

实验指导书

课 程 名 称	单片机及嵌入式系统原理
学 院	信息工程学院
教 师 姓 名	周伟、马小林
授 课 时 间	秋季学期
授 课 班 级	信息工程

目 录

实验一 IO 口流水灯花样设计实现	1
一、实验目的及学时	1
二、实验内容	1
三、预备知识	1
四、实验设备及工具（包括软件调试工具）	1
五、实验原理	1
六、实验步骤	2
七、实验总结	8
八、思考题	8
实验二 双机通信实验	9
一、实验目的及学时	9
二、实验内容	9
三、预备知识	9
四、实验设备及工具（包括软件调试工具）	9
五、实验原理	9
六、实验步骤	11
七、实验总结	11
八、思考题	11
实验三 数据采集显示和变送实验	12
一、实验目的及学时	12
二、实验内容	12
三、预备知识	12
四、实验设备及工具（包括软件调试工具）	12
五、实验原理	12
六、实验步骤	15
七、实验总结	15
八、思考题	15
实验四 带日历秒表的语音报时电子钟	16
一、实验目的及学时	16
二、实验内容	16
三、预备知识	16
四、实验设备及工具（包括软件调试工具）	16
五、实验原理	16
六、实验步骤	17
七、实验总结	18

八、思考题.....	18
实验五 STM32 教室光照度采集显示	19
一、实验目的及学时	19
二、实验内容	19
三、预备知识	19
四、实验设备及工具（包括软件调试工具）	19
五、实验原理	20
七、实验总结	22
八、思考题	22
实验六 嵌入式系统 WIFI 联网	23
一、实验目的及学时	23
二、实验内容	23
三、预备知识	23
四、实验设备及工具（包括软件调试工具）	23
五、实验原理	24
六、实验步骤	24
七、实验总结	28
八、思考题	28

实验一 10 口流水灯花样设计实现

一、实验目的及学时

实验目的：

1. 掌握单片机 I/O 的工作方式。
2. 了解函数的基本结构，能够独立进入程序 Debug，多多动手操作，熟练 Keil 软件环境的一些基本操作。
3. 掌握 C 语言变量类型与取值范围，for、while 等基本语句的用法。

学时：2 学时（设计研究性实验）

二、实验内容

1. 熟悉硬件开发平台、软件开发环境和相关开发工具的使用方法；
2. 让 8 个小灯依次一个接一个的点亮，流动起来，也就是常说的流水灯；
3. 完成一个左移到头接着右移，右移到头再左移的花样流水灯程序。

三、预备知识

具备模拟电子和 C 语言理论知识，熟悉 Keil 单片机开发软件，能编写简单的 C 语言程序。

四、实验设备及工具（包括软件调试工具）

硬件：51 单片机开发板（下载器），PC 机。

软件：Keil 开发及调试软件。

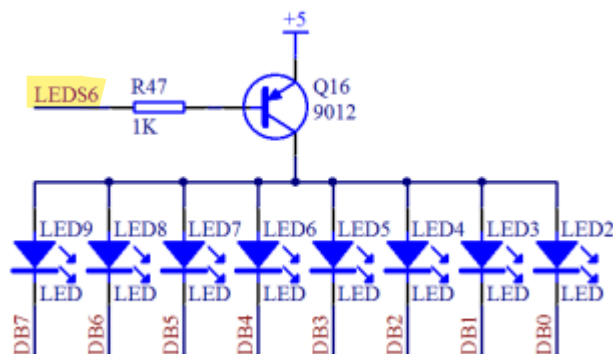
五、实验原理

通过前面的课程，我们可以了解到控制引脚 P0.0 经过 74HC245 控制了 DB0，P0.1 控制 DB1,,,P0.7 控制 DB7。我们还学到一个字节是 8 位，我们如果写一个 P0，就代表了 P0.0 到 P0.7 的全部 8 个位。比如我们写 $P0 = 0xFE$ ；转换成二进制就是 0b11111110，可以通过 P0 来控制所有的 8 个 LED 小灯的亮和灭。我们下边要进行依次亮和灭，怎么办呢？从这里就可以得到方法了，如果想让单片机流水灯流动起来，依次要赋给 P0 的数值就是：0xFE、0xFD、0xFB、0xF7、0xEF、0xDF、0xBF、0x7F。

在我们的 C 语言当中，有一个移位操作，其中<<代表的是左移，>>代表的是右移。比如 $a = 0x01 \ll 1$; 就是 a 的结果等于 $0x01$ 左移一位。大家注意，移位都是指二进制移位，那么移位完了，本来在第 0 位的 1 移动到了第一位上，移动完了低位是补 0 的。所以 a 的值最终是等于 $0x02$ 。

还要学习另外一个运算符~，这个符号是按位取反的意思，同理按位取反也是针对二进制而言。比如 $a = \sim(0x01)$; $0x01$ 的二进制是 $0b00000001$ ，按位取反后就是 $0b11111110$ ，那么 a 的值就是 $0xFE$ 了。需用到的相关电路原理图如下：

1. LED 小灯电路图

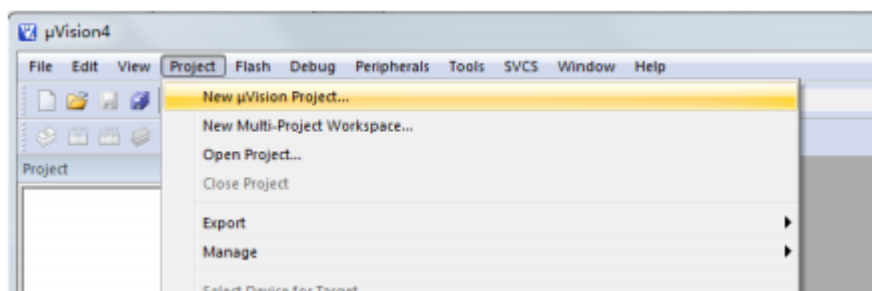


六、实验步骤

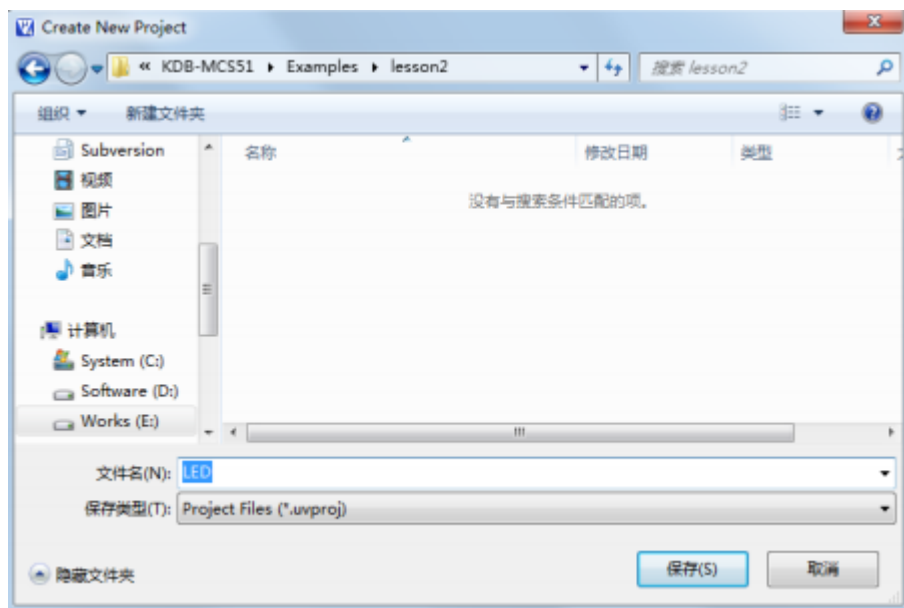
1. 检查开发板是否完整，连接下载器到电脑，打开电源开关。

2. 新建一个工程

对于单片机程序来说，每个功能程序，都必须要有有一个配套的工程（Project），即使是点亮 LED 这样简单的功能程序也不例外，因此我们首先要新建一个工程，打开我们的 Keil 软件后，点击：Project-->New uVision Project...然后会出现一个新建工程的界面：

