向目标板供电（取决于

Pin4、6、8、10: 接地

Pin5: 复位信号。自动识别 51/avr 单片机，自动采用高/低电平复位

Pin7: SCK 同步时钟信号

Pin9: MISO 信号



供电跳线 J3: 设置 IDC-10 的 pin2 是否向目标板供 5V 电源。注意：如果目标板

自带电源，则建议把该跳线断开。如果该跳线连接，则 pin2 将有 5V 电源。此电源为 USB 供电，谨防短路和过载，以免烧坏

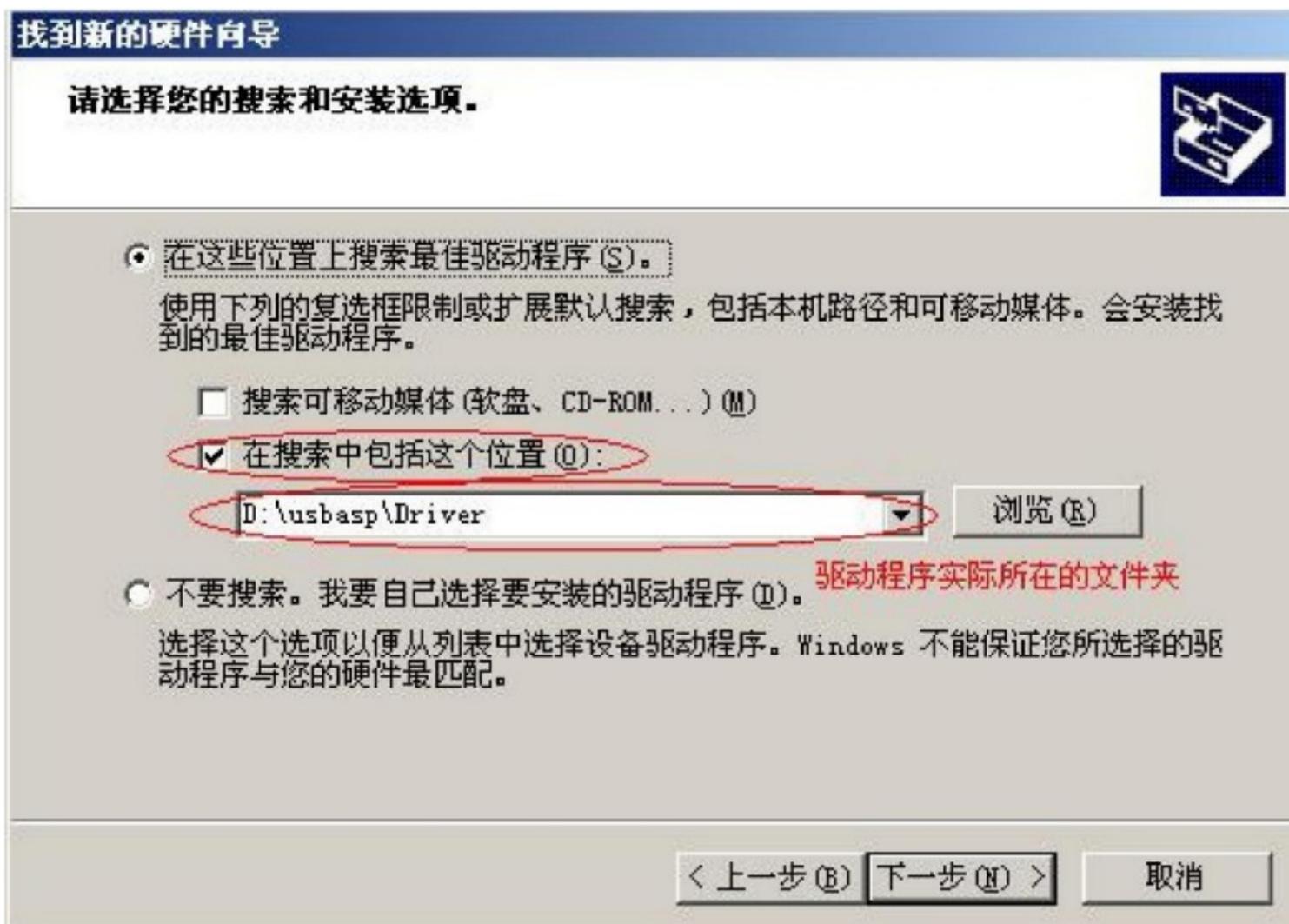
电脑主板！（过流短路保护板带有 500mA 保险丝，可以减少情况带来的危害）

升级跳线 J4: 升级固件时短路，正常使用时开路。

(注意：自己升级造成的下载器故障，可邮寄回本店，免费维修，邮费自理。但本店拒绝直接提供未加密的固件程序！)

USBAP 驱动程序安装：将 USBasp 插入电脑 usb 接口后，系统提示找到硬件，按下列图示安装。



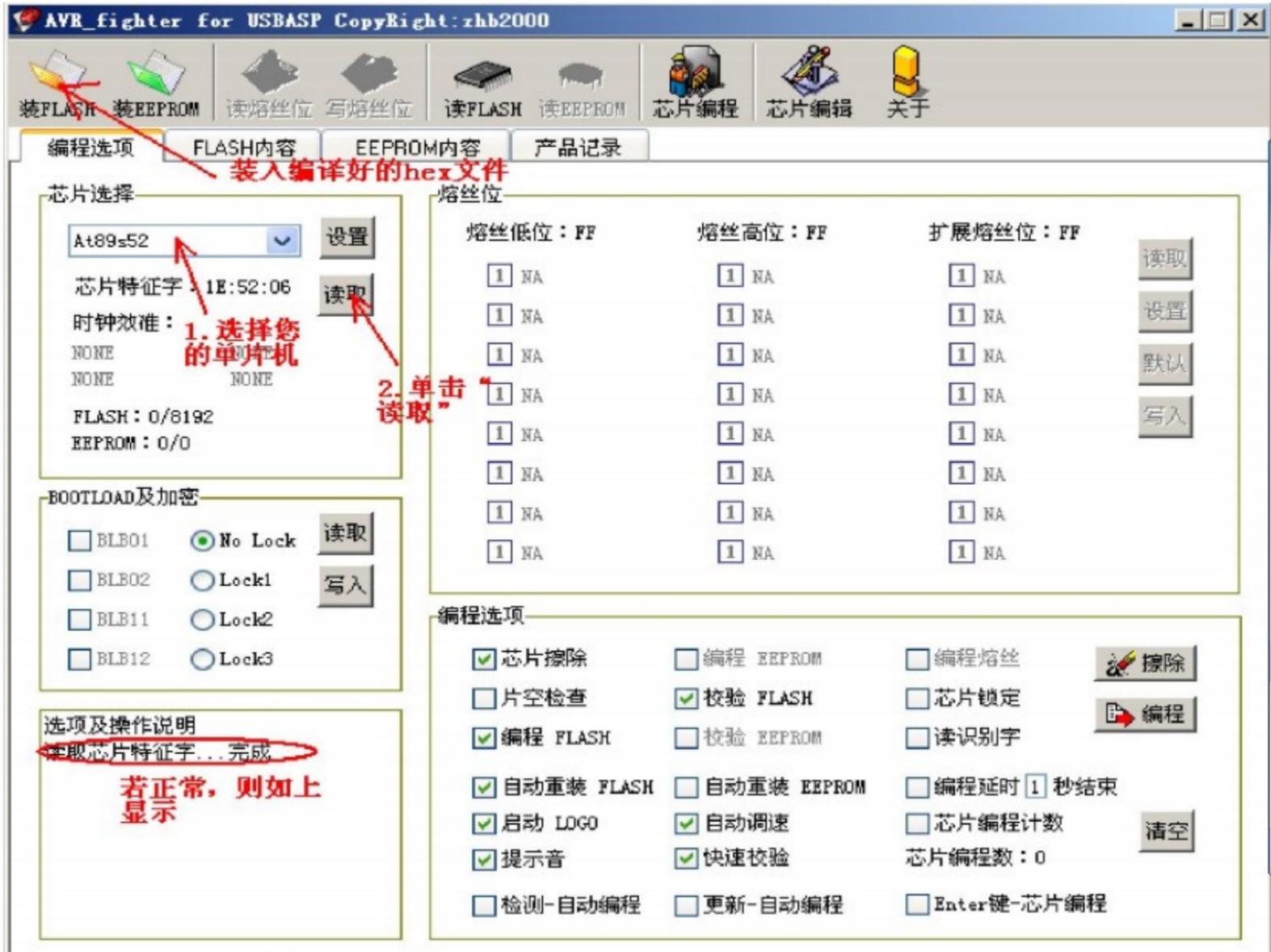


安装成功后，在设备管理器中应该能发现下面的 USBasp 硬件



连接电路（请仔细阅读） 将下载线与单片机正确地相连，那是很重要的。常见的 isp 下载线的接口标准较多，仅 10pin 的标准就有两种。而且这两种标准严重不兼容，插错的话会造成短路事故！请仔细阅读下列说明后，再动手操作！ 本下载线采用的是兼容 ATMEL 的标准。如果你手头有现成的开发板，那么，在您把下载线插头插到你的开发板上之前，请仔细核对您的开发板的下载线接口是否与本下载线接口相同！如果忽略这一步，可能造成下载线、开发板甚至电脑主板的烧毁！只要你能意识到这一点区别，就没那么可怕了。因为不管下载线的接口定义是什么，它们所用到的信号总是那么四个：SCK、MOSI、MISO、RST，还有就是电源 VCC 和地 GND。你只要确定这么几个信号都对上了，就 ok 了。上面介绍了 128 最小系统就有 ASP 下载器和 ATMEGA128 芯片的接线方法(请参考,不同的芯片可能连接方式还不同)。

上位机软件： AVR_Fighter （如下图操作界面）和智峰的 progisp168 都好用！

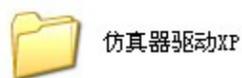


二. USB 型仿真器 JTAG ICE 的驱动安装及软件使用

1. 驱动的安装:

首先看你的电脑是 XP 系统还是 WIN7 系统, 这驱动安装有点复杂, WIN 还分 32 位系统和 64 位系统, 在这我的介绍一下 XP 系统下仿真器驱动的安装, 如果有不太懂或者没装好驱动的朋友可以直接通过 QQ 跟我们沟通.

1.1 最通用的就是 XP 系统了, 在安装仿真器 XP 驱动前, 请不要把你的仿真器插到了电脑上面, 或者 USB 口上不能插有别的 USB 转串口的外插设备, 有的话请拔下来, 电脑 USB 口上这时只能保持有一个 USB 的鼠标就可以了, 再双击安装驱动, 如下图. 在这个文件夹



双击打开直到底, 如下图:



再双击  图标. 显示如下:



再点击下一步,直到完成,即安装完成仿真器的 XP 驱动,在此提醒一下大家,安装驱动时要确保仿真器没有插到电脑上面哦,还有相关的外插设备也要拔掉,如果安装仿真器驱动时,仿真器插到了电脑上面或者有外接相关设备(主要指 USB 转串口设备,可能软件有冲突)在上面,那就需要重新安装驱动了,重新安装驱动大家要注意几个步聚了(目的是先移除前面安装的驱动再重新启动电脑,再安装驱动),如下:

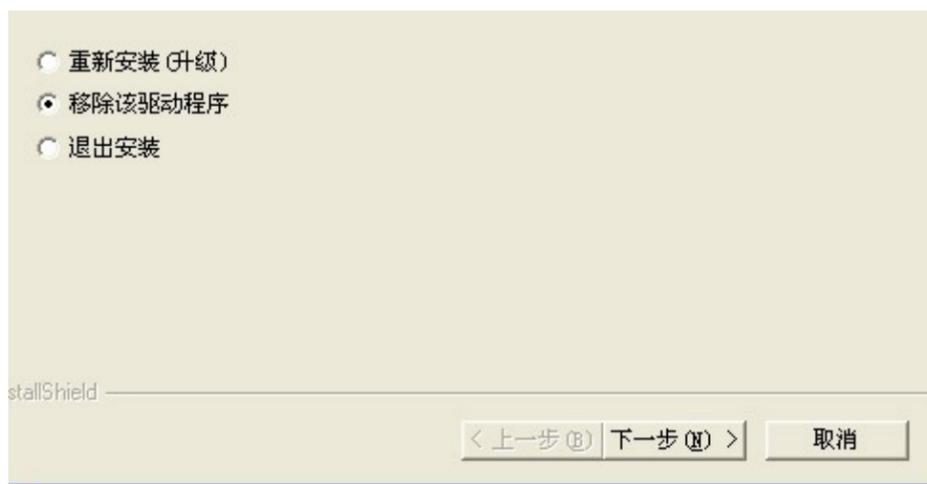
再次双击仿真器 XP 文件夹里面的安装图标



:显示如下



则先要选择移除该驱动程序,如下

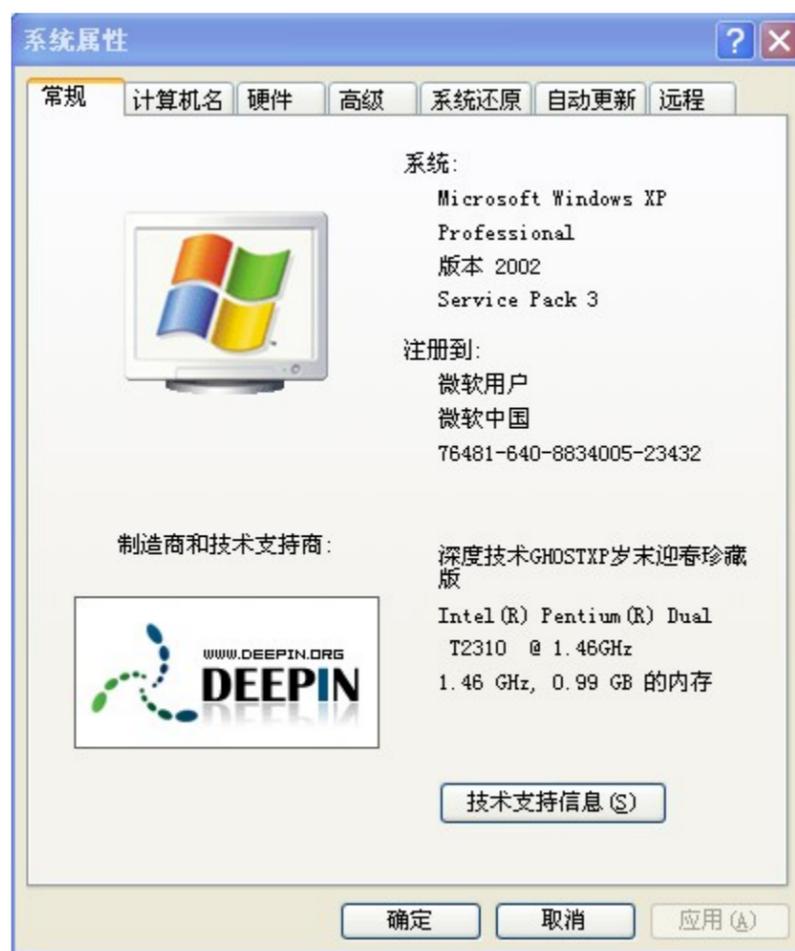


再点击下一步

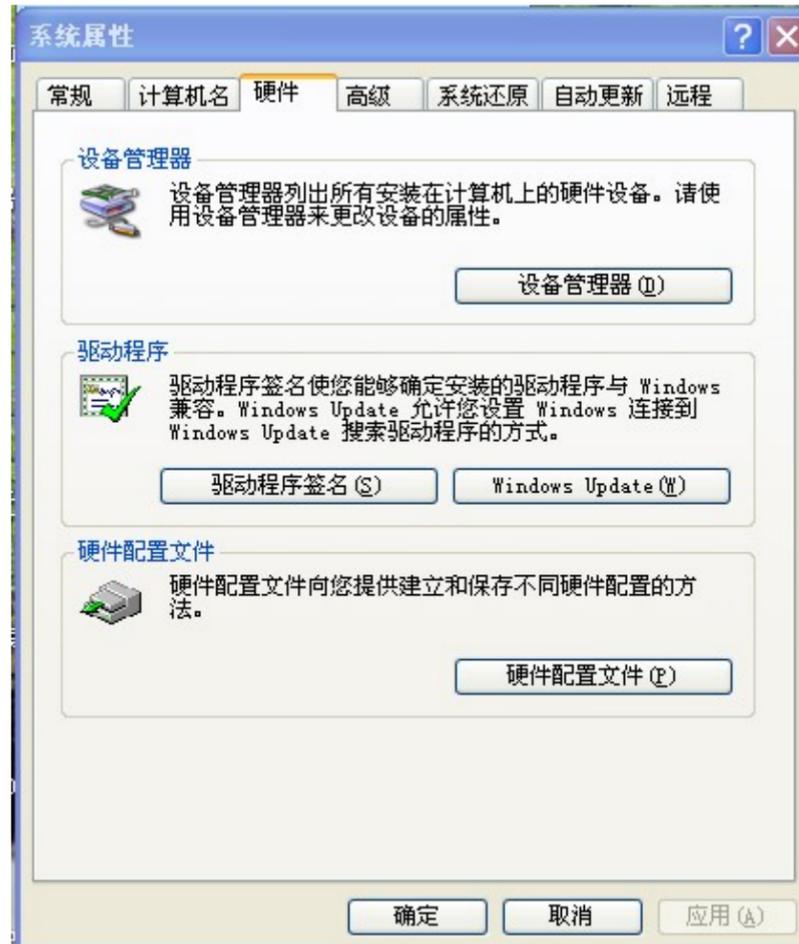


再选择立即重新启动电脑, 再点击完成, 重新启动电脑再按上面安装 XP 驱动方法安装驱动。

成功安装完仿真器驱动后, 插上仿真器, 则电脑会在设备管理器里面有一个 COM 口显示, 请打开电脑的设备管理器查看, 首先点击桌面我的电脑-属性:



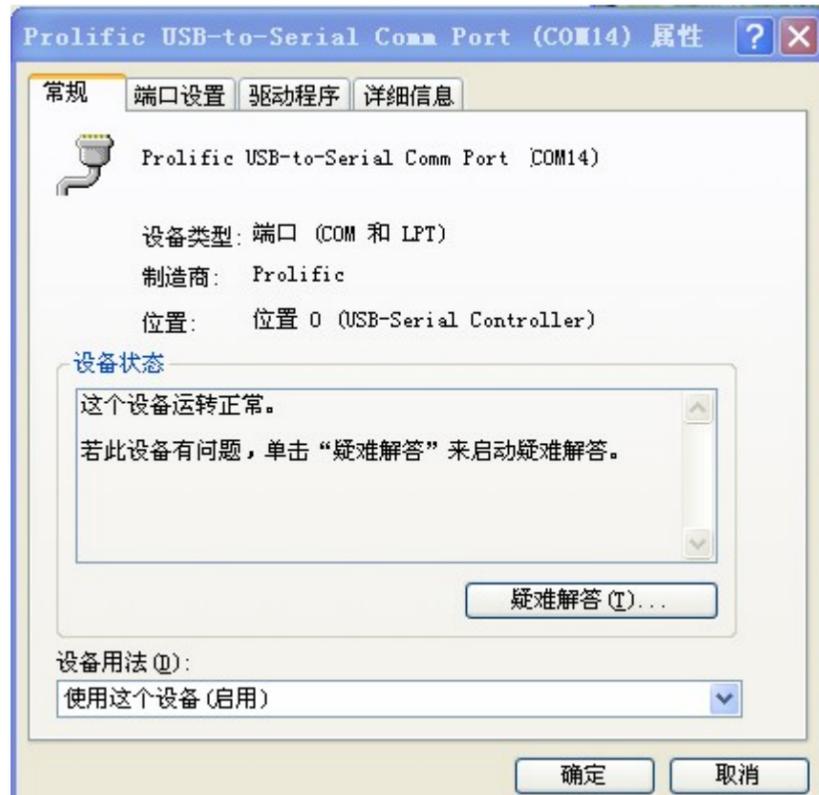
选择 硬件:



打开设备管理器, 显示如下



大家看到, 电脑上面显示了一个 COM14, 这就是插上仿真器安装好驱动后显示的端口, 但这个 COM14 端口号太大了, 要修改到 COM1~COM4 之间, 因为 AVR STUDIO 软件只识别 COM1~COM4, 大于 COM4 就不识别了, 所以在此我们要修改一下端口标号, 右键点击端 COM14 端口的属性, 显示如下:



点击端口设置, 如下图



选择高级, 如下图



点击 COM 端口号右边的向下键头, 选择 COM1~COM4 之间的端口, 一般会跳出一问你是否强行修改的窗口, 点击确认就可以了, 如下图



再点击确认.



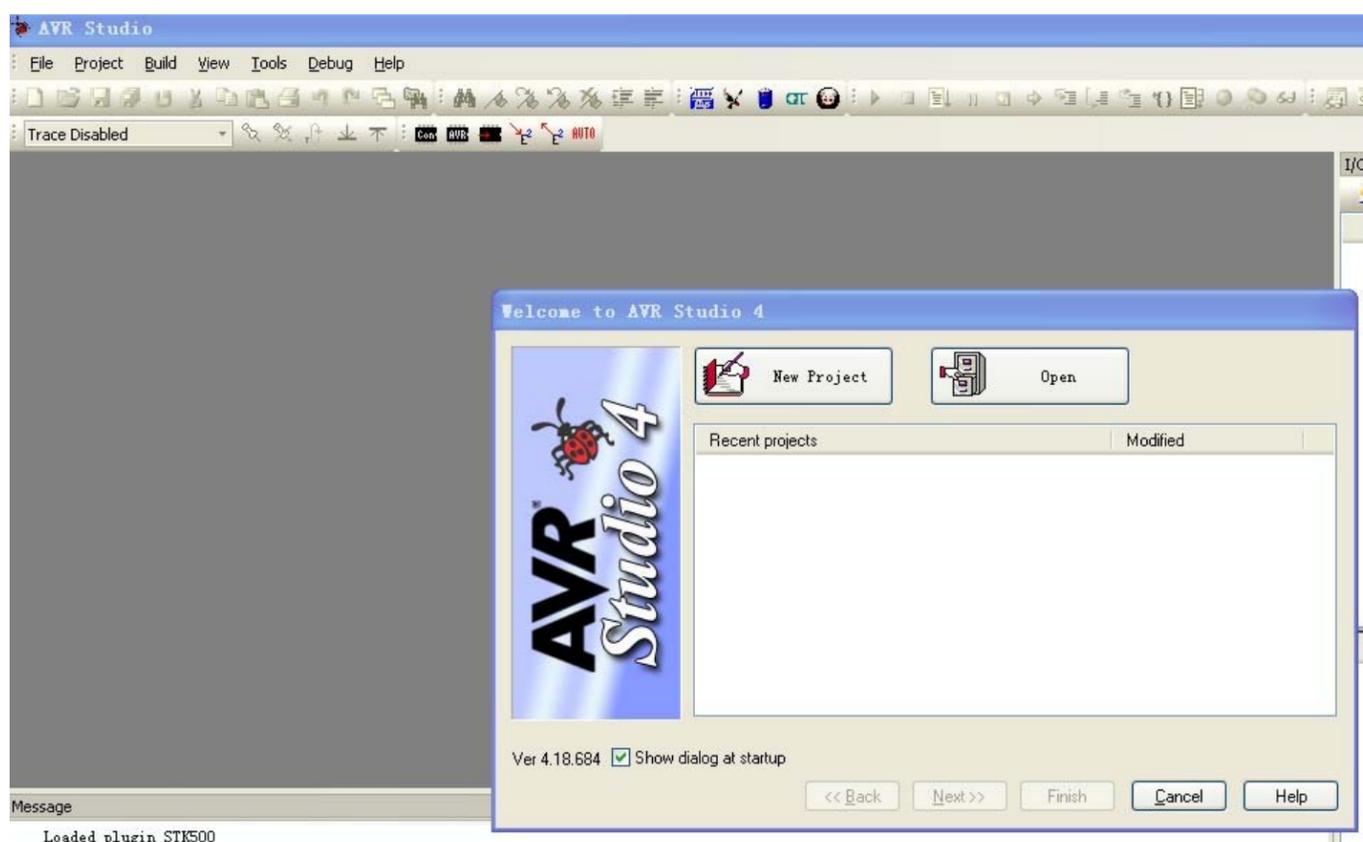
出现这个窗口时,再点击 是(Y),就可以了. 接下来就可以使用仿真器了.

仿真器的软件的使用:

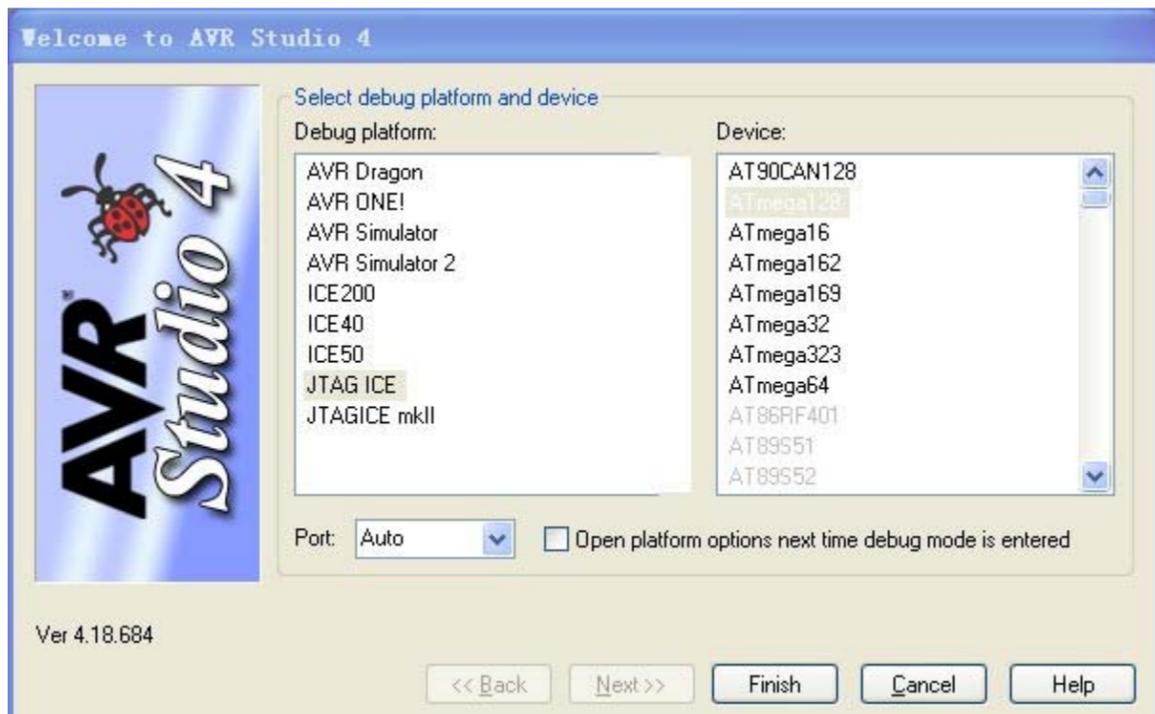
仿真器在使用时,不管是用来仿真还是用来下载,都得注意一个时间性的问题,大家在插上仿真器时,都可以看到,仿真器上面有两个灯亮着,一个恒亮(电源灯),一个不停的闪烁(数据灯),一个灯大概闪烁十秒钟后灭掉,大家在这个闪烁的灯闪烁的十秒钟里面是不能进行仿真和下载操作的,一定要等到这个闪烁的灯灭了再进行操作,千万要注意了,否则连不上(闪烁的十秒钟的时间里面是仿真器升级的时间不是操下载和仿真的时间.这个只有在仿真器升级时才会在这时间里面操作)

1, 首先介绍仿真器来进行仿真, 步骤如下:

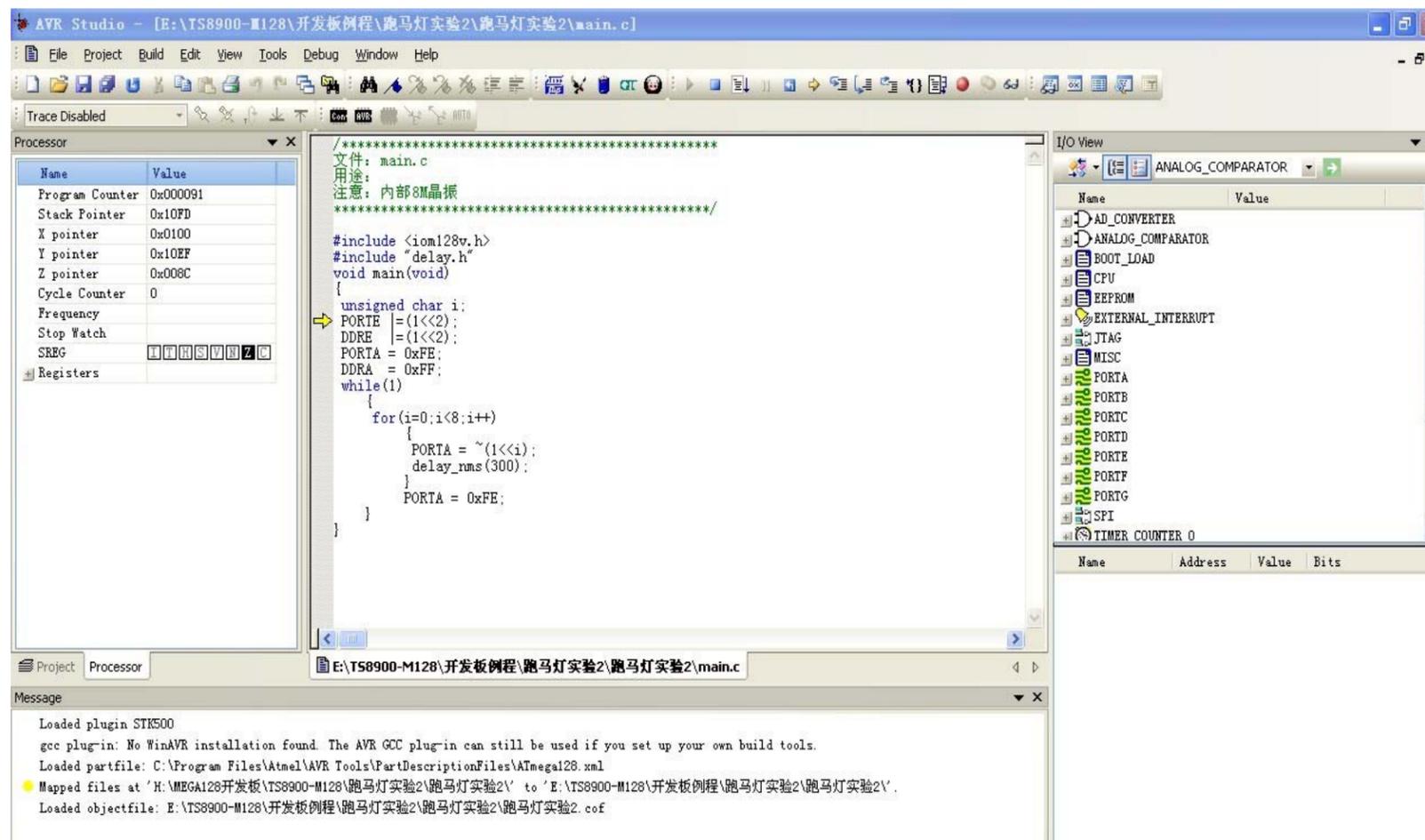
打开 AVR STUDIO 软件



选择打开, 点击 OPEN 选择 ICC 软件生成的. COF 文件, 一直点确认, 直到显示工具和芯片选择窗口, 如下图



选择 JTAG ICE 和 ATMEGA128, 再点击 Finish 完成. 接着下一步点击确认, 直到进入仿真调试界面, 如下:



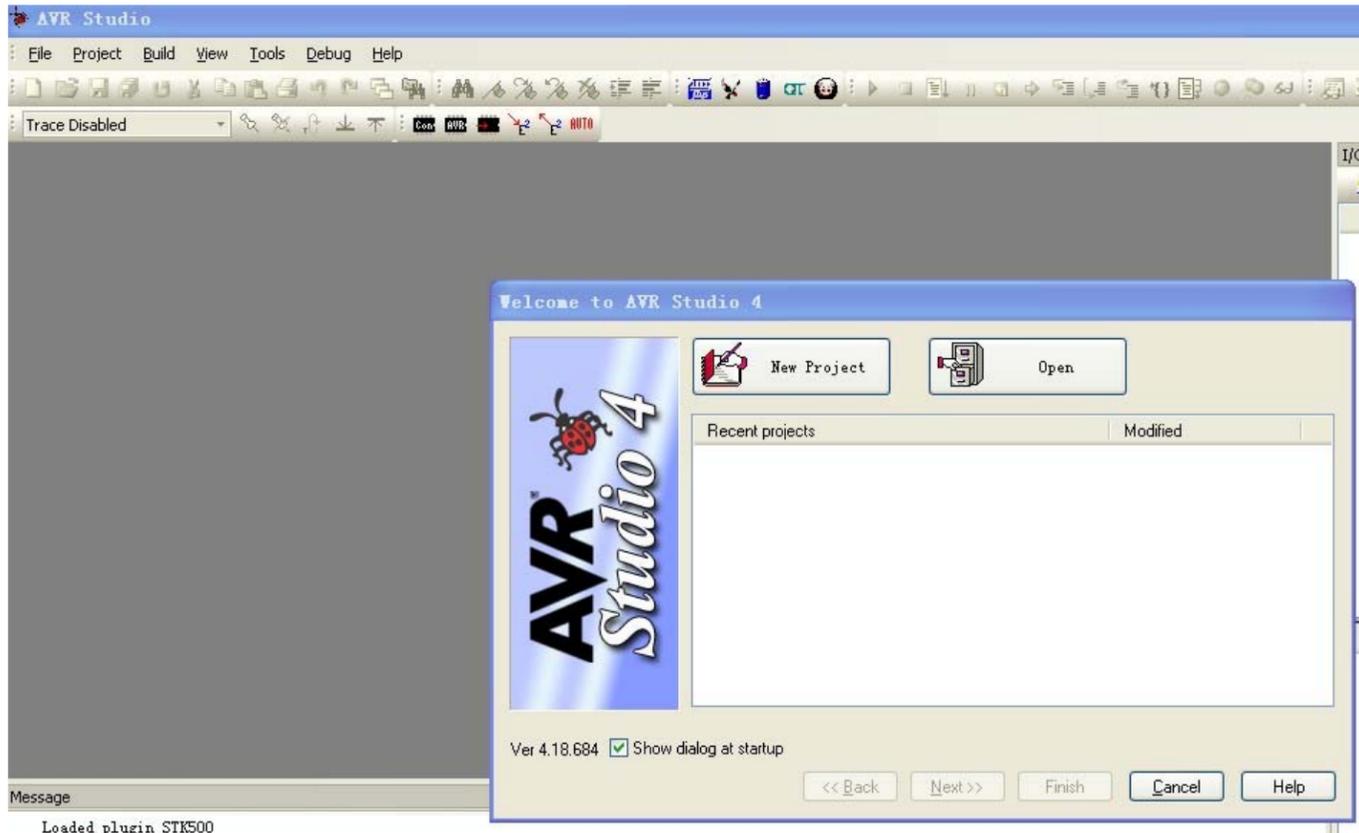
点击下面的图标即可进行单片调试, 设置断点等操作了, 如下图:



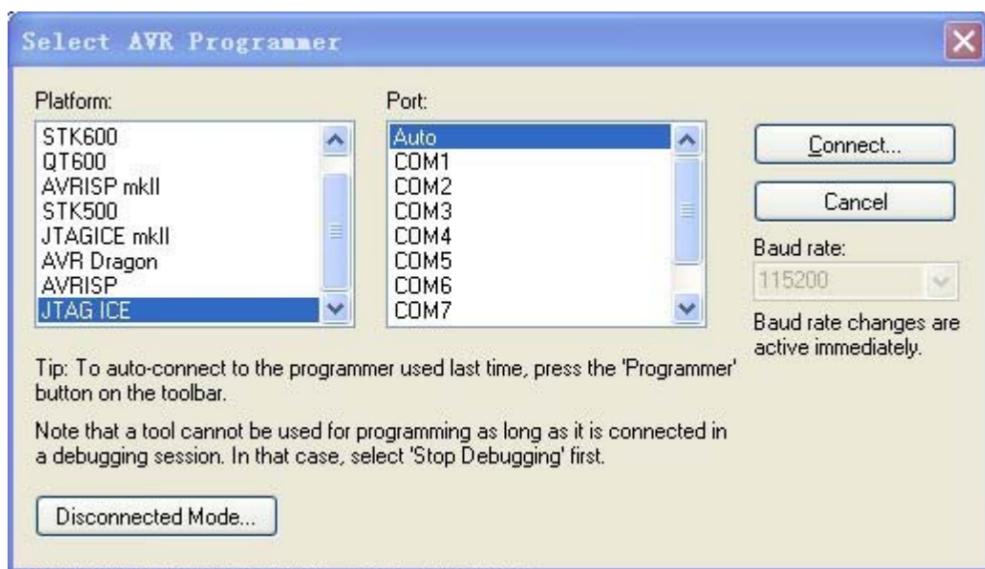
2. 仿真器的下载操作步骤:

首先如进行上面的仿真一样先打开 AVR STUDIO 软件, 再点击窗口下面的取消 “Cancel” 按键

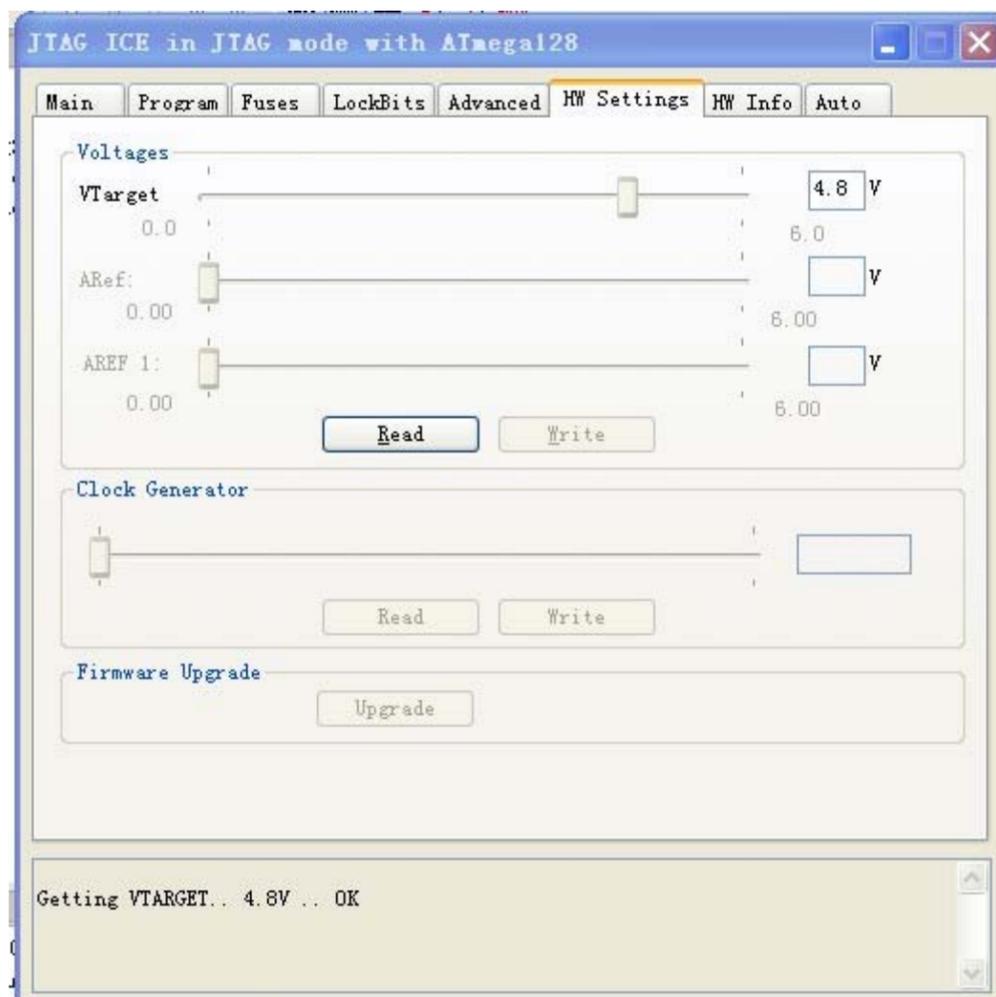
如下图:



再点击如下连接图标： 显示窗口如下图：



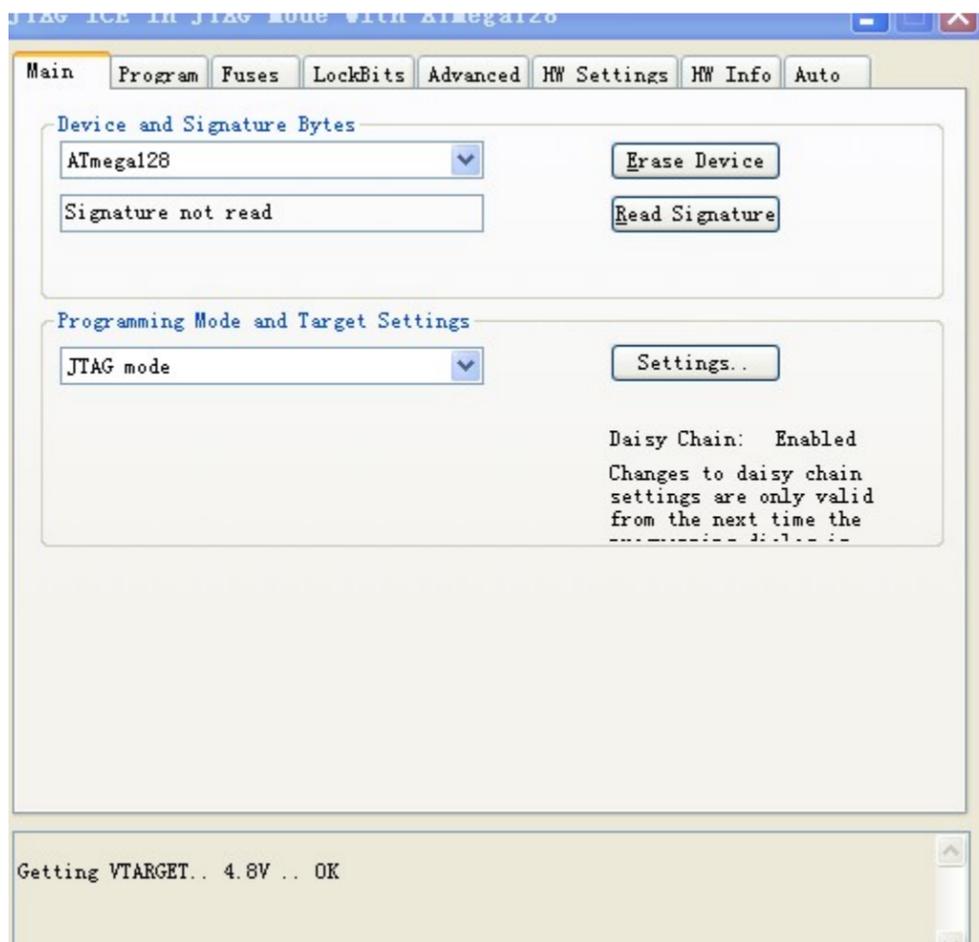
请选择 JTAG ICE 和 AUTO，再点击 Connect，接着就进入了下载界面，如下：



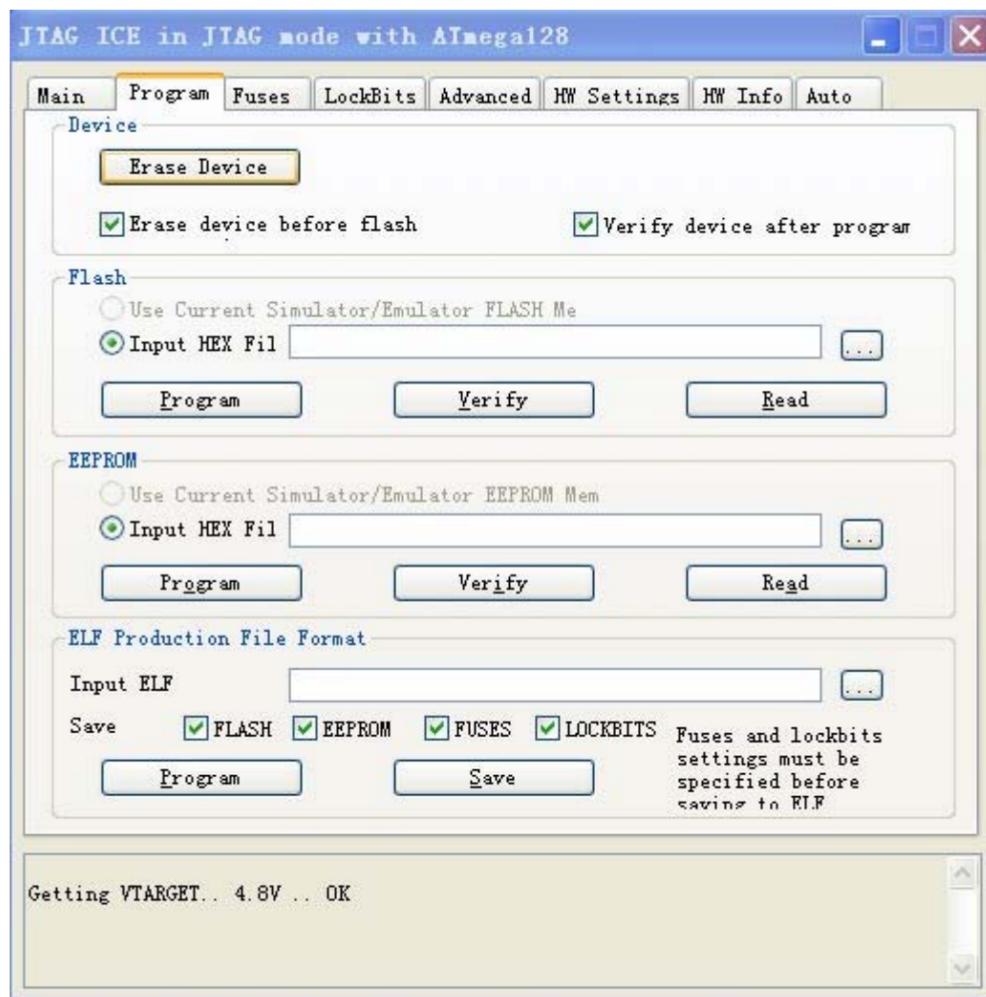
大家看到上面有一排选择不同的功能用的选项，如下图：



首先，选择 Main 窗口，选择好要下载芯片



再点击 Program



再从 Flash 那里加载要下载的程序



再点击 Program 则程序就下载进去了。

在这里强调一下，如果大家对 FUSE 也就是熔丝位不是很熟悉，建议大家暂时不要去修改，先学会了，等懂得了，了解除了 AVR，再去修改。我们的板在出厂前都是配置好了熔丝位的，内部 8M，我们的大多数例程都是根据这个熔丝位写的，少数除外。

