

1. 概述

SH99F01 是一颗高集成度的电力线通信片上系统(SOC)，内部集成了单机器周期增强型 8051 内核的单片机和高性能双模电力线载波通信收发器(Powerline Transceiver 简称 PLT)。

单片机内部采用 FLASH 存储程序代码，可以多次烧写或者擦除，方便用户调试和在线更新。载波通信收发器采用中颖电子完全自主知识产权的高性能调制解调电路实现，可适应各种复杂的电力线信道环境，具有高灵敏度和强抗干扰性能，并且具有极大的灵活性和高性价比优势，可适应各种不同的电力线载波应用需求。

SH99F01 完全兼容中颖 8051 系列单片机应用开发环境，在线仿真和下载烧录可以通过 JET51 仿真器实现。该工具基于 Keil μ vision 集成开发环境，通过 JTAG 方式和单片机连接，实现全功能在线仿真和下载烧录功能。量产烧录采用中颖 ISP51 或 Pro-06 量产烧录器，提供 SH99F01 演示板供用户评估开发。

1.1. 封装

SH99F01 提供 TSSOP28 封装。

1.2. 演示板(Target Board)

SH99F01 提供了一套演示板供客户熟悉芯片功能，如下图。演示板除了提供单片机调试功能外，还提供电力线载波专用接口电路，用户在代码调试完成后可直接上电力线测试载波传输效果。