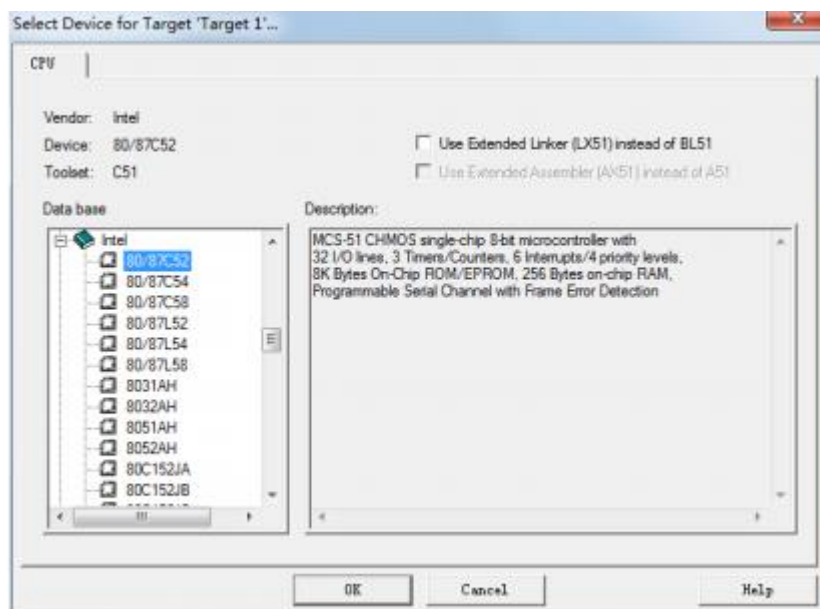


我们在硬盘上建立了一个 lesson2 的目录，然后把 LED 这个工程的路径指定到这里，这样方便今后管理程序，不同的功能程序放到不同的文件夹下，并且给这个工程起一个名字叫做 LED，软件会自动添加扩展名.uvproj。如图所示：



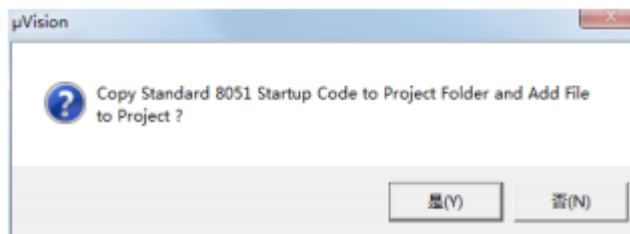
我们直接点保存就可以了，工程会自动保存成 LED.uvproj 文件，下次要打开 LED 这个工程时，可以直接找到文件夹，双击这个.uvproj 文件就可以直接打开了。

保存之后会弹出一个对话框，这个对话框让我们选择单片机型号。因为 Keil 软件是外国人开发的，所以我们国内的 STC89C52 并没有上榜，但是只要选择同类型号就可以了。因为 51 内核是由 Intel 公司创造的，所以这里我们直接选择 Intel 公司名下的 80/87C52 来代替，这个选项的选择对于后边的编程没有任何的不良影响。如图所示：

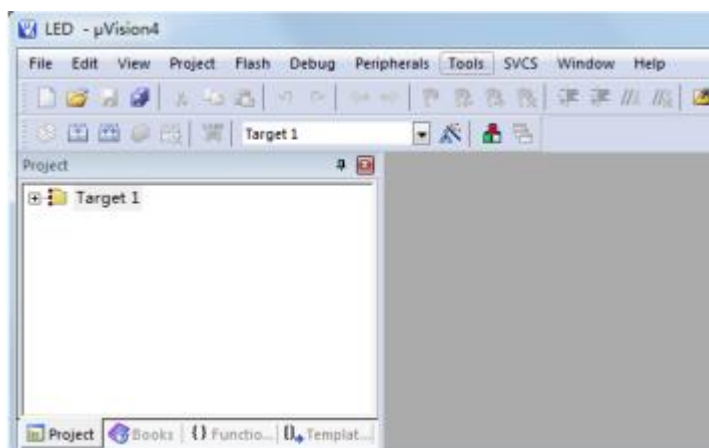


点击 OK 之后，会弹出一个对话框，如图 2-11 所示，每个工程都需要一段启动代码，如果点“否”编译器会自动处理这个问题，如果点“是”，这部分代码会提供给我们用户，我们就可以按需要自己去处理这部分代码，那这部分代

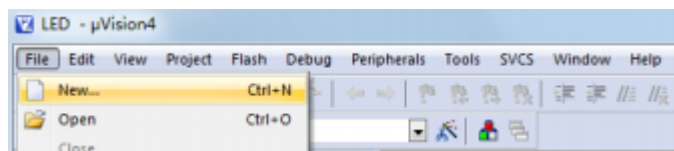
码在我们初学 51 的这段时间内，一般是不需要去修改的，但是随着技术的提高和知识的扩展，我们就有可能会需要了解这块内容，因此这个地方我们选一下“是”，让这段代码出现，但是我们暂时不需要修改它，大家知道这么回事就可以了。

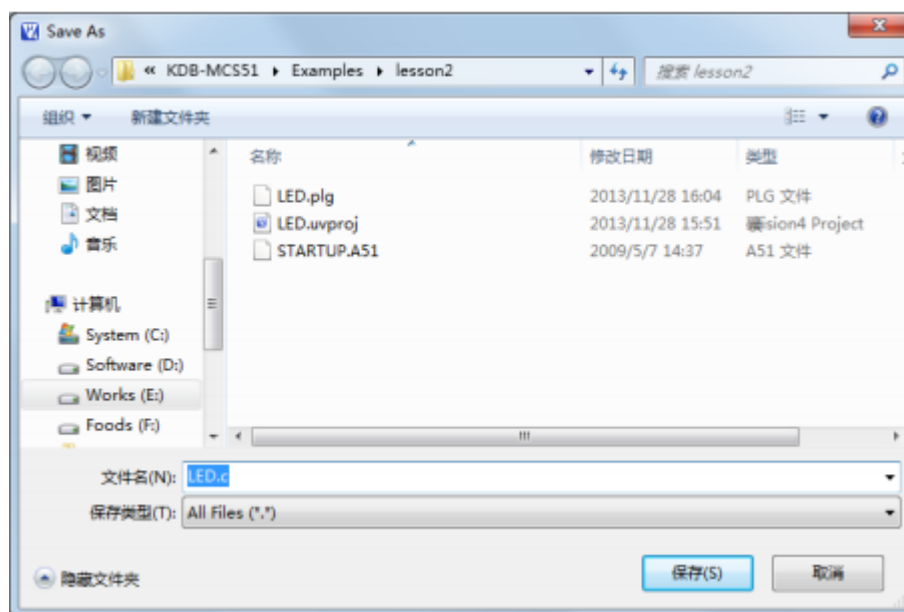


这样工程就建立好了，如下图所示，如果我们点击 Target 1 左边的加号，会出现我们刚才加入的初始化文件 **STARTUP.A51**，这个我们先不管。

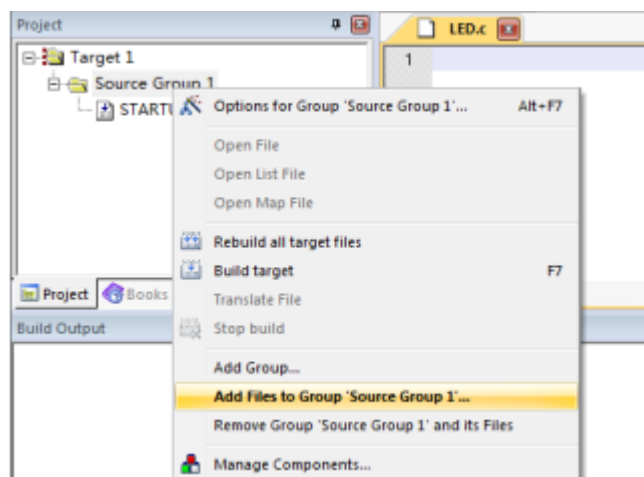


工程有了之后，我们要建立编写代码的文件，点击 File-->New，如图 2-13，新建一个文件，也就是我们编写程序的平台。然后点 File-->Save 或者直接点击那个 Save 的快捷键，可以保存文件，保存时我们把它命名为 LED.c，这个地方必须加上.c，因为如果写**汇编语言**，这个地方的扩展名是**.asm**，头文件就是**.h** 等等，我们编写的是 C 语言程序，这个地方必须自己添加文件的扩展名.c，如图所示：

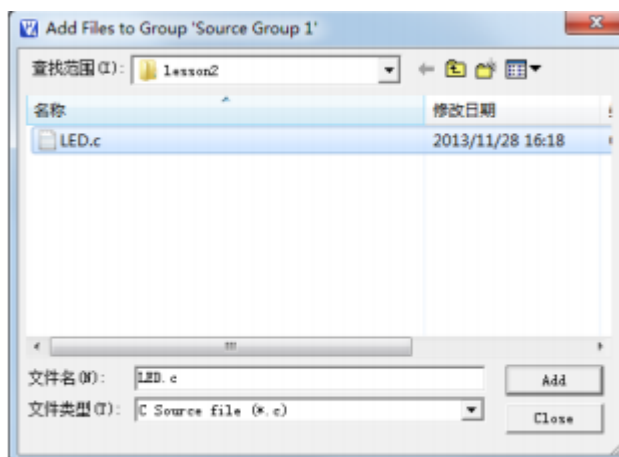




现在我们就可以在我们建立好的文件中输入我们的程序代码了，在编写之前还有个工作要做。我们每做一个功能程序，必须要新建一个工程，一个工程代表了单片机要实现的一个功能。但是一个工程，有时候我们可以把我们的程序分多个文件写，所以每写一个文件，我们都要添加到我们所建立的工程中去，鼠标右键点 Source Group 1，点 Add Files to Group„Source Group 1“...，如图所示：