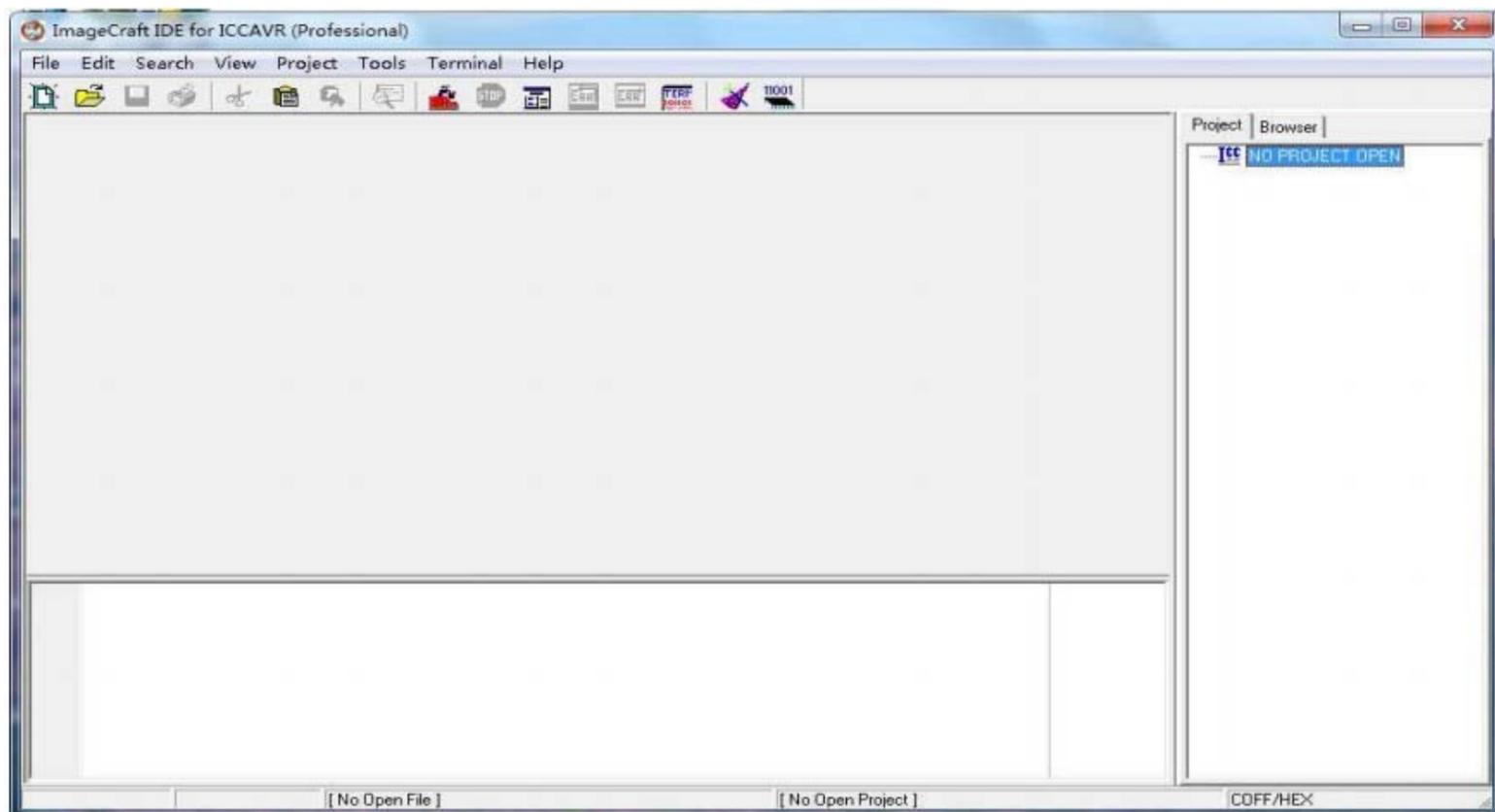
好了，上面讲了这么多，接下来就开始我们的第一个 AVR 程序的编写学习吧，

第七章 怎么样编写第一个 AVR 程序

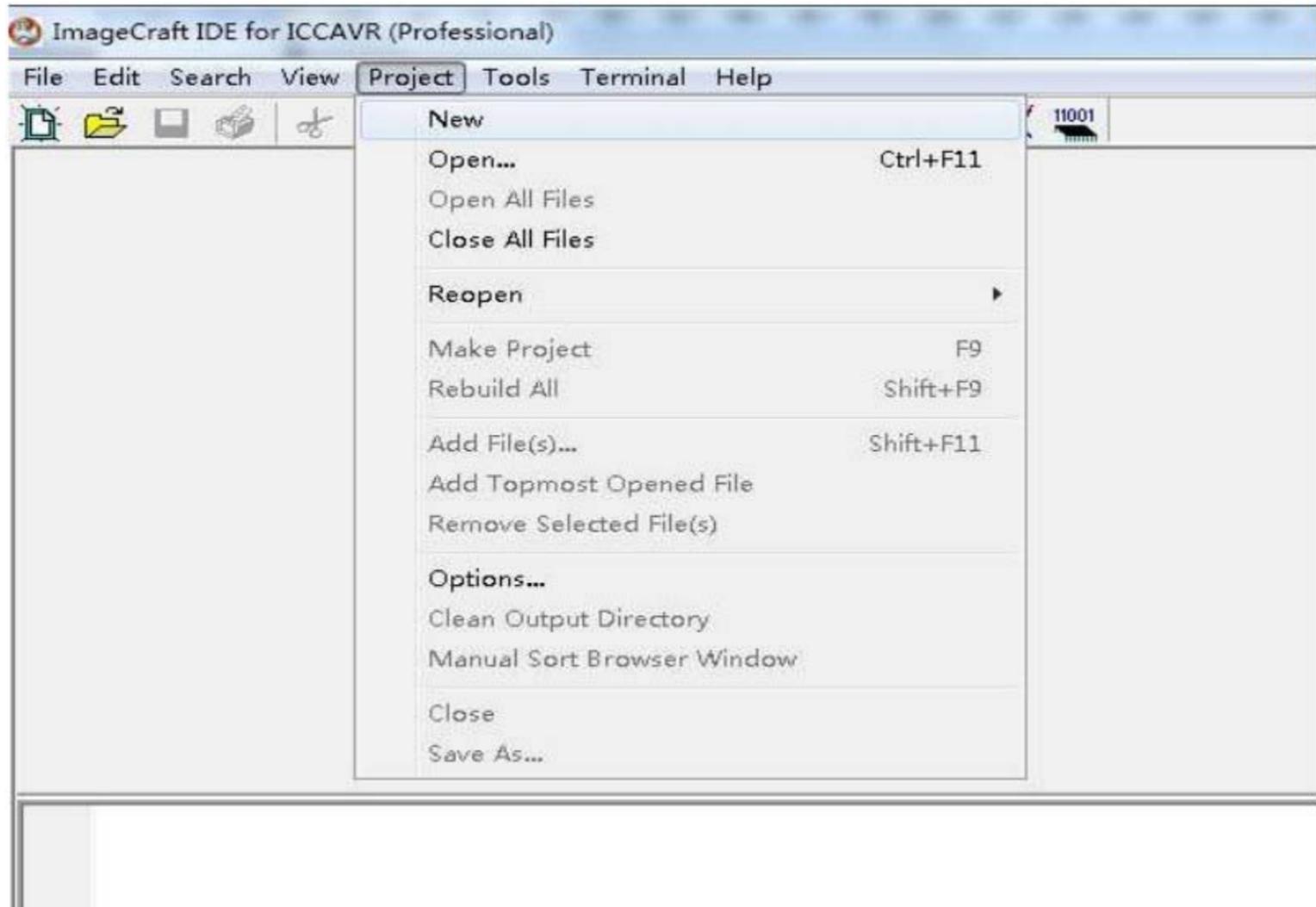
硬件驱动和软件安装配置完成后，我们便可以开始使用 TS8900-M128 来进行学习开发了。

下面我们以发光二极管闪烁灯的实现来说明整个开发使用流程，程序使用高级语言（C 语言），汇编语言的使用请用户自行学习和使用。

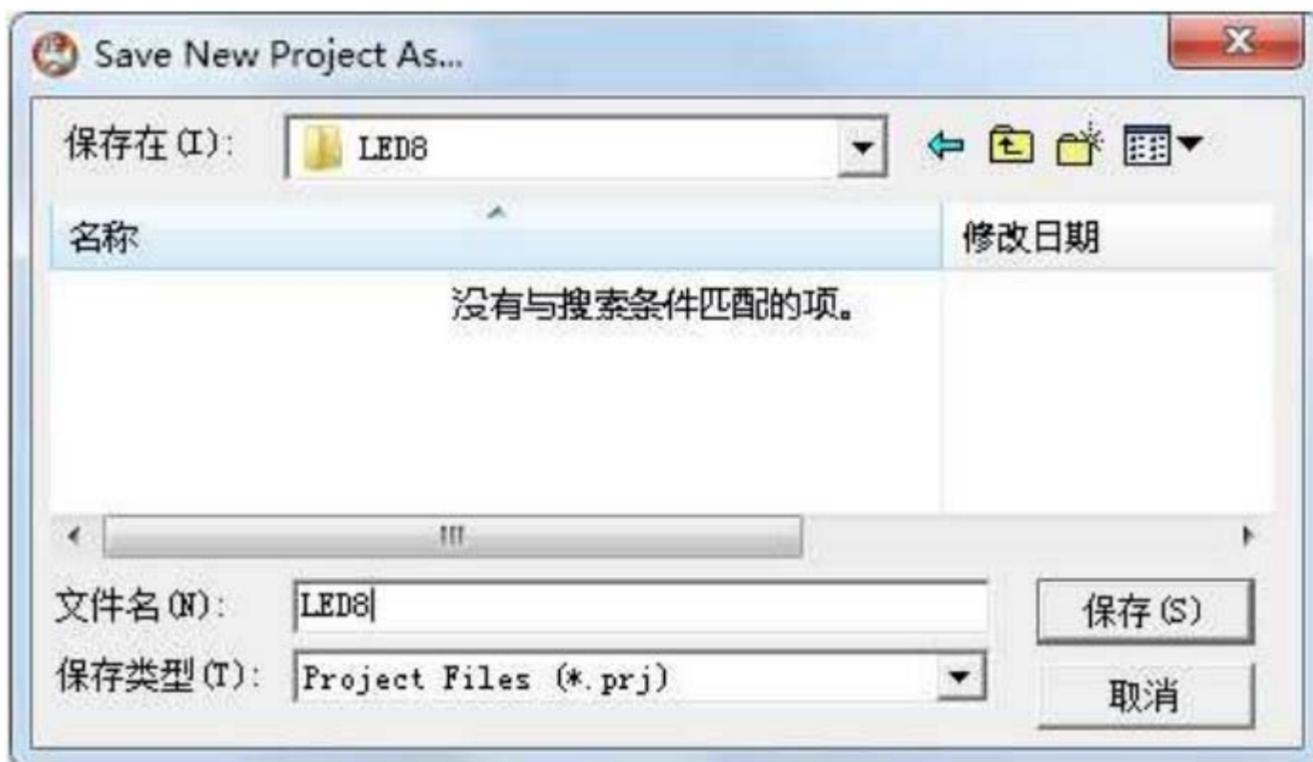
ICCAVR 编译环境下的使用项目建立程序的编写与编译将安装好的 ICCAVR 打开，如下图：



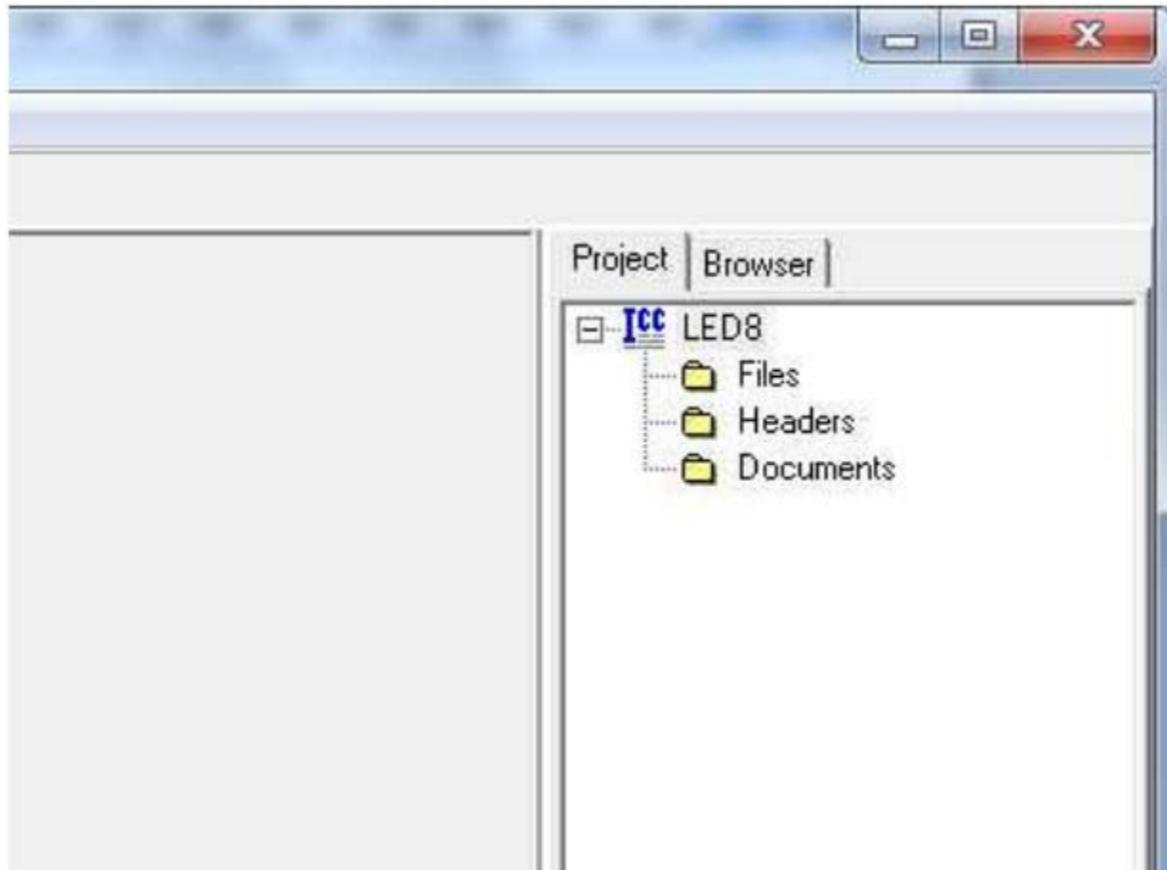
由上图可以看出，ICCAVR 的界面非常简洁，下面我们依然以 8 位数码管闪烁实验来开始建立项目。单击菜单栏的“Project”，选择“New”，如下图：



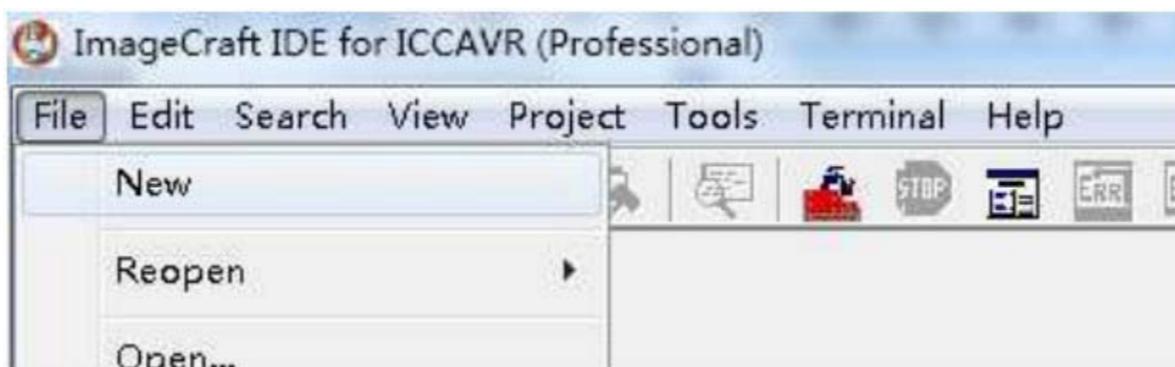
选择项目要保存的路径，ICCAVR 的项目路径可以根据用户自己设定，没有特别要求，这里项目命名为 LED8，如下图：