在弹出的对话框中，单击 LED.c 选中它，然后点 Add，或者直接双击 LED.c 都可以将文件加入到这个工程下，然后单击 Close，关闭添加。这个时候大家会看到在 Source Group 1 下边又多了一个 LED.c 文件。

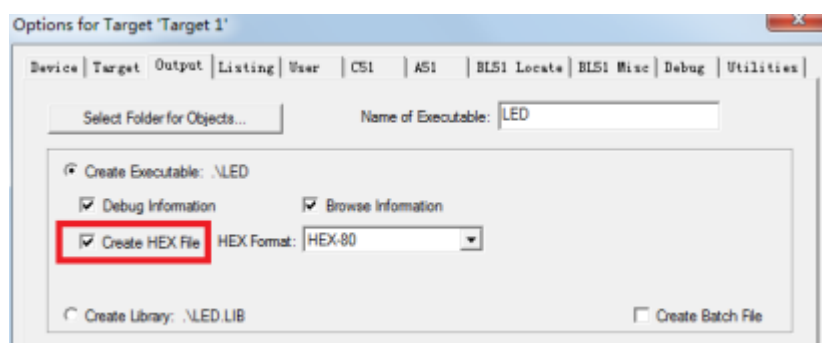


3. 程序编译

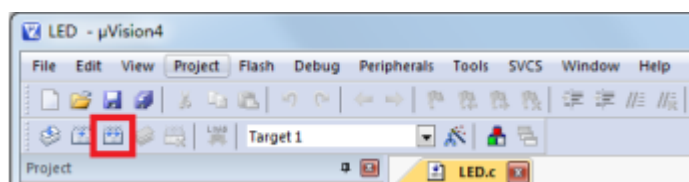
程序编好了，我们要对我们的程序进行**编译**，生成我们需要的可以下载到单片机里的文件，在编译之前，我们先要勾选一个选项，Project-->Options for Target „Target1“..., 或者直接点图中红框内的快捷图标：



在弹出的对话框中，点击 **Output** 选项页，勾选其中的 “ **Create HEX File** ”复选框，然后点 **OK**，如图 2-18 所示。



设置好以后呢，点击“ **Project-->rebuild all target files**”，或者鼠标点击图 2-19 中红框内的快捷图标，就可以对程序进行编译了。



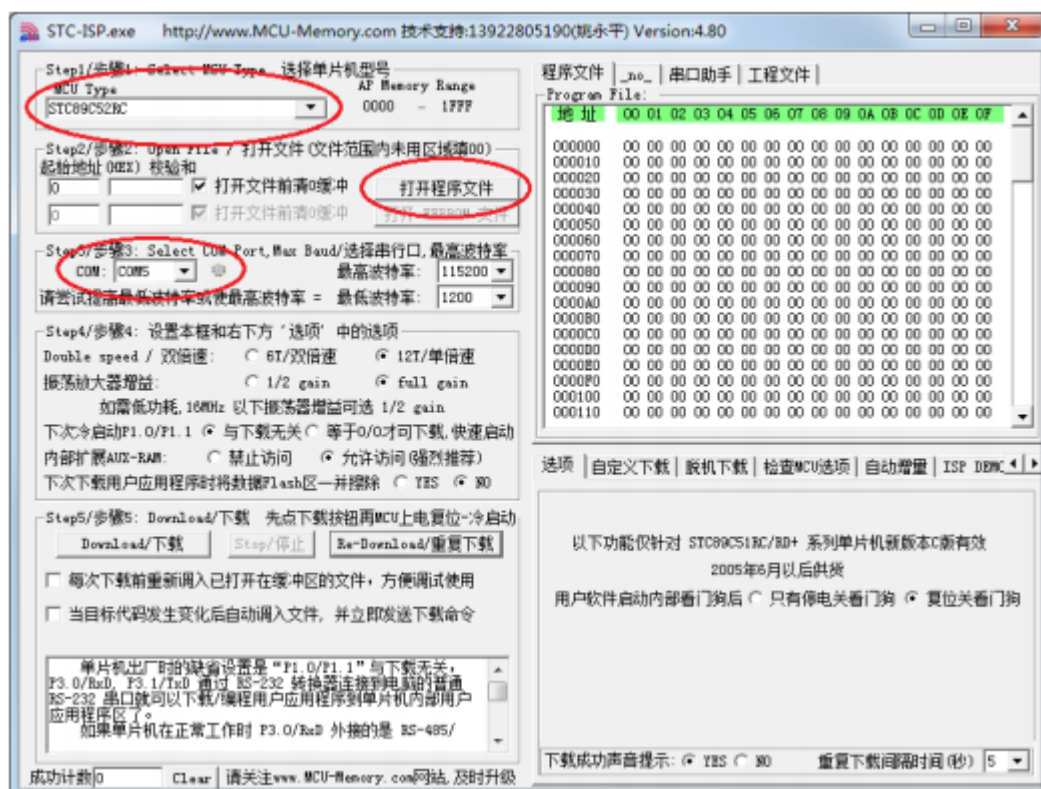
编译完成后，在我们的 Keil 下方的 Output 窗口会出现相应的提示，大家注意看图 2-20，这个窗口告诉我们编译完成后的情况，**data=9.0**，指的是我们的程序使用了单片机内部的 256 字节 RAM 资源中的 9 个字节，**code=29** 的意思是使用了 8K 代码 Flash 资源中的 29 个字节。当提示“0 Error(s), 0 warning(s)”表示我们的程序没有错误和警告，就会出现“creating hex file from “LED” ...”，意思是从当前工程生成了一个 HEX 文件，我们要下载到单片机上的就是这个 HEX 文件。如果出现有错误和警告提示的话，就是 Error 和 warning 不是 0，那么我们就需要对程序进行检查，找出问题，解决好了再进行编译产生 HEX 才可以。

4. 程序下载

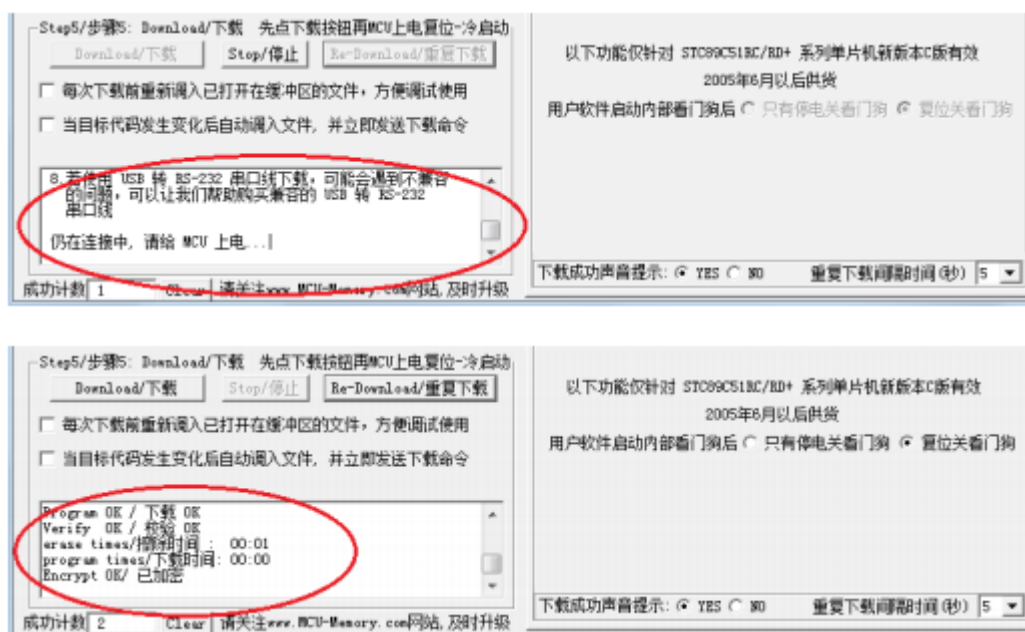
首先，我们要把硬件连接好，把板子插到我们的电脑上，打开设备管理器查看所使用的是哪个 COM，如图所示，找到“USB-SERIAL CH340(COM5)”这一项，这里最后的数字就是开发板目前所使用的 COM 端口号。



然后 STC 系列单片的下载软件——STC-ISP，如图所示。



下载软件列出了 5 个步骤：第一步，选择单片机型号，我们现在用的单片机型号是 **STC89C52RC**，这个一定不能选错了；第二步，点击“打开程序文件”，找到我们刚才建立工程的那个 lesson2 文件夹，找到 **LED.hex** 这个文件，点击打开；第三步，选择刚才查到的 COM 口，波特率使用默认的就行；第四步，这里的所有选项都使用默认设置，不要随便更改，有的选项改错了以后可能会产生麻烦。第五步，因为 **STC 单片机要冷启动下载**，就是先点下载，然后再给单片机**上电**，所以我们先关闭板子上的电源开关，然后点击“Download/下载”按钮，等待软件提示你请上电后，如图 2-23 所示，然后再按下板子的电源开关，就可以将程序下载到单片机里边了。当软件显示“**已加密**”就表示程序下载成功了，如图所示：