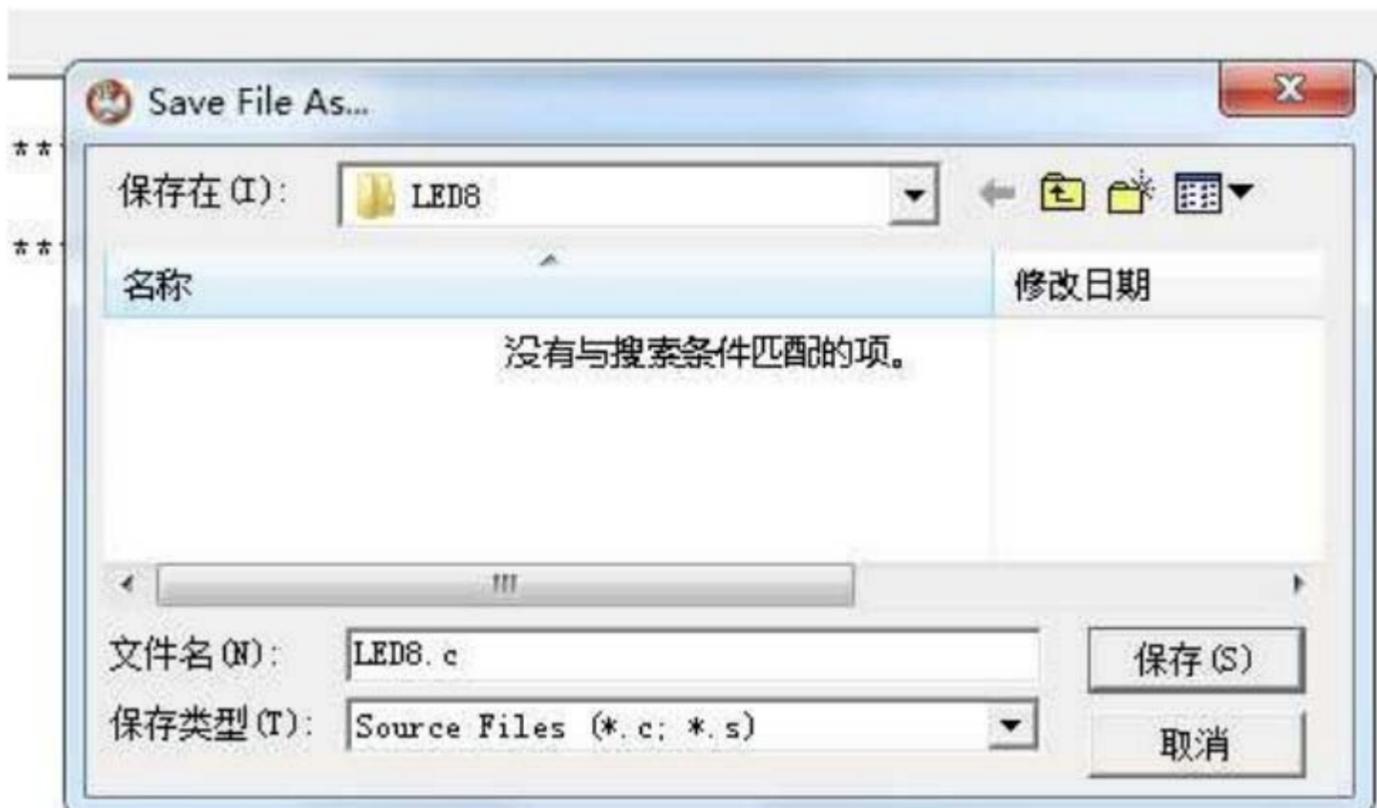
点击保存即可，这时发现右边出现了项目信息，如下图：



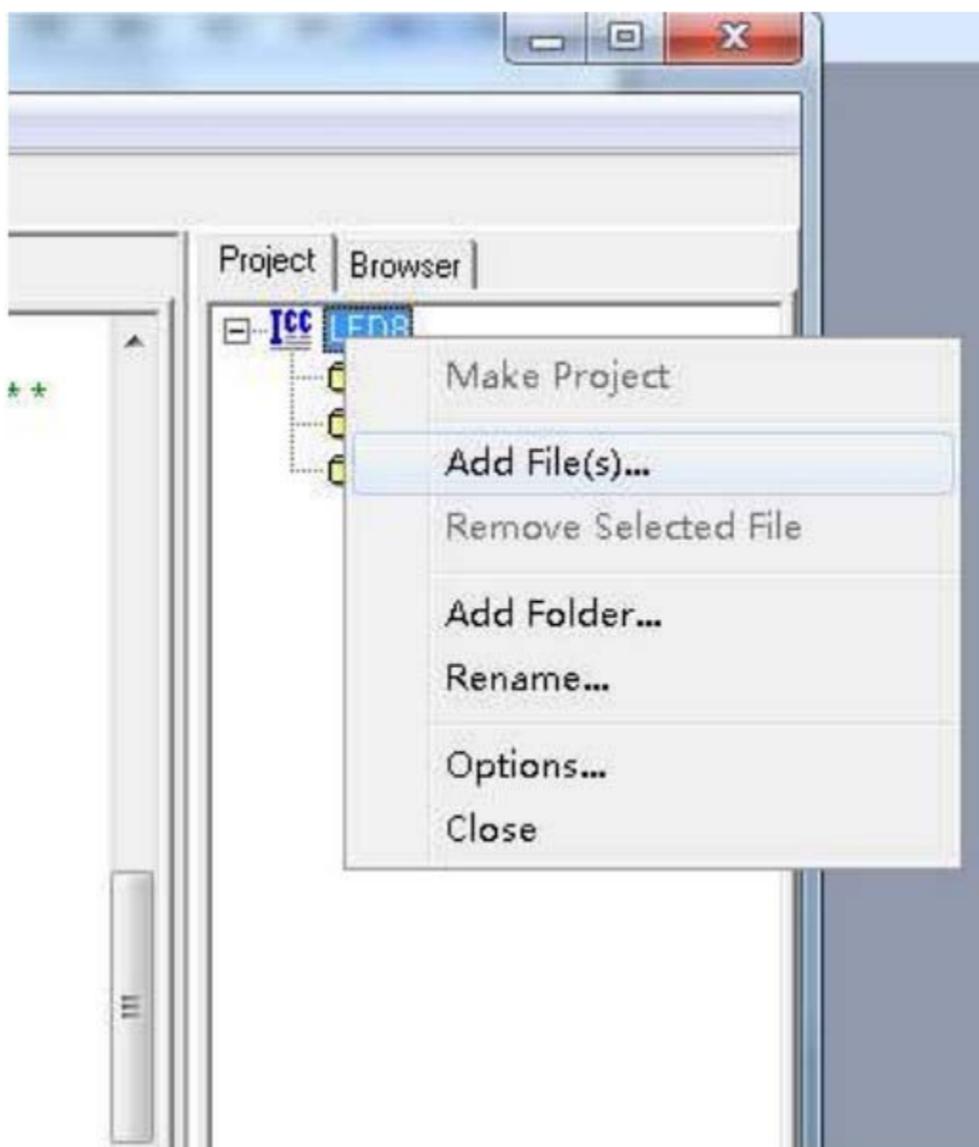
下面我们要建立源文件，点击菜单栏“File”下的“New”即可建立新文件，也可以用快捷键建立，如下图：



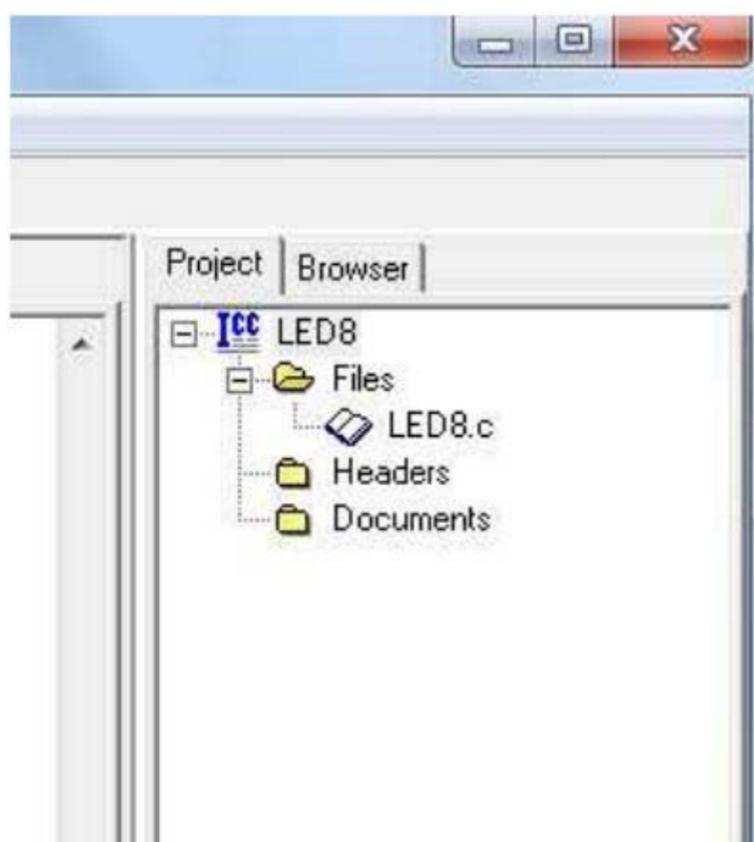
编辑源文件后，点击保存，命名为 LED8.c，如下图：



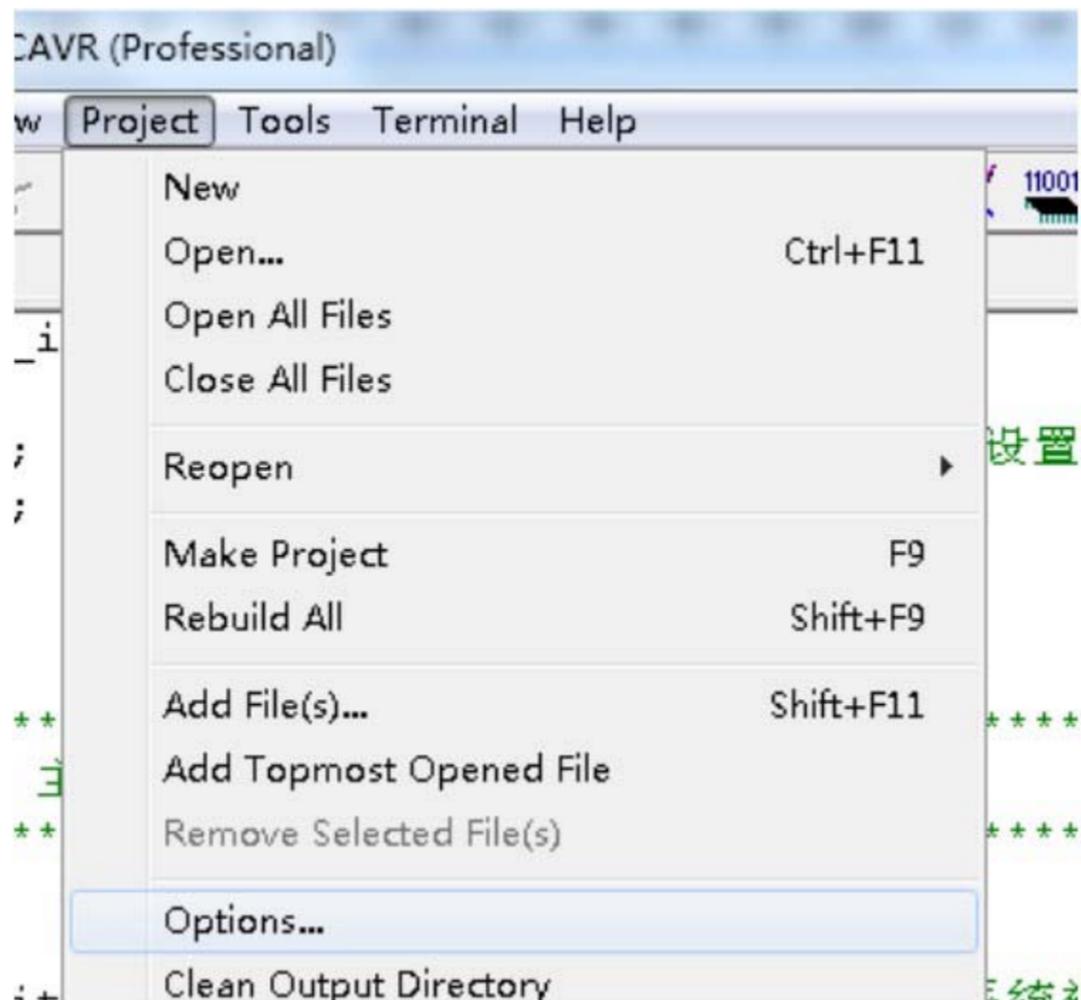
ICCAVR 的源文件必须手动加入到项目中，加入方法为：在右侧项目信息栏右键单击项目名，选择“Add File(s)”，选择刚才保存的项目文件即可添加进去。如下图：



添加后如下图：



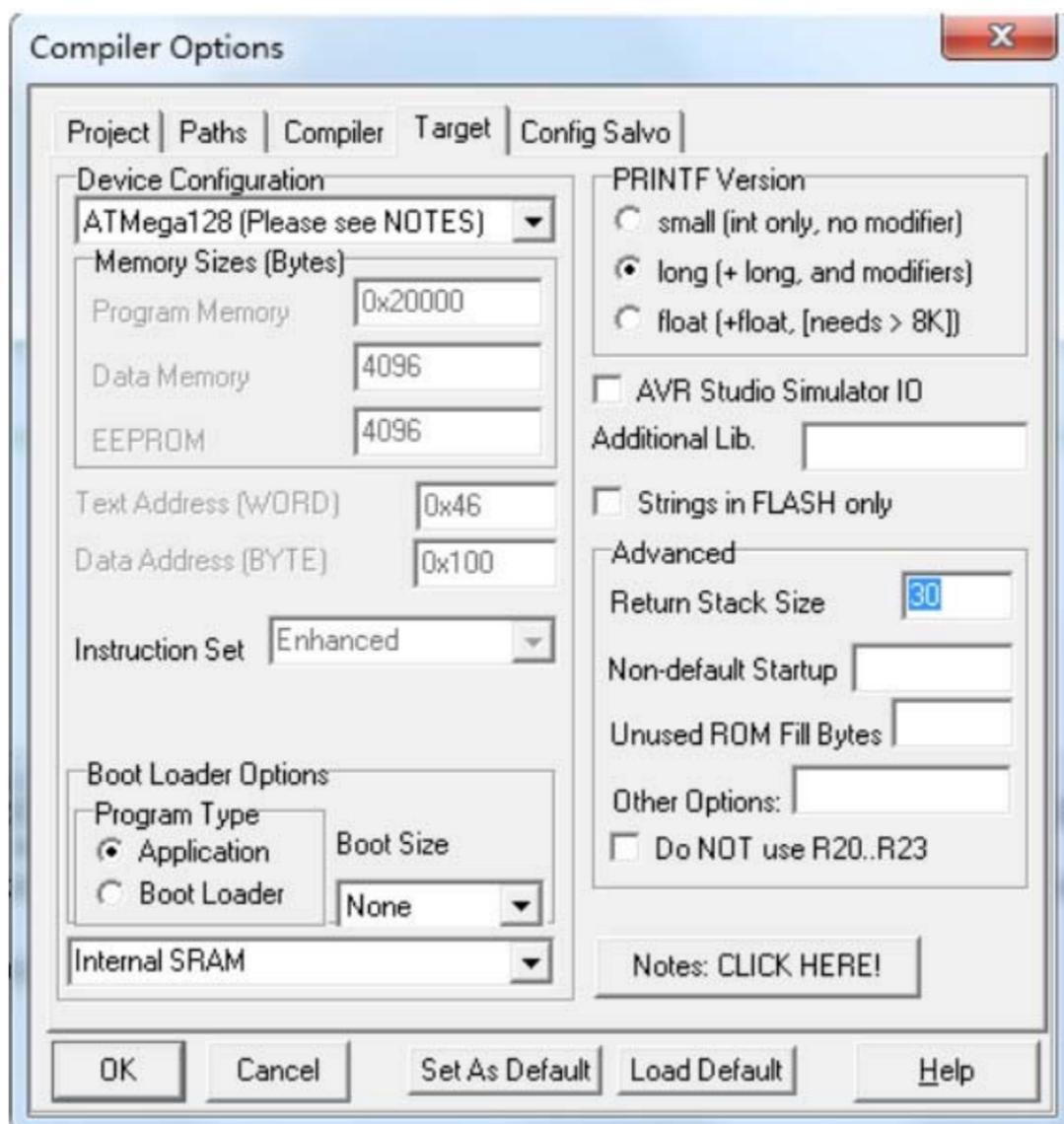
下面，我们对项目属性进行设置，包括芯片型号等。 点击“Project”下“Options...” ，如下图：



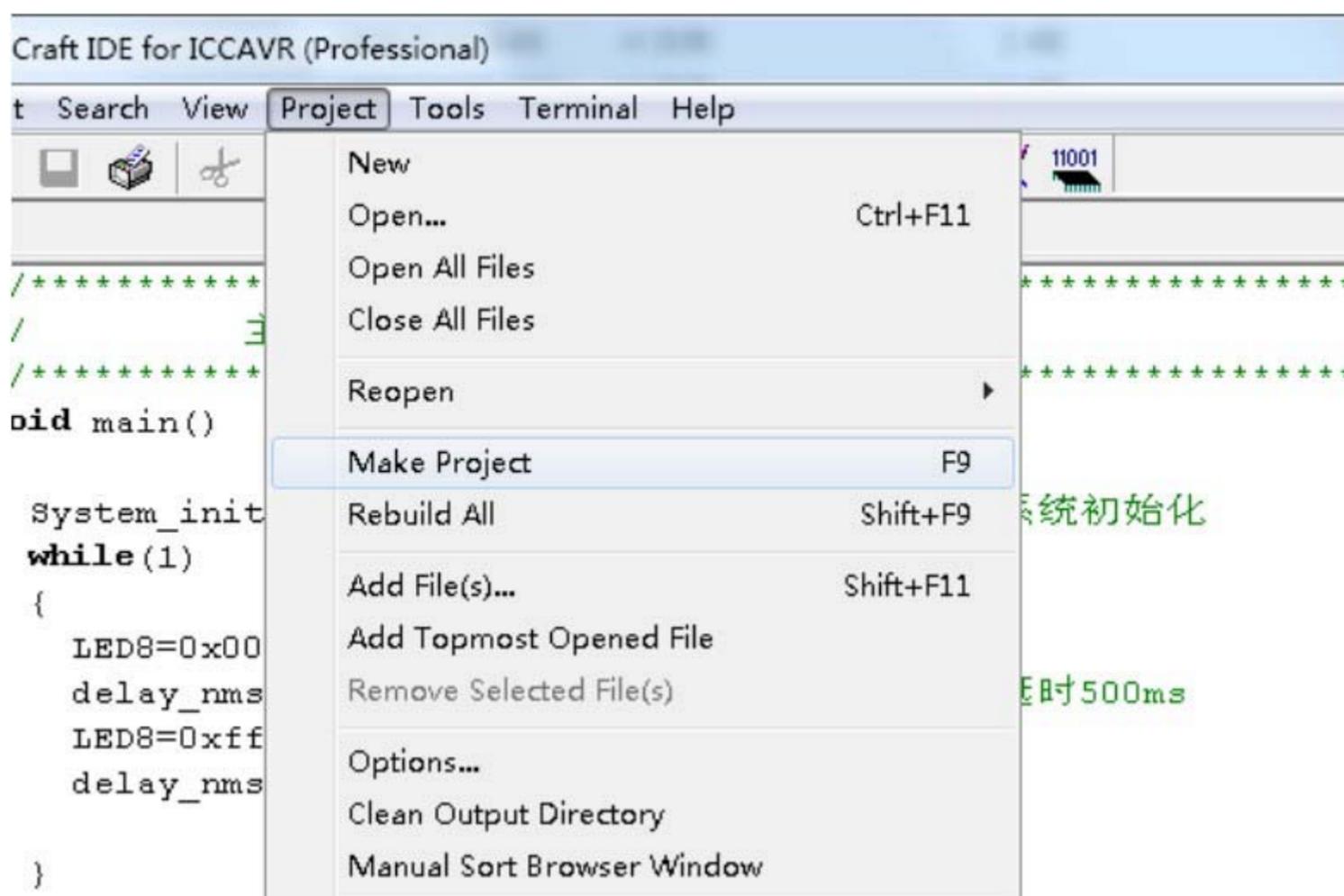
出现设置对话框，我们要对相关信息设置。这里要设置“Compiler”和 Target，由于在综合实验中，会使用单片机 Flash 来存储数据，这里我们中“Treat ‘const’ as ‘__flash’”选项，选中后的效果如下图：



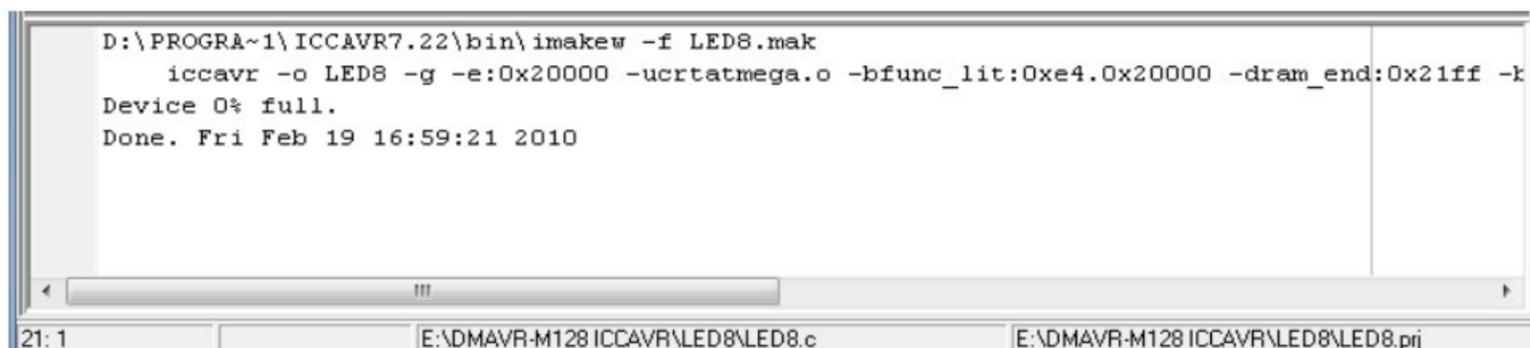
对 Target 进行设置，如下图：



点击 OK 即可完成设置。下面就可以对项目进行编译了，点击“Project”下“Make Project”或者直接按 F9，即可，如下图：



编译完成后，在下面信息栏会出现编译信息，是否有语法错误等，如下图：



```
D:\PROGRA~1\ICCAVR7.22\bin\imakew -f LED8.mak
iccavr -o LED8 -g -e:0x20000 -ucrtatmega.o -bfunc_lit:0xe4.0x20000 -dram_end:0x21ff -k
Device 0% full.
Done. Fri Feb 19 16:59:21 2010
```

至此，我们完成了我们的第一个实验在 ICC 下的编译操作，再利用 ASP 下载器或者或者 JTAG ICE 仿真器把 ICC 生成的 HEX 文件下载到开发板中就可以了，请参考 ISP 下载和 JTAG 下载使用说明，只要您认真学习和体会，定能快速掌握

第八章 开发板实验例程指导

一. 流水灯实验：

【实验目的】

1. 熟悉简单 IO 操作。
2. 理解位操作的概念和操作方法。
3. 学习查表法。

【实验设备】

- 1, PC 一台，带有 windows xp 或者更高级的操作系统
- 2, TS8900-M128 一块和下载器或者仿真器一个；
- 3, AVRStudio 集成开发环境
- 4, ICC 编译器

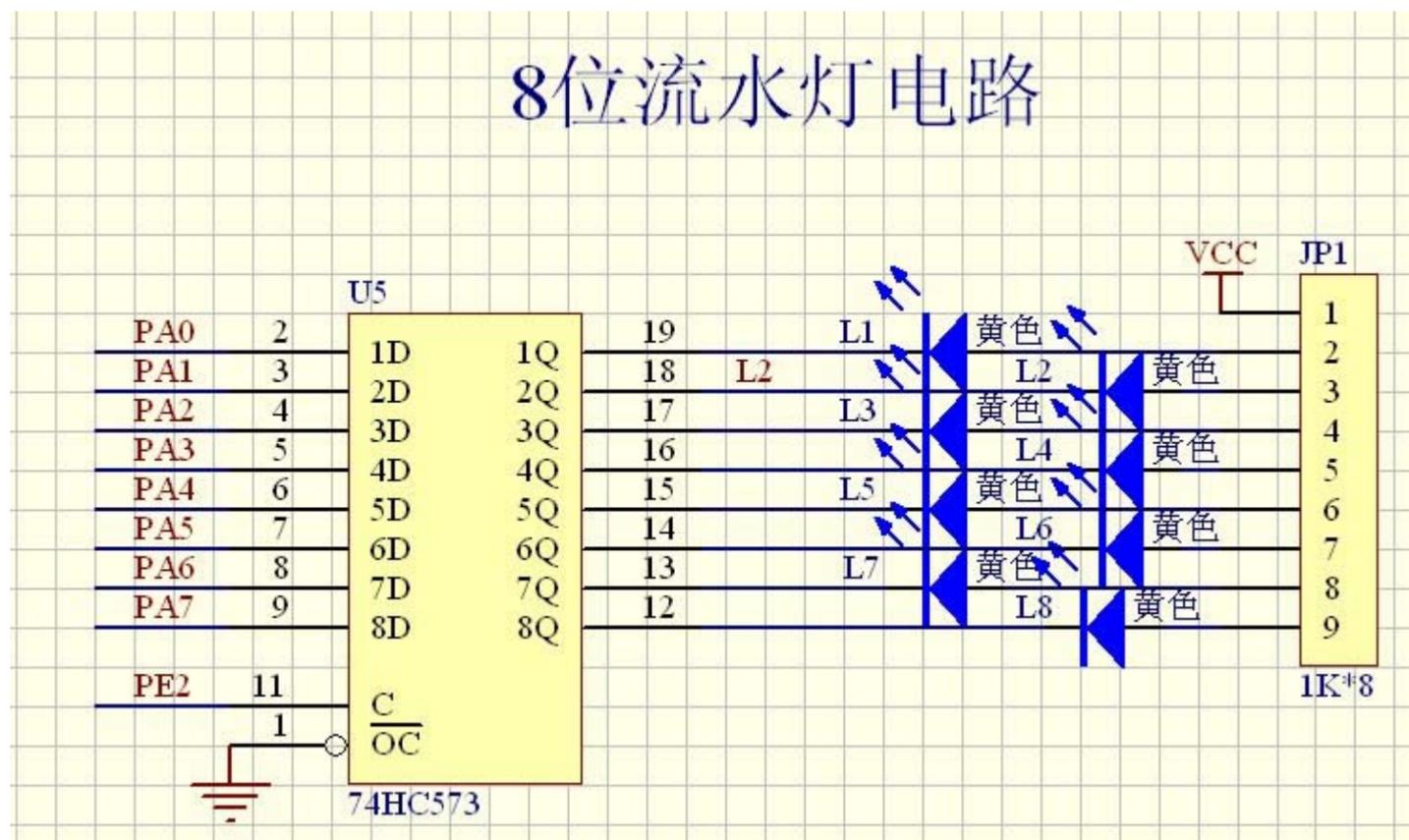
【实验要求】

通过简单的 IO 操作和延时控制，实现跑马灯现象。

【实验原理】

查看 M128 开发板电路原理图，了解 LED 的电路连接，如下图所示。图中 PA 口是先连到锁存器（74HC573），再通

过锁存器来控制灯的亮灭, 其中 JP1 是一个 8 个并排的 1K 的电阻, 用来给发光管限流的。



电路原理讲解: 例如 PA 口输出一个 8 位数据,

当 PE2 为高电平时 ($PORTE = (1 \ll 2)$), 则锁存器输出和 PA 口一样, 输入和输出相当直通, PA 变化时, LED 也会跟着变化;

当 PE2 控制脚为低电平时 ($PORTE \& \sim (1 \ll 2)$), 则输入和输出断开, PA 口不管怎么变化, LED 都不会变化;

实验注意事项:

1. 本店仿真器或者下载器都可以直接给开发板供 5V 直流电, 在使用仿真器和下载器时, 不需要另外给开发板加 USB 电或者外接电源供电, 如果需要外接电源供电, 请拨下仿真器或者下载器上面的电源跳线帽, 再外加 USB 线供电或者外接电源 8~12V 供电, 以下操作都一样, 不再作说明, 请学习者注意;
2. 仿真器连到开发板, 仿真器不管是下载还是仿真只需要连到 JTAG 接口, 不许连到 ISP 接口, 不然有可能会烧板;
3. 下载器连到开发板, 下载器只能作下载程序用, 不能仿真, 请连到 ISP 下载接口, 不许连到 JTAG 接口, 不然有可能会烧板;

实验步骤:

请用下载或者调试工具连好开发板, 供好电了开发板电源灯 L9 会亮;

打开软件 ICC 软件编译修改程序;

打开 AVR STUDIO 用仿真器下载或者调试程序, 或者打开下载器专用下载软件下载; (请参考光盘里面的下载器软件文件夹)

二. 数码管显示实验

【实验目的】

熟悉 SPI 通信协议。

熟练掌握共阴数码管的结构, 学会使用。

学习利用 SPI 驱动 74HC595。

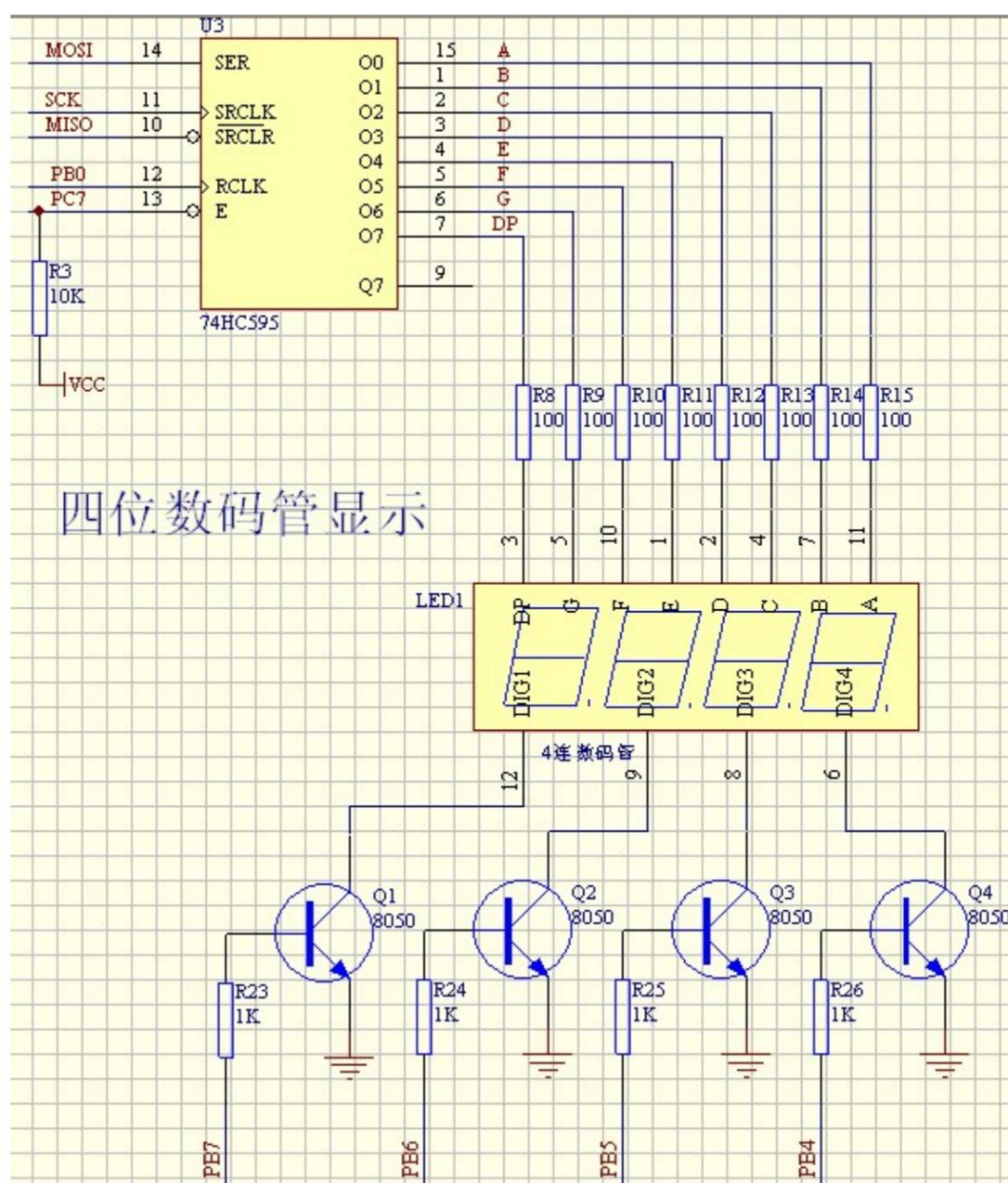
学会使用单片机定时器。

【实验设备】 略, 同上;

【实验要求】 利用定时器 T1 实现 1S 加 1 定时, 数码管显示秒数。

【实验原理】

查看 M128 开发板电路原理图, 如下图所示。



电路原理讲解: 单片机 SPI 口 (PB0, PB1, PB2, PB3) 通过控制 74HC595 芯片来给数码管送位码显示, 其中 R8~R15 是段限流电阻. 通过 PB 口 (PB4, PB5, PB6, PB7) 控制 NPN 三极管 (Q1, Q2, Q3, Q4) 来给数码管送段显示;

74HC595 芯片原理: 很多初学者认为数码管直接驱动起来好学一点, 在中间加了一个 595 芯片就感觉到吃力了, 所以在此我特别找了一些资料, 在此介绍一下 595, 让大家快速的掌握这个知识点.