程序下载完毕后，就会自动运行。

七、实验总结

本实验主要结合 8 位单片机掌握 C 语言变量类型与取值范围, for、while 等基本语句的用法。其重点在于了解函数的基本结构，能够独立进入程序 Debug，多多动手操作，熟练 Keil 软件环境的一些基本操作。通过本实验，使得学生掌握左移、右移的应用。

八、思考题

请仔细阅读 **LED.c** 文件的代码，总结出单片机 C 语言编程与普通 PC 机中的 C 语言编程有何不同，至少给出 2 点不同之处。

实验二 双机通信实验

一、实验目的及学时

实验目的：

1. 理解 UART 串口通信的基本原理和通信过程。
2. 掌握通信的底层操作原理。
3. 学会通过配置寄存器，实现串口通信的基本操作过程。

学时：2 学时（设计研究性实验）

二、实验内容

1. 用 UART 串行通信实现 2 个单片机之间半双工通信；
2. 单片机 1 发送一个数据，单片机 2 可以收到；单片机 2 发送一个数据，单片机 1 可以收到；

三、预备知识

具备模拟电子和 C 语言理论知识，理解 UART 串口通信的基本原理和通信过程。熟悉 Keil 单片机开发软件，能编写简单的 C 语言程序。

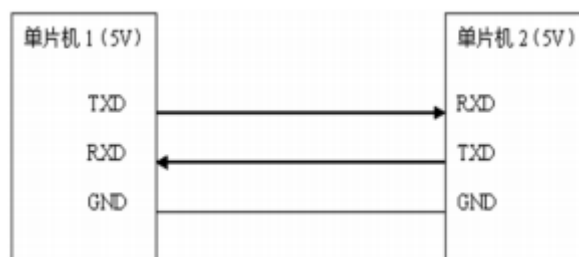
四、实验设备及工具（包括软件调试工具）

硬件：51 单片机开发板（下载器），PC 机。

软件：Keil 开发及调试软件。

五、实验原理

STC89C52 有两个引脚是专门用来做 UART 串行通信的，一个是 P3.0 一个是 P3.1，它们还分别有另外的名字叫做 RXD 和 TXD，由它们组成的通信接口就叫做串行接口，简称串口。用两个单片机进行 UART 串口通信，基本的演示图如下所示。



图中，GND 表示单片机系统电源的参考地，TXD 是串行发送引脚，RXD 是串行接收引脚。两个单片机之间要通信，首先电源基准得一样，所以我们要把两个单片机的 GND 相互连接起来，然后单片机 1 的 TXD 引脚接到单片机 2 的 RXD 引脚上，即此路为单片机 1 发送而单片机 2 接收的通道，单片机 1 的 RXD 引脚接到单片机 2 的 TXD 引脚上，即此路为单片机 2 发送而单片机 1 接收的通道。这个示意图就体现了两个单片机相互收发信息的过程。

发送而单片机 1 接收的通道。这个示意图就体现了两个单片机相互收发信息的过程。当单片机 1 想给单片机 2 发送数据时，比如发送一个 0xE4 这个数据，用二进制形式表示就是 0b11100100，在 UART 通信过程中，是低位先发，高位后发的原则，那么就让 TXD 首先拉低电平，持续一段时间，发送一位 0，然后继续拉低，再持续一段时间，又发送了一位 0，然后拉高电平，持续一段时间，发了一位 1，一直到把 8 位二进制数字 0b11100100 全部发送完毕。这里就涉及到了一个问题，就是持续的这“一段时间”到底是多久？由此便引入了通信中的一个重要概念——波特率，也叫做比特率。

波特率就是发送二进制数据位的速率，习惯上用 baud 表示，即我们发送一位二进制数据的持续时间=1/baud。在通信之前，单片机 1 和单片机 2 首先都要明确的约定好它们之间的通信波特率，必须保持一致，收发双方才能正常实现通信，这一点大家一定要记清楚。约定好速度后，我们还要考虑第二个问题，数据什么时候是起始，什么时候是结束呢？不管是提前接收还是延迟接收，数据都会接收错误。在 UART 通信的时候，一个字节是 8 位，规定当没有通信信号发生时，通信线路保持高电平，当要发送数据之前，先发一位 0 表示起始位，然后发送 8 位数据位，数据位是先低后高的顺序，数据位发完后再发一位 1 表示停止位。这样本来要发送一个字节的 8 位数据，而实际上我们一共发送了 10 位，多出来的两位其中一位起始位，一位停止位。而接收方呢，原本一直保持的高电平，一旦检测到了一位低电平，那就知道了要开始准备接收数据了，接收到 8 位数据位后，然后检测到停止位，再准备下一个数据的接收。如下图所示。



实际上是一个时域示意图，就是信号随着时间变化的对应关系。比如在单片机的发送引脚上，左边的是先发生的，右边的是后发生的，数据位的切换时间就是波特率分之一秒。

51 单片机的 UART 串口的结构由串行口控制寄存器 SCON、发送和接收电

路三部分构成，先来了解一下串口控制寄存器 SCON。如表 11-1 表 11-2 所示。

表 11-1 SCON——串行控制寄存器的位分配（地址 0x98、可位寻址）

位	7	6	5	4	3	2	1	0
符号	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
复位值	0	0	0	0	0	0	0	0

表 11-2 SCON——串行控制寄存器的位描述

位	符号	描述
7	SM0	这两位共同决定了串口通信的模式 0~模式 3 共 4 种模式。我们最常用的就是模式 1，也就是 SM0=0，SM1=1，下边我们重点就讲模式 1，其它模式从略。
6	SM1	
5	SM2	多机通信控制位（极少用），模式 1 直接清零。
4	REN	使能串行接收。由软件置位使能接收，软件清零则禁止接收。
3	TB8	模式 2 和 3 中要发送的第 9 位数据（很少用）。
2	RB8	模式 2 和 3 中接收到的第 9 位数据（很少用），模式 1 用来接收停止位。
1	TI	发送中断标志位，当发送电路发送到停止位的中间位置时，TI 由硬件置 1，必须通过软件清零。
0	RI	接收中断标志位，当接收电路接收到停止位的中间位置时，RI 由硬件置 1，必须通过软件清零。

对于串口的四种模式，模式 1 是最常用的，就是我们前边提到的 1 位起始位，8 位数据位和 1 位停止位。

六、实验步骤

- 1、配置串口为模式 1。
- 2、配置定时器 T1 为模式 2，即自动重装模式。
- 3、根据波特率计算 TH1 和 TL1 的初值，如果有需要可以使用 PCON 进行波特率加倍。
- 4、打开定时器控制寄存器 TR1，让定时器跑起来。

七、实验总结

本次实验可进一步巩固串口通信的原理及实践操作。通过本实验，使得学生巩固两个单片机之间相互通信的原理，并将理论课中所学的定时器重载的计算应用到实验中，从而加深对定时器原理的理解。

八、思考题

请总结单片机 UART 串口通信的实现流程。

实验三 数据采集显示和变送实验

一、实验目的及学时

实验目的：

1. 能够灵活使用 1602 液晶显示任意字符串。
2. 掌握 A/D 的基本概念和性能指标。
3. 掌握PCF8591的 I2C 软件编写。

学时：2 学时（设计研究性实验）

二、实验内容

1. 将 AD 采集到的数值显示到液晶屏上，调节 R62，AIN0 采样值相应变化；
2. PCF8591 利用串行 I2C 总线接口与单片机通信传送 AD 采样数据；

三、预备知识

具备模拟电子和 C 语言理论知识，理解 A/D 的基本概念和性能指标。熟悉 Keil 单片机开发软件，能编写简单的 C 语言程序。

四、实验设备及工具（包括软件调试工具）

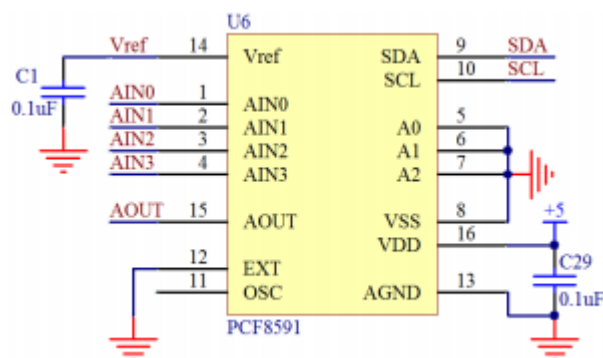
硬件：51 单片机开发板（下载器），PC 机。

软件：Keil 开发及调试软件。

五、实验原理

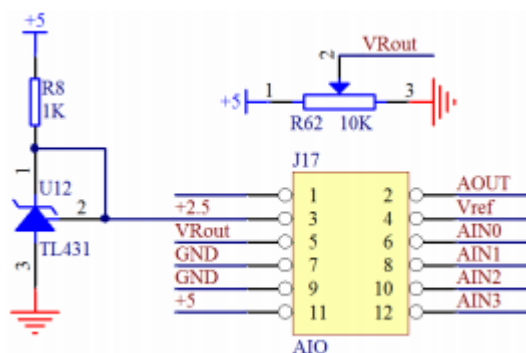
1. PCF8591 的硬件接口。

PCF8591 是一个单电源低功耗的 8 位 CMOS 数据采集器件，具有 4 路模拟输入，1 路模拟输出和一个串行 I2C 总线接口用来与单片机通信。与前面讲过的 24C02 类似，3 个地址引脚 A0、A1、A2 用于编程硬件地址，允许最多 8 个器件连接到 I2C 总线而不需要额外的片选电路。器件的地址、控制以及数据都是通过 I2C 总线来传输，我们先看一下 PCF8591 的原理图，如下图所示。