74HC595 是具有 8 位移位寄存器和一个存储器, 三态输出功能的集成块。移位寄存器和存储器是分别的时钟。数据在 SCHcp 的上升沿输入, 在 STcp 的上升沿进入的存储寄存器中去。如果两个时钟连在一起, 则移位

寄存器总是比存储寄存器早一个脉冲。移位寄存器有一个串行移位输入 (Ds)，和一个串行输出 (Q7')，和一个异步的低电平复位，存储寄存器有一个并行 8 位的，具备三态的总线输出，当使能 OE 时 (为低电平)，存储寄存器的数据输出到总线。

74595 的数据端：

00—07：八位并行输出端，可以直接控制数码管的 8 个段。

Q7：级联输出端。我将它接下一个 595 的 SI 端。

SI：串行数据输入端。

74595 的控制端说明：

/SCLR(10 脚)：低点平时将移位寄存器的数据清零。通常我将它接 Vcc。

SCK(11 脚)：上升沿时数据寄存器的数据移位。QA→QB→QC→...→QH；下降沿移位寄存器数据不变。(脉冲宽度：5V 时，大于几十纳秒就行了。我通常都选微秒级)

RCLK(12 脚)：上升沿时移位寄存器的数据进入数据存储寄存器，下降沿时存储寄存器数据不变。通常我将 RCK 置为低点平，当移位结束后，在 RCK 端产生一个正脉冲 (5V 时，大于几十纳秒就行了。我通常都选微秒级)，更新显示数据。

/G(13 脚)：高电平时禁止输出 (高阻态)。如果单片机的引脚不紧张，用一个引脚控制它，可以方便地产生闪烁和熄灭效果。比通过数据端移位控制要省时省力。(这个引脚不管是否用到，都要给他下拉。否则显示会不稳定，万一上拉，将没有任何显示。)

程序说明：

每当 SCK 上升沿到来时，SI 引脚当前电平值在移位寄存器中左移一位，在下一个上升沿到来时移位寄存器中的所有位都会向左移一位，同时 Q7' 也会串行输出移位寄存器中高位的值，这样连续进行 8 次，就可以把数组中每一个数 (8 位的数) 送到移位寄存器；然后当 RCLK 上升沿到来时，移位寄存器的值将会被锁存到锁存器里，并从 Q1~7 引脚输出

附参考子程序：

约定：

CLOCK_L //存储寄存器的脉冲输入口 (低电平)

CLOCK_H (高电平)

DATA_L //串行数据输入端 (低电平)

DATA_H (高电平)

CLK_L //位移寄存器的脉冲输入口 (低电平)

CLK_H // (高电平)

/*595 串入并出*/

void Write_595(uchar data) //data 为需要串入的数据，为 1 个 8 位的二进制数

```

{
uchar j, filter;
filter = 0x01;//验证位
for(j=0; j<8; ++j)
{
if((data&filter)==0) //对每一位进行验证
DATA_L; //数据输入口拉低
else
DATA_H; //数据输入口拉高
CLOCK_H; //存储寄存器拉高
CLOCK_L;// 存储寄存器拉低, 脉冲改变, 数据存入寄存器
filter <<= 1;
}
}

```

并出的话给一个 CLK 的跳变就行了。

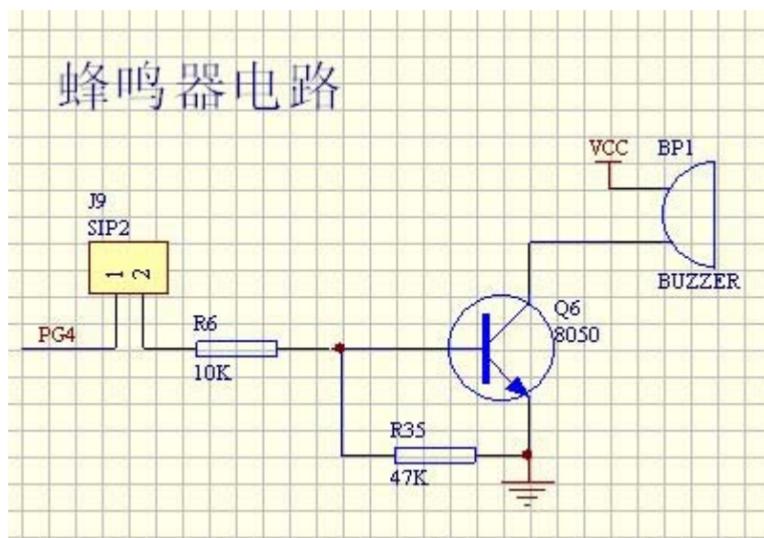
三. 蜂鸣器实验

【实验目的】 学习蜂鸣器的驱动方法；

【实验设备】 略，同上；

【实验要求】 蜂鸣器在按下开发板最下面一排按键时会响，不同按键响不同次数，同时数码管显示 1, 2, 3, 4。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。



电路原理讲解: J9 是一个跳线帽, 通过一个限流电阻 R6 控制 NPN 三极管来驱动蜂鸣器, 当 IO 口输出一个高电平时, 打开蜂鸣器, 蜂鸣器响, 当 IO 口输出一个低电平时, 蜂鸣器关掉。

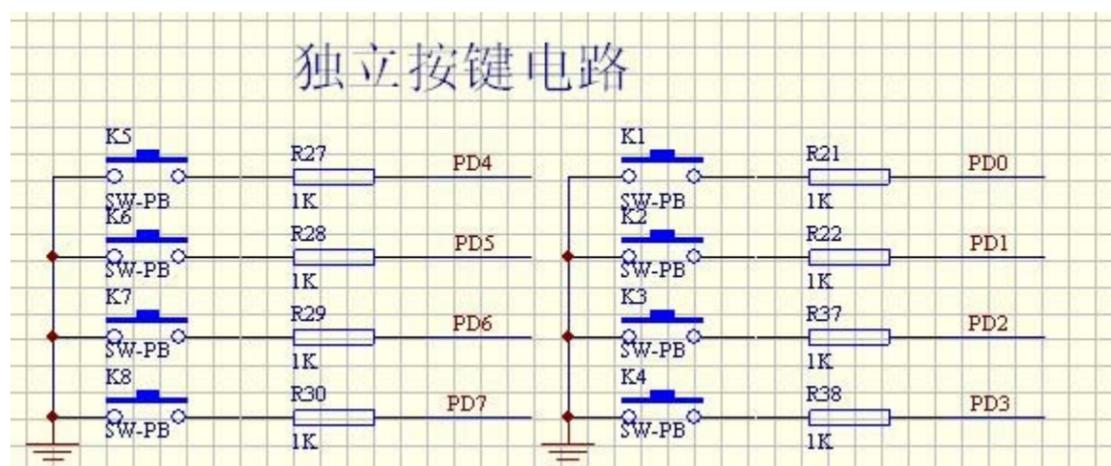
四. 按键实验

【实验目的】

【实验设备】 略，同上；

【实验要求】。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。



电路原理讲解：上图是 8 个独立按键，中间有一个限流电阻，防止强下拉；当按下按键时，IO 口会是一个低电平；

注意事项，在做按键实验时，首先要设置上拉 IO 口电阻，并设置为输入，例如：`PORTD=0XFF; DDRD=0X00;`

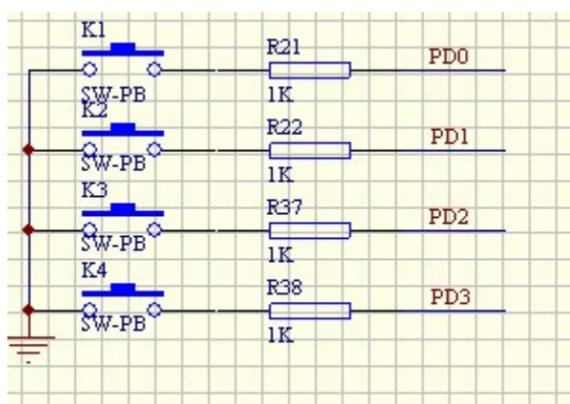
五. 中断实验

【实验目的】

【实验设备】 略，同上；

【实验要求】。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。



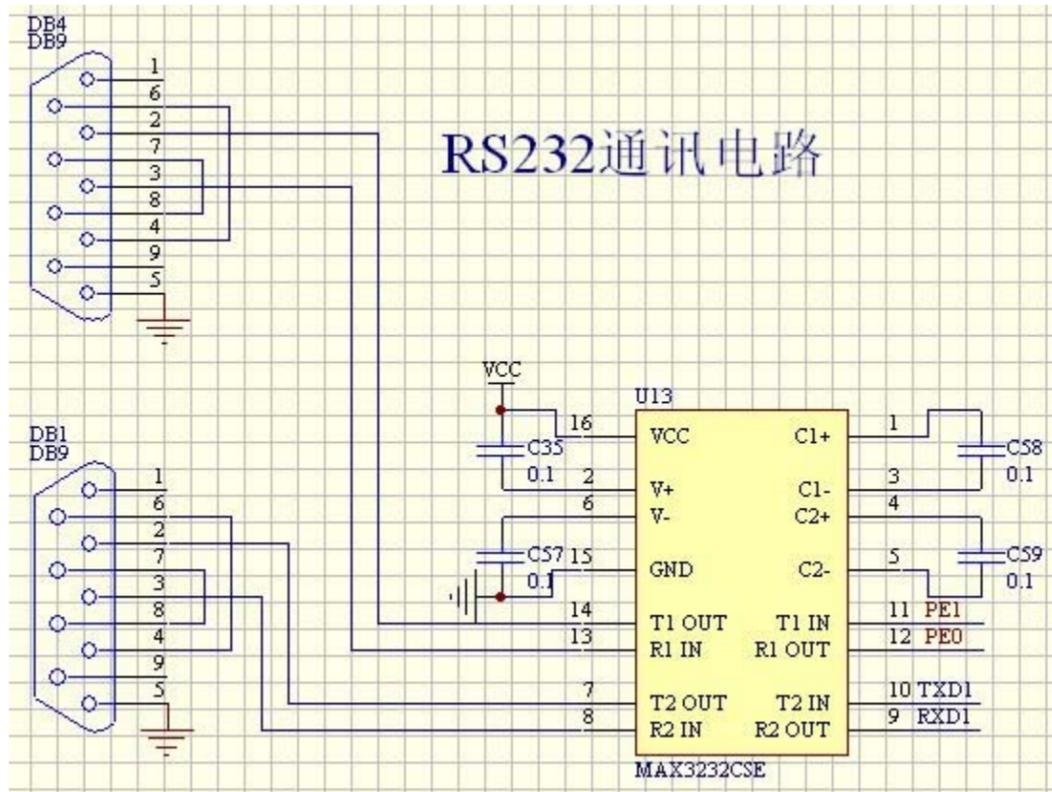
六. 串口通讯实验

【实验目的】

【实验设备】 略，同上；

【实验要求】。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。



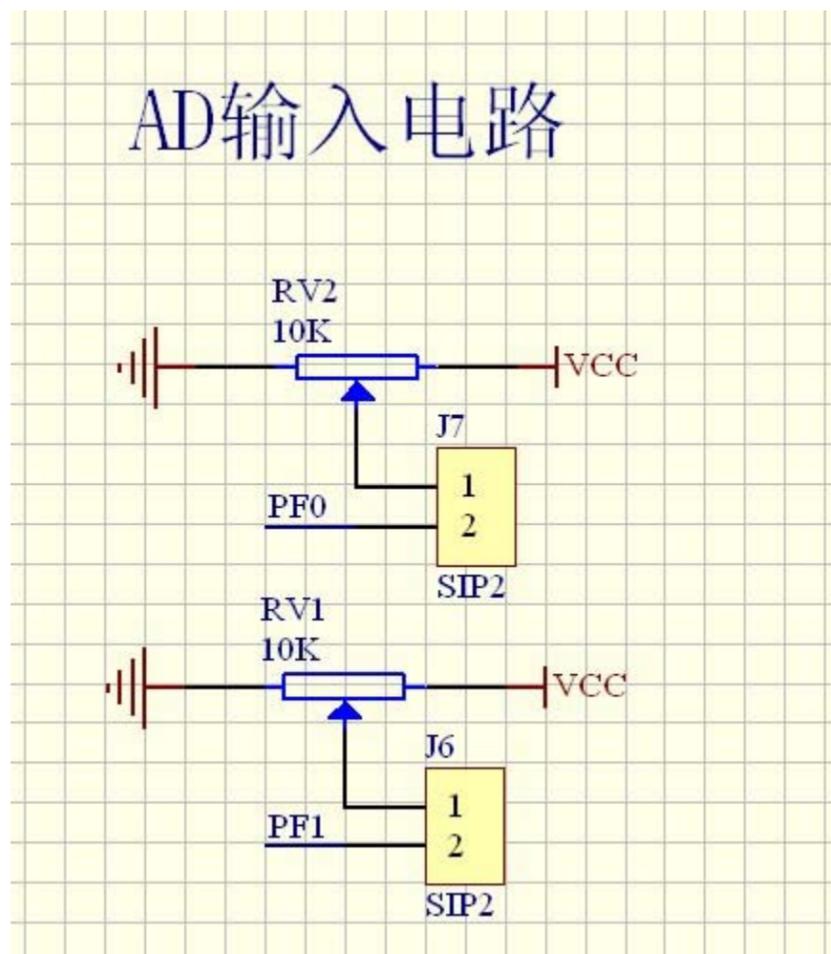
七. AD 转换实验

【实验目的】

【实验设备】 略，同上；

【实验要求】。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。



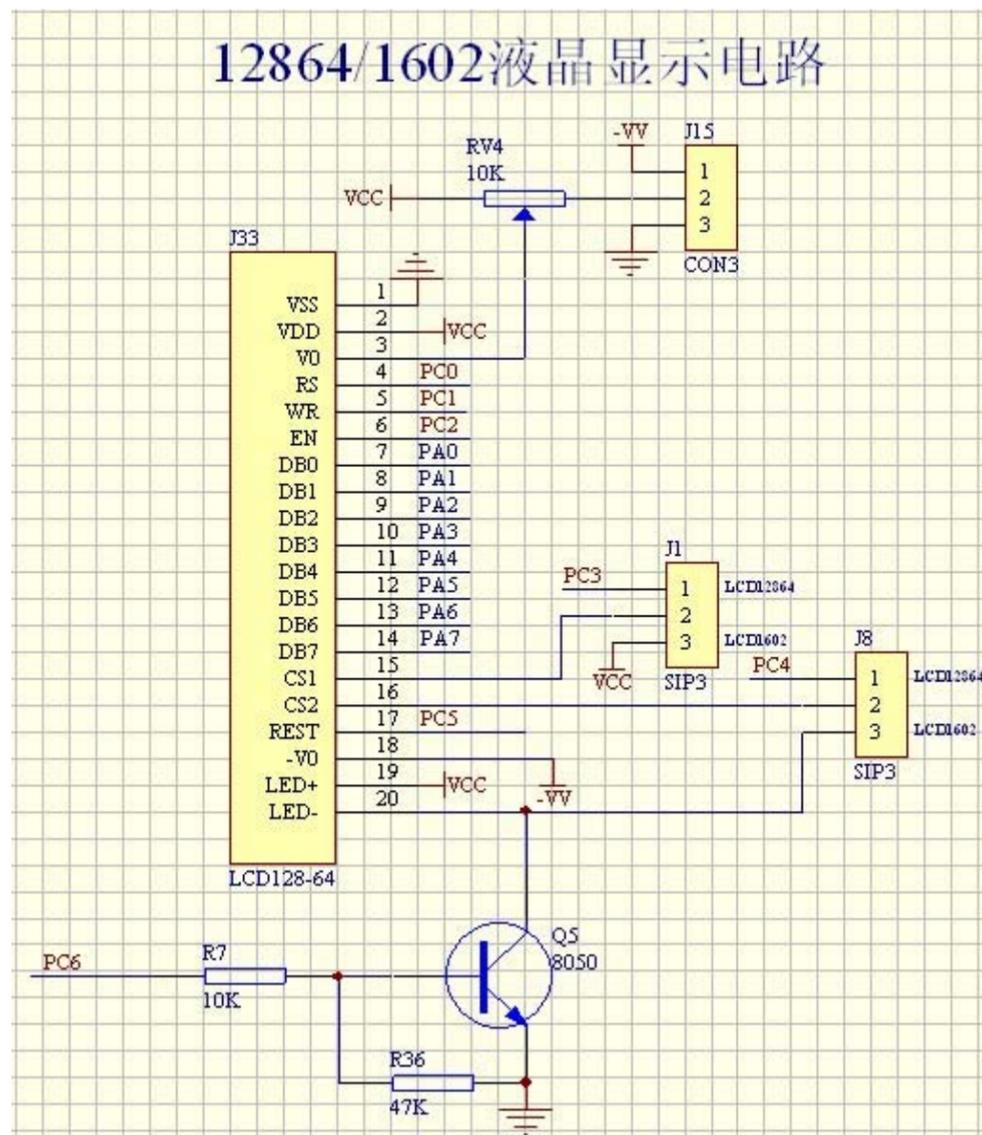
八. 1602 液晶显示实验

【实验目的】

【实验设备】 略，同上；

【实验要求】。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。



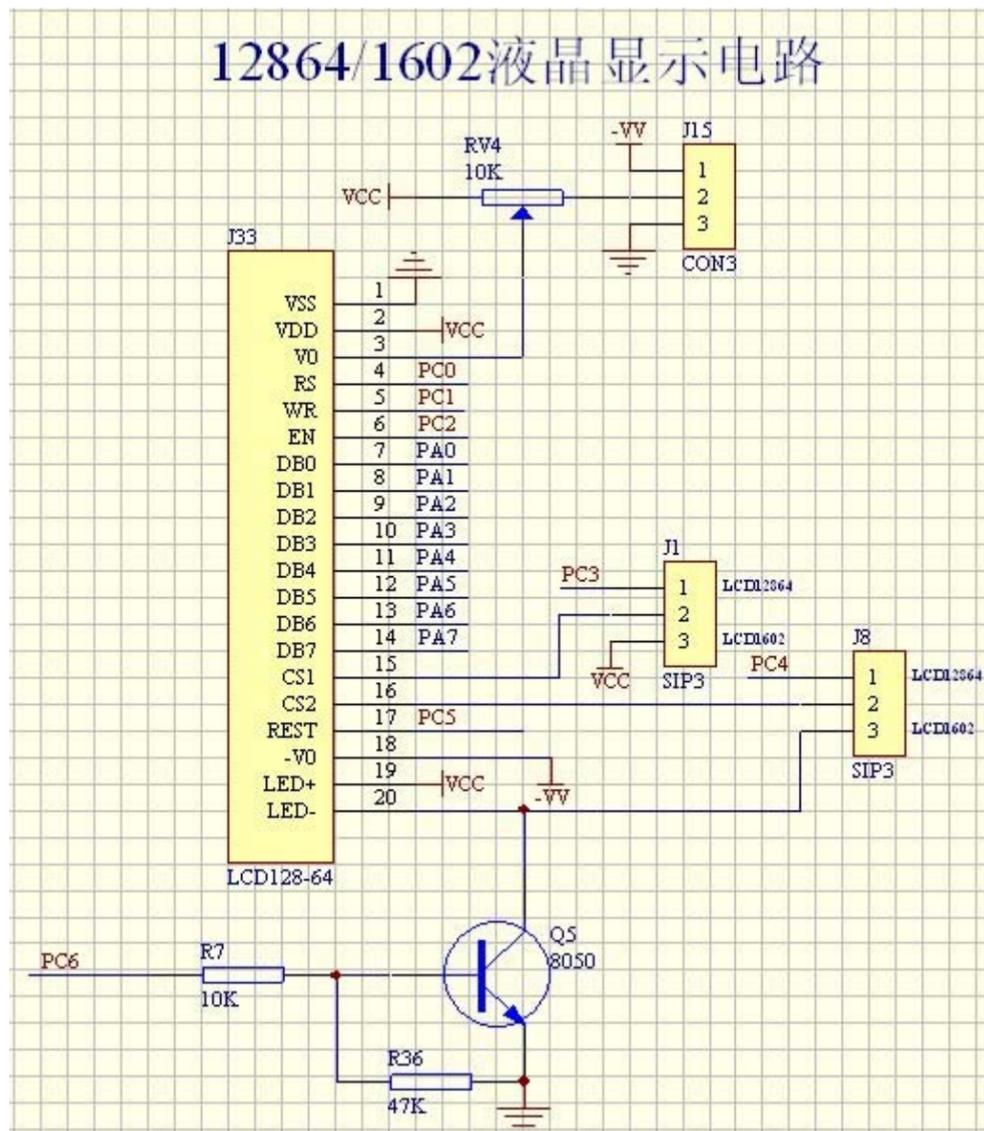
九. 12864 液晶显示实验

【实验目的】

【实验设备】 略，同上；

【实验要求】。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。



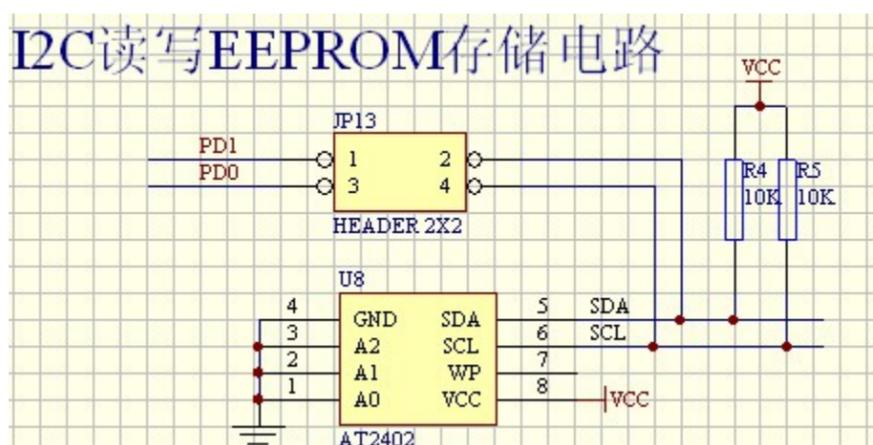
十. AT24C02 实验

【实验目的】

【实验设备】 略，同上；

【实验要求】。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。



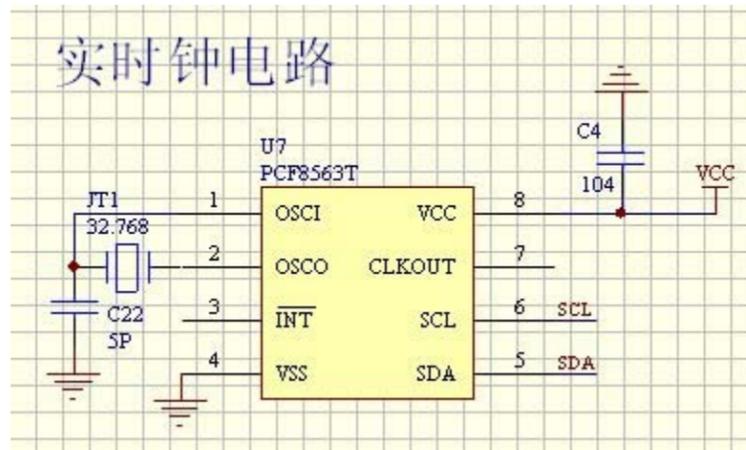
十一. PCF8563T 时钟显示实验

【实验目的】

【实验设备】 略，同上；

【实验要求】。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。



十二. PWM 实验

【实验目的】

【实验设备】 略，同上；

【实验要求】。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。

十三. 定时器实验

【实验目的】

【实验设备】 略，同上；

【实验要求】。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。

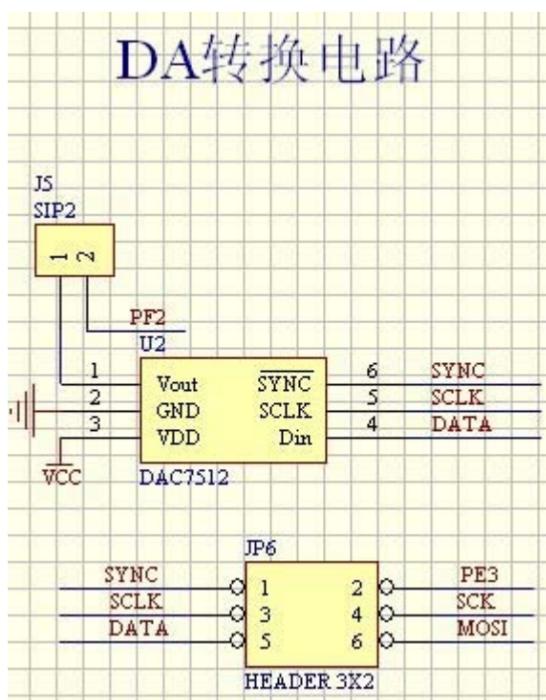
十四. DA 转换实验

【实验目的】

【实验设备】 略，同上；

【实验要求】。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。



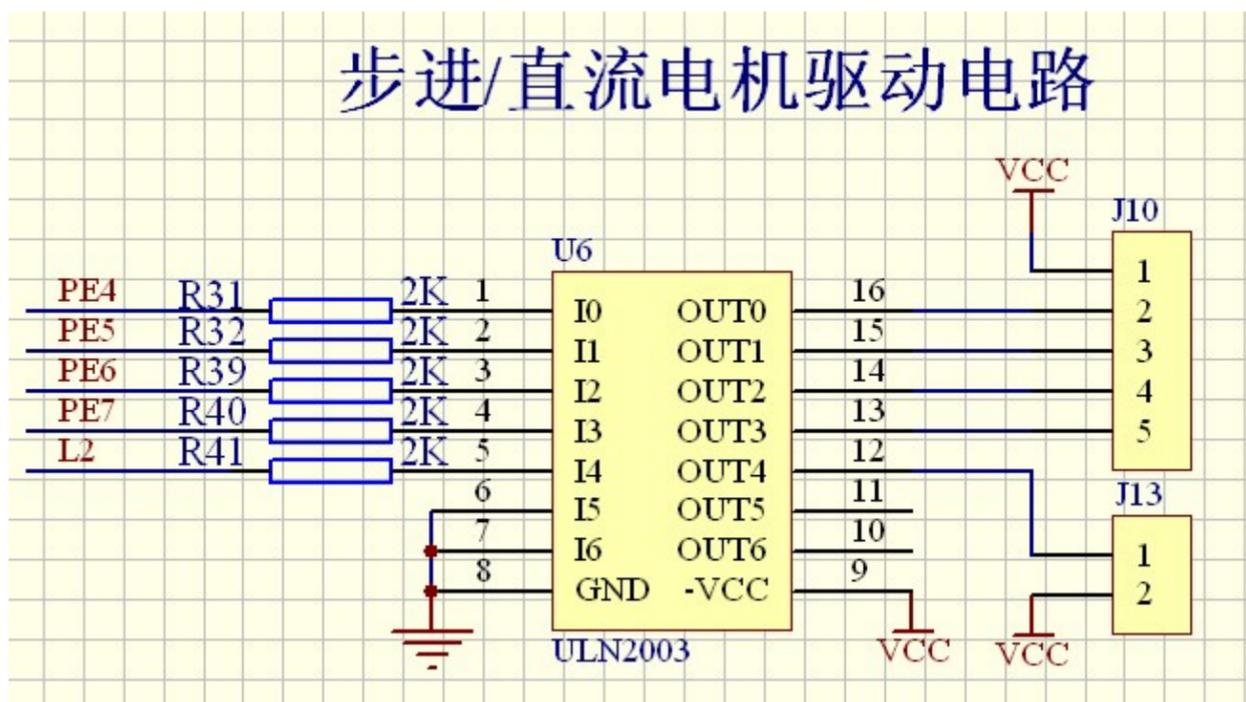
十五. 步进电机实验

【实验目的】

【实验设备】 略，同上；

【实验要求】。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。



十六. DS18B20 实验

【实验目的】

【实验设备】 略，同上；

【实验要求】。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。

DS18B20 数字温度计是 DALLAS 公司生产的 1-Wire，即单总线器件，具有线路简单，体积小特点。因此用它来组成一个测温系统，具有线路简单，在一根通信线，可以挂很多这样的数字温度计，十分方便。

DS18B20 的特点

- (1) . 只要求一个端口即可实现通信。
- (2) . 在 DS18B20 中的每个器件上都有独一无二的序列号。
- (3) . 实际应用中不需要外部任何元器件即可实现测温。
- (4) . 测量温度范围在 -55.0°C 到 $+125.0^{\circ}\text{C}$ 之间。
- (5) . 数字温度计的分辨率用户可以从 9 位到 12 位选择。
- (6) . 内部有温度上、下限告警设置。

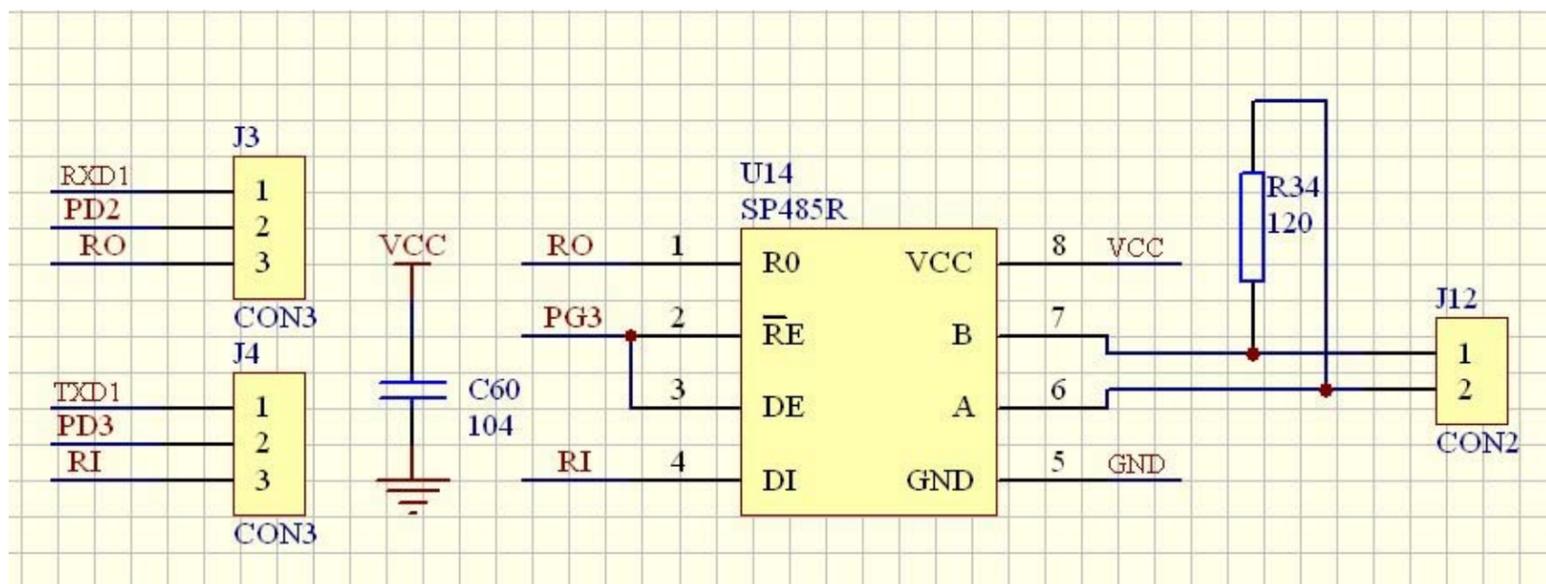
十七. 485 能讯实验

【实验目的】

【实验设备】 略，同上；

【实验要求】。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。



MAX485 是一个 8 个引脚的芯片，它是一个标准的 RS485 收发器，只能进行半双工的通讯，内含一个输出驱动器和一个信号接收器。MAX485 具有低功耗设计，静态电流仅为 300uA。MAX485 具有三态输出特性，在使用 MAX485 时，总线最多可以同时连接 32 个 MAX485 芯片。通讯波特率可以达到

2.5M

图 1 是 MAX485 的俯视图和逻辑图。

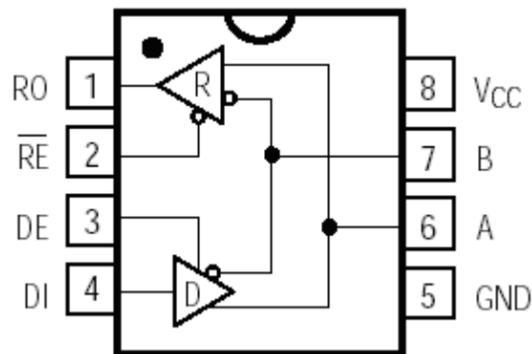


图 1 MAX485 逻辑图

下面是 MAX485 的引脚定义：

RO（引脚 1）：接收信号的输出引脚。可以把来自 A 和 B 引脚的总线信号，输出给单片机。是 COMS 电平，可以直接连接到单片机。

RE（引脚 2）：接收信号的控制引脚。当这个引脚低电平时，RO 引脚有效，MAX485 通过 RO 把来自总线的信号输出到单片机；当这个引脚高电平时，RO 引脚处于高阻状态。

DE（引脚 3）：输出信号的控制引脚。当这个引脚低电平时，输出驱动器无效；当这个引脚高电平时，输出驱动器有效，来自 DI 引脚的输出信号通过 A 和 B 引脚被加载到总线上。是 COMS 电平，可以直接连接到单片机。

DI（引脚 4）：输出驱动器的输入引脚。是 COMS 电平，可以直接连接到单片机。当 DE 是高电平时，这个引脚的信号通过 A 和 B 脚被加载给总线。

GND（引脚 5）：电源地线。

A（引脚 6）：连接到 RS485 总线的 A 端。

B（引脚 7）：连接到 RS485 总线的 B 端。

Vcc（引脚 8）：电源线引脚。电源 $4. \leq V_{cc} \leq 5. \text{V}$ 。

在一般情况下，可以直接把 MAX485 和单片机连接在一起。连接方法如上图所示。

MAX485 的控制引脚 2 和引脚 3 可以分别控制，也可以共同控制如图所示，在图中当 PG3（ $PORTG \mid = (1 \ll 3)$ ）为高电平时，MAX485 作为输出驱动器使用，来自单片机 TXD 的输出信号通过 A 和 B 引脚加载到 RS485 总线上；当 PG3（ $PORTG \& \sim (1 \ll 3)$ ）为低电平时，MAX485 作为信号接收器使用，来自 RS485 总线的信号通过 RO（1 号引脚）被读到单片机的 RXD

MAX485 的控制引脚 2 和 3 无论是分别控制还是共同控制，接收器和驱动器都不能够同时工作，因此 MAX485 只能工作在单工状态下或半双工状态下。

十八. 红外摇控实验

【实验目的】

【实验设备】 略，同上；

【实验要求】。

【实验原理】 查看 M128 开发板电路原理图，如下图所示。