其中引脚 1、2、3、4 是 4 路模拟输入，引脚 5、6、7 是 I2C 总线的硬件地址，8 脚是数字地 GND，9 脚和 10 脚是 I2C 总线的 SDA 和 SCL。12 脚是时钟选择引脚，如果接高电平表示用外部时钟输入，接低电平则用内部时钟，我们这套电路用的是内部时钟，因此 12 脚直接接 GND，同时 11 脚悬空。13 脚是模拟地 AGND，在实际开发中，如果有比较复杂的模拟电路，那么 AGND 部分在布局布线上要特别处理，而且和 GND 的连接也有多种方式，这里大家先了解即可。在我们板子上没有复杂的模拟部分电路，所以我们将 AGND 和 GND 接到一起。14 脚是基准源，15 脚是 DAC 的模拟输出，16 脚是供电电源 VCC。PCF8591 的 ADC 是逐次逼近型的，转换速率算是中速，但是它的速度瓶颈在 I2C 通信上。由于 I2C 通信速度较慢，所以最终的 PCF8591 的转换速度，直接取决于 I2C 的通信速率。由于 I2C 速度的限制，所以 PCF8591 得算是个低速的 AD 和 DA 的集成，

Vref 基准电压的提供有两种方法。一是采用简易的原则，直接接到 VCC 上去，但是由于 VCC 会受到整个线路的用电功耗情况影响，一来不是准确的 5V，实测大多在 4.8V 左右，二来随着整个系统负载情况的变化会产生波动，所以只能用在简易的、对精度要求不高的场合。方法二是使用专门的基准电压器件，比如 TL431，它可以提供一个精度很高的 2.5V 的电压基准，这是我们通常采用的方法。如下图所示。

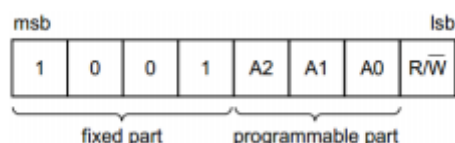


图中 J17 是双排插针，大家可以根据自己的需求选择跳线帽短接还是使用杜邦线连接其它外部电路，二者都是可以的。在这个地方，我们直接把 J17 的 3 脚和 4 脚用跳线帽短路起来，那么现在 Vref 的基准源就是 2.5V 了。分别把 5 和 6、7 和 8、9 和 10、11 和 12 用跳线帽短接起来的话，那么我们的 AIN0 实测的就是电位器的分压值，AIN1 和 AIN2 测的是 GND 的值，AIN3 测的是 +5V 的

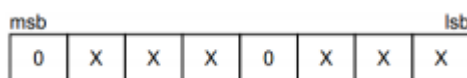
值。这里需要注意的是，AIN3 虽然测的是+5V 的值，但是对于 AD 来说，只要输入信号超过 Vref 基准源，它得到的始终都是最大值，即 255，也就是说它实际上无法测量超过其 Vref 的电压信号的。需要注意的是，所有输入信号的电压值都不能超过 VCC，即+5V，否则可能会损坏 ADC 芯片。

2.PCF8591 的软件编程

PCF8591 的通信接口是 I2C，那么编程肯定是要符合这个协议的。单片机对 PCF8591 进行初始化，一共发送三个字节即可。第一个字节，和 EEPROM 类似，是器件地址字节，其中 7 位代表地址，1 位代表读写方向。地址高 4 位固定是 0b1001，低三位是 A2, A1, A0，这三位我们电路上都接了 GND，因此也就是 0b000，如下图所示。

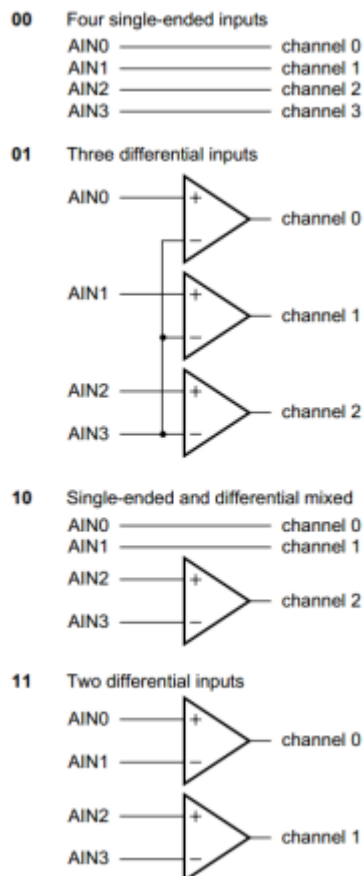


发送到 PCF8591 的第二个字节将被存储在控制寄存器，用于控制 PCF8591 的功能。其中第 3 位和第 7 位是固定的 0，另外 6 位各自有各自的作用，如下图所示，对于串口的四种模式，模式 1 是最常用的，就是我们前边提到的 1 位起始位，8 位数据位和 1 位停止位。



控制字节的第 6 位是 DA 使能位，这一位置 1 表示 DA 输出引脚使能，会产生模拟电压输出功能。第 4 位和第 5 位可以实现把 PCF8591 的 4 路模拟输入配置成单端模式和差分模式，单端模式和差分模式的区别，我们在 17.5 节有介绍，这里大家只需要知道这两位是配置 AD 输入方式的控制位即可，如下图所示。

控制字节的第 2 位是自动增量控制位，自动增量的意思就是，比如我们一共有 4 个通道，当我们全部使用的时候，读完了通道 0，下一次再读，会自动进入通道 1 进行读取，不需要我们指定下一个通道，由于 A/D 每次读到的数据，都是上一次的转换结果，所以同学们在使用自动增量功能的时候，要特别注意，当前读到的是上一个通道的值。



六、实验步骤

- 1、短接 J17 的 3、4 脚、5、6 脚。
- 2、编写程序，直到编译成功。
- 3、下载调试。
- 4、调节 R62，看 AIN0 采样值是否相应变化。

七、实验总结

本次实验可进一步巩固 PCF8591 的硬件原理及软件实践操作。通过本实验，使得学生巩固 I2C 通信的原理，并将理论课中所学的 AD 分辨率、转换速率应用到实验中，从而加深对 ADC 性能指标的理解。

八、思考题

- (1) 请简要说明 AD 分辨率的计算方法；
- (2) 请总结 AIN0 采样值随 R62 调节的变化规律。

实验四 带日历秒表的语音报时电子钟

一、实验目的及学时

实验目的：

1. 能够灵活使用 1602 液晶、数码管等显示任意字符串。
2. 掌握 A/D、D/A 的基本概念和性能指标。
3. 掌握定时器、中断的应用。

学时：2 学时（设计研究性实验）

二、实验内容

1. 具有走时（包括时间、日期、星期）、板载按键校时功能外，还提供闹钟、温度测量、红外遥控校时这几项实用功能；
2. 时间用数码管来显示；日期、闹钟设置（蜂鸣器）、温度等辅助信息显示到液晶上；星期用点阵显示；校时用遥控器或者板载按键。

三、预备知识

具备模拟电子和 C 语言理论知识，理解 51 单片机的基础知识。熟悉 Keil 单片机开发软件，能编写简单的 C 语言程序。

四、实验设备及工具（包括软件调试工具）

硬件：51 单片机开发板（下载器），PC 机。

软件：Keil 开发及调试软件。

五、实验原理

1. 结构划分。

功能上大概列举出来了，那么我们就可以把程序源代码划分为这样几个模块：DS1302 作为走时的核心自成一个模块；点阵、数码管、独立 LED 都属于 LED 的范畴，控制方式类似，也都需要动态扫描，所以把它们整体作为一个模块；液晶是另一个显示模块；按键和遥控器的驱动各自成为一个模块。

我们要进行一个实际产品或者项目开发的时候，首先电路原理图是确定的，所使用的单片机的引脚也是明确的，还有一些比如类型说明，一些特殊的全局参

数及宏声明，我们会放到一个专门的头文件中，在这里我们命名为 `config.h`，即全局的配置文件。这个 `config.h` 中包含了系统所共同使用的类型声明以及宏声明。

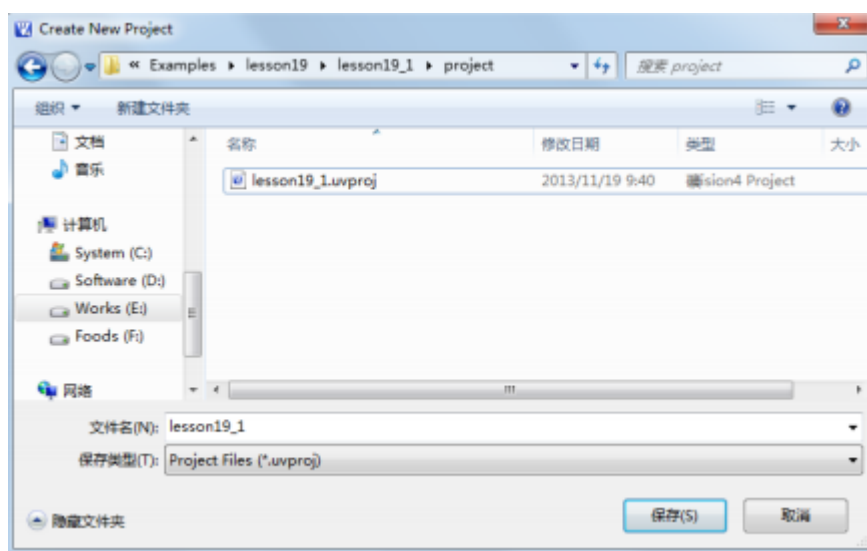
2. 代码编写

1602 液晶的底层驱动我们之前都已经写过了，直接拿过来用就行了。

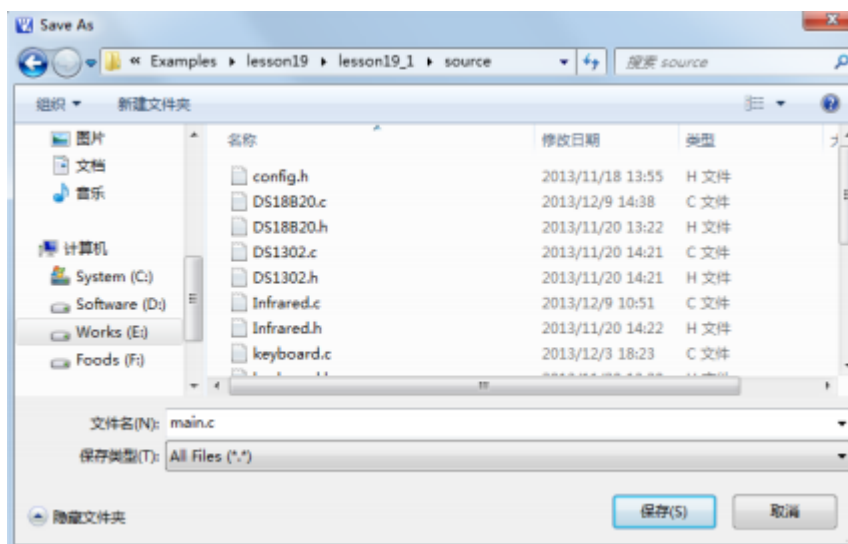
红外所要实现的功能是和按键完全一样的，但是如果说我们把红外按键的代码解析出来后，再去判断键码做相应的操作显得有点多余了。我们的处理方式是，把红外的按键代码解析出来后，把它们映射成标准键盘的键码，就跟板载按键的映射一样，这样红外和板载按键就可以很方便的共用一套应用层接口了，我们的应用层代码也只需要写一遍就 OK，而不需要针对不同的输入设备做不同的函数，从这里是不是又能体会到一次程序接口标准化和结构层次化的好处呢。但红外键码的映射与板载按键的映射不同，红外键码值不像矩阵按键的行列那样有规律，所以我们这里用一个二维数组来完成这个映射，二维数组每一行的第一个元素是红外遥控器的键码，第二个元素是该键要映射成的标准键码，不要的按键直接映射成 0 即可。这样，当收到一个红外键码后，在这个二维数组每行的第一个元素中查找相同值，找到后即把该行的第二个元素作为参数调用按键动作函数即可。

六、实验步骤

1、在实际项目开发中，我们不仅仅希望我们的源程序、头文件等文件结构规范、代码编写规范，而且还希望我们的工程文件规整规范，方便维护。因此我们首先新建一个 `lesson19_1` 的文件夹，用来存放我们本章的工程文件。而后我们新建工程保存的时候，在 `lesson19-1` 文件夹内再建立一个文件夹，取名为 `project`，专门用于存放工程文件的，如下图所示。



2、然后我们新建文件，保存的时候，在 `lesson19_1` 目录再建立一个文件夹，取名为 `source` 文件夹，专门用来存放我们的源代码，如下图所示。



3、最后，随便看一个之前的例子都能看到，工程编译后会生成很多额外的文件，这些文件可以统称为编译输出文件，我们在 lesson19_1 目录下再建立一个 output 文件夹来存放这些文件，要改变输出文件它们的路径，需要修改两个地方：首先进入 Options for Target，选择 Output 选项页，点击 Select Folder for Objects，在弹出的对话框中选择新建的 output 文件夹；然后再进入 Listing 选项页，点击 Select Folder for Listings，同样指定 output 文件夹即可。

七、实验总结

本次实验是对单片机学习知识整体运用的一个考核。通过本实验，使得学生可以独立完成一个开发项目的代码，并将理论课中所学的按键、显示、遥控器、采集等综合应用到实验中，从而加深对理论知识的理解。

八、思考题

- (1) 请参照本实验的核心代码及硬件结构，总结运用单片机进行小型功能设计的总体流程；
- (2) 请依据自身实验过程，结合自身体会，总结几点单片机开发的注意事项。请至少给出 3 条。

实验五 STM32 教室光照度采集显示

一、实验目的及学时

实验目的：

- 1、熟悉 MDK 创建和配置 STM32 工程项目的流程；
- 2、熟悉 STM32 官方库文件的应用；
- 3、熟悉 STM32 光照采集的相关知识；
- 4、了解 STM32ADC 在嵌入式开发中的应用；
- 5、熟悉串口通信的基本操作
- 6、

学时：2 学时（设计研究性实验）

二、实验内容

本次实验通过用 C 语言编写 STM32 教室光照度采集显示的程序，在官方例程的基础上，进行二次开发，实现实时采集教室光照度，并能够传输到 PC 端进行显示。建立正确的嵌入式开发编程理念，以及在嵌入式系统中相应的开发流程，区别于 51 进行对比学习，深入掌握光照采集的配置过程和使用，了解 STM32 与 51 单片机的优点。

三、预备知识

基本嵌入式单片机硬件知识、单片机软件编程 C 语言、程序创建和调试的基本方法、程序下载的基本操作、固件库单通道 ADC 采集的配置方法。

四、实验设备及工具（包括软件调试工具）

硬件：STM32 挑战者开发板， JTAG 下载盒一个，USB 转串口线一根、电源线一根。

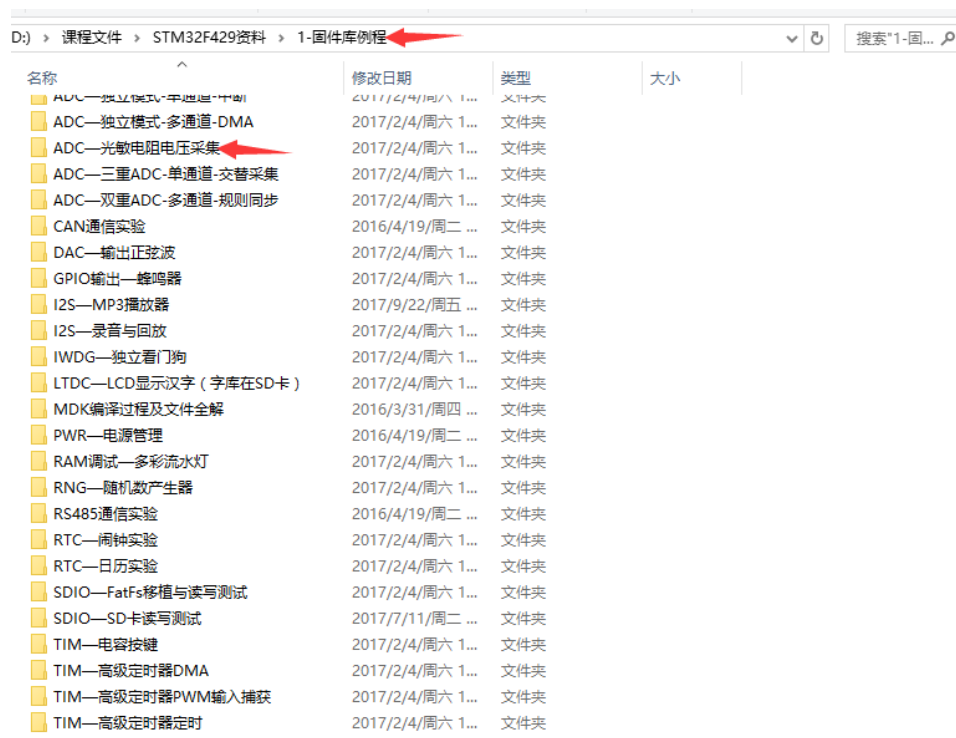
软件：STM32 官方库；PC 操作系统 Windows 98、Windows 2000 或 Windows XP；KEIL MDK 集成开发环境；USB 转串口驱动。

五、实验原理

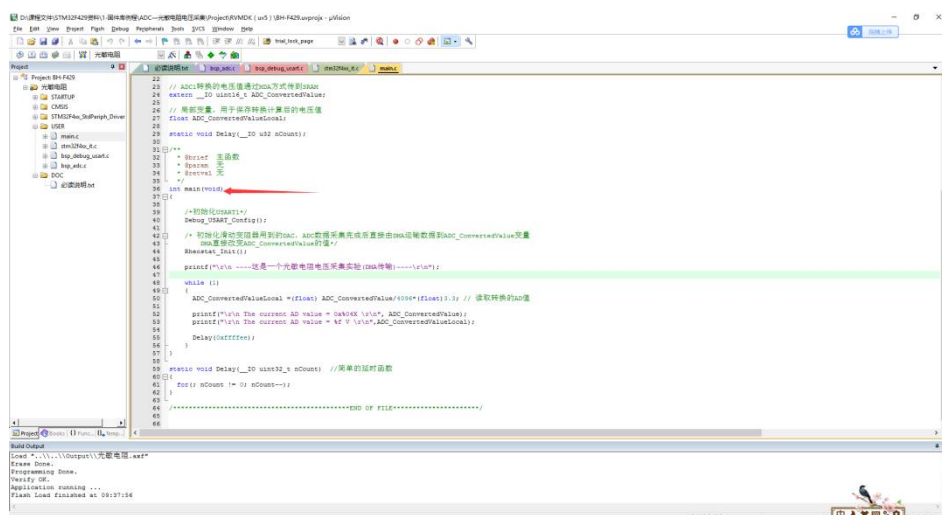
- 1、通过光敏电阻分压，得到一个跟光强有关的模拟电压；
- 2、由单片机 AD 转换把电压转换为数值；
- 3、将数值通过串口传送到 PC 电脑观察，通过数值的变化了解教室的光强变化。

六、实验步骤

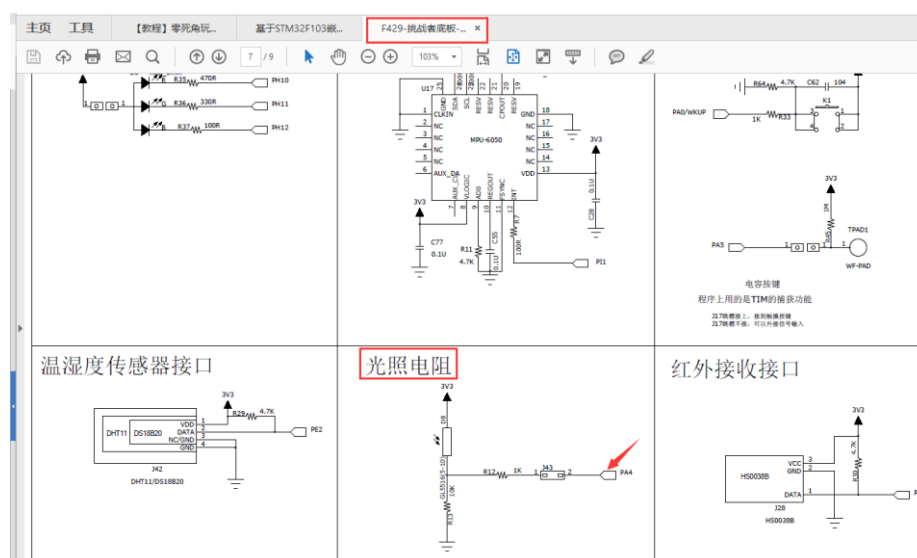
- 1、拷贝 STM32 开发板资料盘中固件库例程到电脑上。



找到 ADC-光敏电阻电压采集文件，打开工程。



2、找到挑战者底板-原理图，找到光照电阻部分，跳线帽：使用跳线帽连接 板子右侧的 LS<--->PA4 排针，得知相应的光照电阻引脚之后再去查看相应底层程序的设计及理解。



3、分析相应主函数代码，并且与 51 单片机进行对比。

```
int main(void)
{
    /*初始化USART1*/
    Debug_USART_Config();

    /* 初始化滑动变阻器用到的DAC, ADC数据采集完成后直接由DMA运输数据到ADC_ConvertedValue变量
    DMA直接改变ADC_ConvertedValue的值*/
    Rheostat_Init();

    printf("\r\n ----这是一个光敏电阻电压采集实验(DMA传输)----\r\n");

    while (1)
    {
        ADC_ConvertedValueLocal = (float) ADC_ConvertedValue/4096*(float)3.3; // 读取转换的AD值

        printf("\r\n The current AD value = 0x%04X \r\n", ADC_ConvertedValue);
        printf("\r\n The current AD value = %f V \r\n", ADC_ConvertedValueLocal);

        Delay(0xffffee);
    }
}
```

4、编译通过后调试，具体的下载过程见【教程】零死角玩转 STM32 第 3 章 如

何用 DAP 仿真器下载程序



3.1 仿真器简介

本书配套的仿真器为 Fire-Debugger，遵循 ARM 公司的 CMSIS-DAP 标准，支持所有基于 Cortex 内核的单片机，常见的 M3、M4 和 M7 都可以完美支持，其外观见图 3-1。

Fire-Debugger 支持下载和在线仿真程序，支持 XP/WIN7/WIN8/WIN10 这四个操作系统，免驱，不需要安装驱动即可使用，支持 KEIL 和 IAR 直接下载，非常方便。



5、使用 USB 转串口连接 PC 和 STM32 单片机，在 PC 上的串口调试小助手进行查看光照采集的数据，并且可以再开发板上通过遮挡光照传感器进行调试并查看相应的采集数据。串口配置设置为 115200-N-8-1。

七、实验总结

本次实验是对嵌入式单片机学习知识整体运用的一个考核。通过本实验，使得学生可以独立完成一个高性能高密度嵌入式系统开发项目的初级代码，并将理论课中所学的 AD 转换，通信等知识结合实际，从而加深对理论知识的理解。

八、思考题

- (1) 请参照本实验的核心代码及硬件结构，总结嵌入式单片机和 51 单片机在系统资源上的差异以及需要用到固件库做单片机丰富的外部设备资源的配置；
- (2) 请依据自身实验过程，结合自身体会，总结几点单片机开发的注意事项。请至少给出 3 条。

实验六 嵌入式系统 Wifi 联网

一、实验目的及学时

实验目的：

- 1、熟悉 MDK 创建和配置 STM32 工程项目的流程；
- 2、熟悉 STM32 官方库文件的应用；
- 3、熟悉 STM32wifi 通信的相关知识；
- 4、了解 STM32DCMI 数字摄像头接口的应用；
- 5、熟悉摄像头 ov5640 的基本原理。

学时：2 学时（设计研究性实验）

二、实验内容

本次实验通过用 C 语言编写 STM32wifi 摄像头实验，在官方例程的基础上，可以进行二次开发，实现实时采集摄像头的视频信号，并能够在同一网关下实时传输到 PC 端进行显示。建立正确的嵌入式开发编程理念，以及在嵌入式系统中相应的开发流程，理解无线网络通信的基本概念，在同一网关下进行无线传输视频信号，对摄像头 ov5640 的工作原理有一定的了解，能清楚通信过程以及程序代码的阅读分析。

三、预备知识

基本单片机硬件知识、单片机软件编程语言、程序创建和调试的基本方法、程序下载的基本操作、摄像头 ov5640 的使用说明是、无线网络通信的基础知识、STM32DCMI 数字摄像头接口的应用。

四、实验设备及工具（包括软件调试工具）

硬件：STM32 挑战者开发板， JTAG 下载盒一个，摄像头 OV5640, 电源线一根。

软件：STM32 官方库；PC 操作系统 Windows 98、Windows 2000 或 Windows XP；KEIL MDK 集成开发环境。

五、实验原理

- 1、通过 DCIM 接口将摄像头图像采集到嵌入式单片机系统
- 2、通过 wifi 将图像传递到 PC 电脑观察。

六、实验步骤

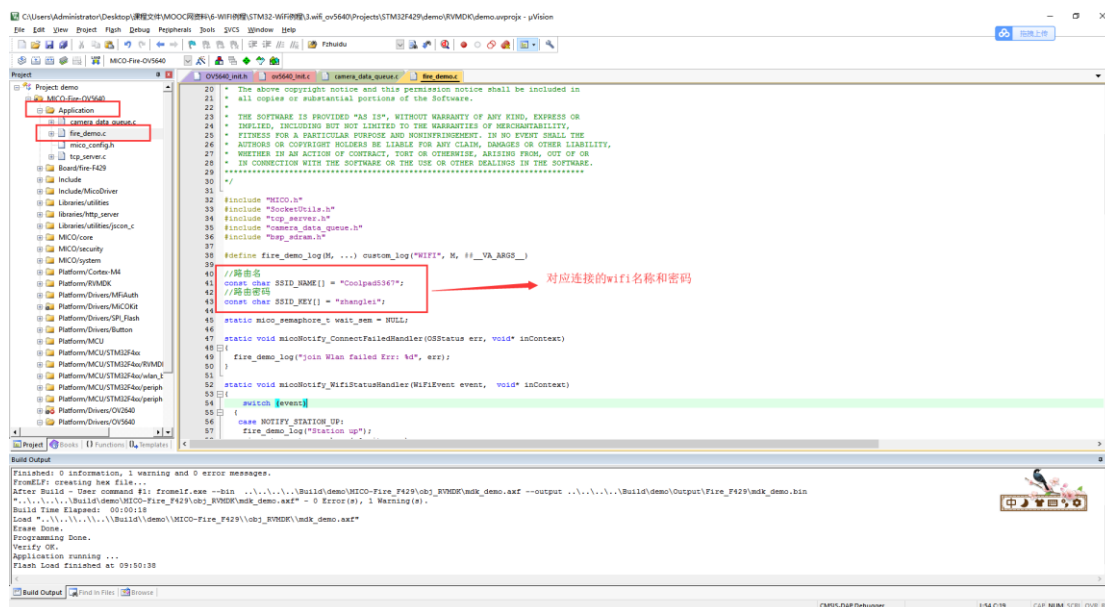
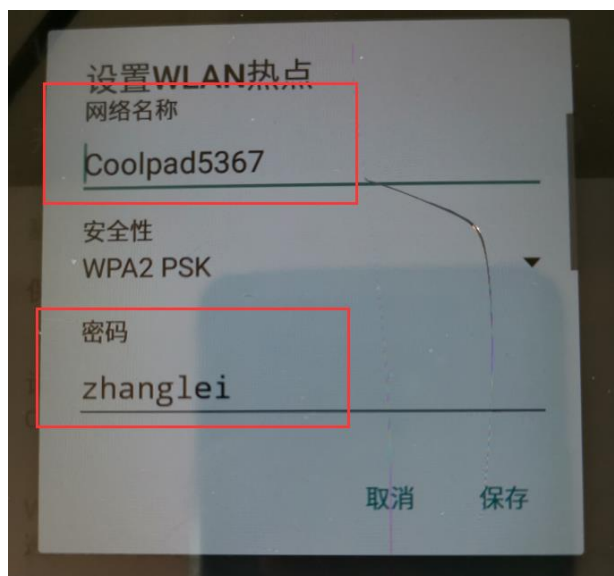
- 1、Wifi 例程比较多，一定要仔细阅读相关到的资料，在现有工程基础上可进行二次开发。在资料盘中或者无线网盘中拷贝 wifi 例程资料到电脑上。



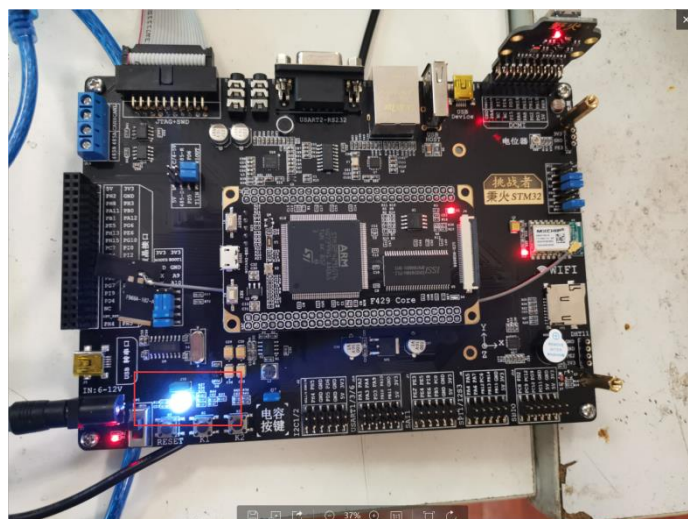
- 2、WIFI 例程打开之后文件很多，按照指定路径找到 ov5640 的源程序代码。



- 3、打开 demo 源代码，需要我们修改连接的 IP 地址，在实验室环境下，可以用自己的手机打开热点，找到相应的账号和密码。

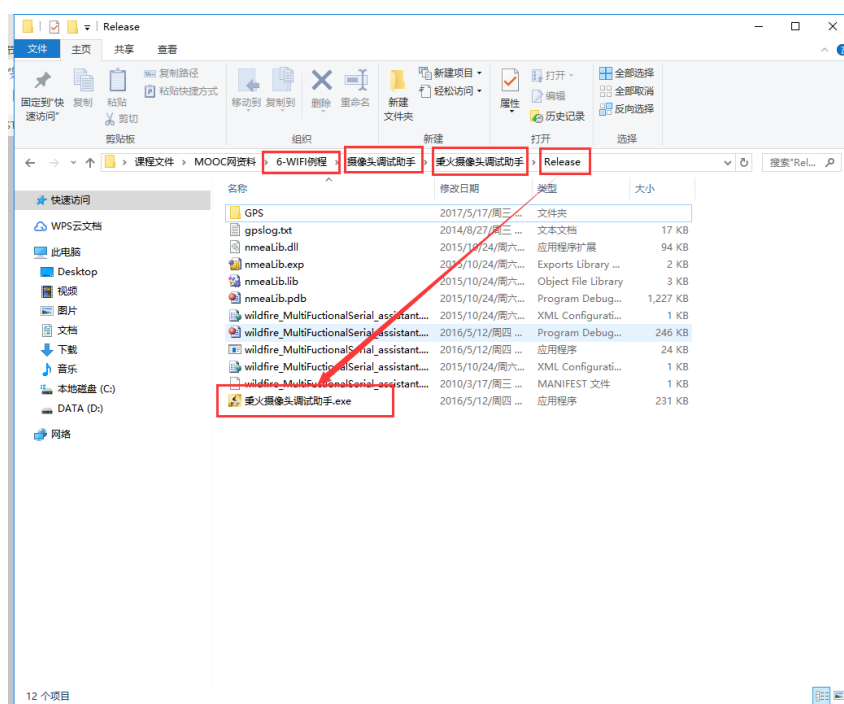


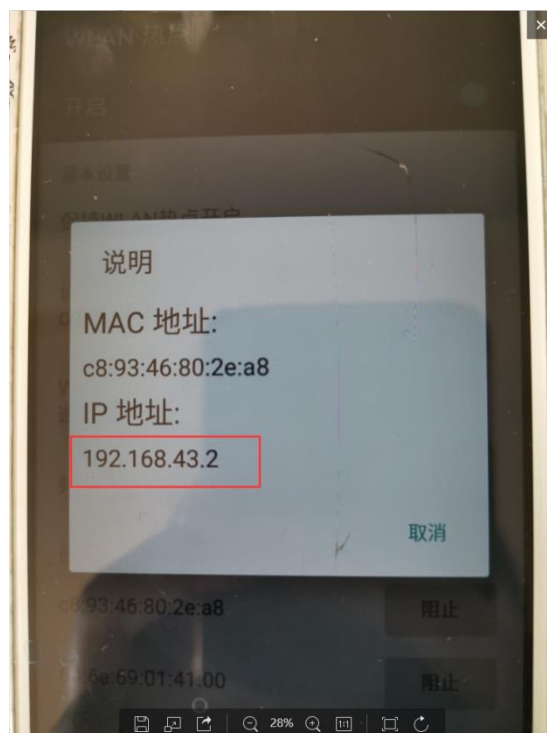
4、ov5640 搭载到 stm32 核心开发板上的 DCMI 输出接口上，上电之后下载刚刚修改过的程序到核心板中，当核心板中对应的灯亮时代表视频信号已经在进行



传输中。

5、找到摄像头调试助手，打开之后查找手机上对应的 IP 地址，然后在摄像头调试助手输入对应的 IP 地址，打开之后就可以拍摄想拍的内容了。





七、实验总结

本次实验是对嵌入式单片机学习知识整体运用的一个考核。通过本实验，使得学生可以独立完成较大数据流量的嵌入式系统开发项目代码，并将理论课中所学嵌入式系统数据处理和数据通信等知识结合实际，从而加深对理论知识的理解。

八、思考题

- (1) 请参照本实验的核心代码及硬件结构，总结嵌入式单片机和 51 单片机在系统资源上的差异以及需要用到 DMA 提高数据交换速度和 TCP 等协议栈的配置和运用；
- (2) 请依据自身实验过程，结合自身体会，总结至少给出 3 条心得体会。