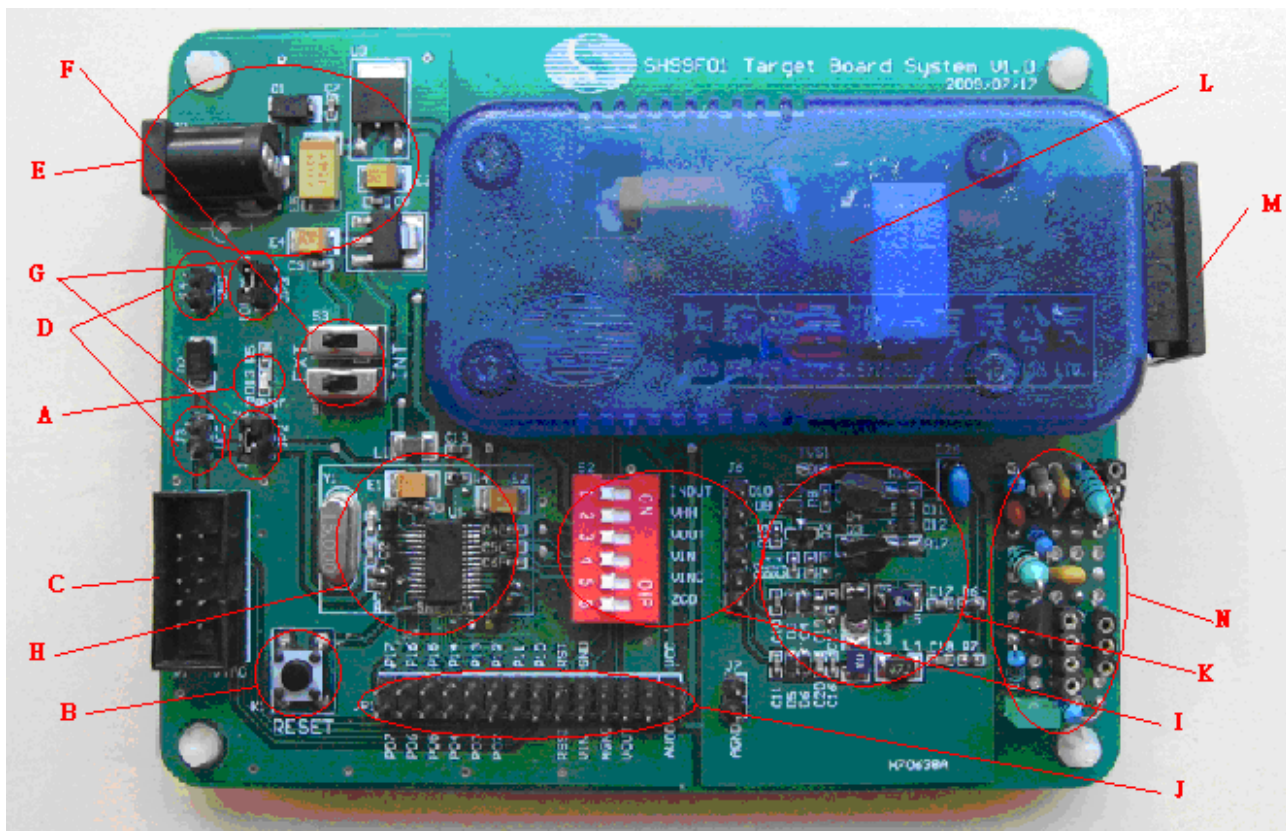


图1-1 SH99F01 演示板

板中各部分功能说明：

A. 电源灯。

B. 复位按键。

C. JTAG 仿真接口

JTAG 仿真接口和 JET51 仿真器一一对应，可用 10 芯扁平线直接连接。仿真时，演示板的供电电源可通过仿真接口的 VCC 管脚直接获得。

D. 外部供电接口 1(VCC/GND)

SH99F01 供电电源接口，通过此接口可提供可调的供电电源。

E. 外部供电接口 2(内正外负)

外接 9~15V 电源(典型 12V)，经过板上 U2 和 U3 稳压电路产生 3.3V 和 5.0V 电压作为 SH99F01 供电电源使用，通过 S1 和 S3 拨码开关可选择 VCC 以 3.3V 供电或 5.0V 供电。另外，Target Board 使用的载波功率驱动电路电源 VHH 直接由此接口提供。

注：演示板供电可由外部供电接口 1 或外部供电接口 2 或仿真接口提供，无论哪种方式，在外部供电接口 1 处电压即为演示板供电电压。

F. 演示板供电电压选择。



全部拨到左边(EXT 侧), 则 VCC 为 3.3V 供电, 同时 SH99F01 的模拟供电 AVDD 也通过 VCC 提供; 都拨到右边(INT 侧)则 VCC 为 5.0V 供电, AVDD 通过芯片内建 3.3V 稳压源供电 (需使能 OP_REG33 代码选项)。

注: 要求 S1 和 S3 两个开关必须同时拨到左边或右边, 不允许一左一右配置。

G. 电流测试接口。可分别测量芯片 VCC 电流和 AVDD 电流。

H. SH99F01 最小应用电路。U1 为目标芯片 SH99F01, TSSOP28 封装。

I. 模拟信号隔离开关和测试点。

S2 用于隔离 SH99F01 和后端 Analog Interface 电路之间连接, J6 为 Analog Interface 信号接口, 分别为: 耦合前端信号 INOUT, 驱动电源 VHH, 信号输出 VOUT, 信号输入 VIN, 信号输入第二通道 VINCOM, 过零检测脉冲输入 ZCD。

J. 28 PIN 通用调试接口。

SH99F01 相应的 port 口拉出到此接口(部分 pin 未拉出), 供客户开发调试。

K. Analog Interface 电路。

包括发送端载波功率驱动电路和接收端两通道 LC 滤波电路, 有关驱动和滤波电路详细设计请参考“SH99F01_AN_PLT”。

L. Coupler Interface 电路。

包括高压隔离耦合, AC 过零检测, EFT 保护电路。此电路与整个演示板电路是隔离的。

注: 此区域为 AC 区域, 使用时请勿打开盒盖!

M. AC 信号接口。接专用 8 字尾电源线。

此 AC 接口只完成载波信号的传输, 不对演示板提供电源, 后者电源通过外部接口 1 和 2 提供。

N. 调试扩展区。

可供客户扩展硬件电路用于调试。



2. 仿真

2.1. 仿真软件的安装

安装步骤

1. 安装 Keilc uvision2 (V2.34 及以上版本) 或者 uvision3。
2. 安装中颖 8051 组件库 JET51_Keil_Vxxx.exe(其中 Vxxx 表示版本号)到 Keilc 的安装目录下 (例如: c:\keil\).
3. 将 JET51 仿真器通过 USB 连接入计算机, 此时, 红色电源灯和绿色 USB 灯会同时亮起, 说明组件库安装成功。

2.2. 库文件更新

中颖 8051 库文件中包含了各颗芯片的 Flash 烧写选项定义, 芯片头文件, 芯片描述等信息。由于中颖 8051 系列在不断增加和更新, 相应的库文件也需要及时更新。

安装步骤

1. 安装中颖 8051 库文件更新文件 Flash_LibUpdate_Vxxx.exe(其中 Vxxx 表示版本号)到 Keilc 的安装目录下 (例如: c:\keil\).
- 注: 为支持 SH99F01, 需把 Flash 库文件更新到 V2.0 以上。
2. 打开\uv2\SinoDevice\version.ini 文件, 对照 Version 确认安装成功。
 3. 如果安装了中颖 8051 其他开发工具, 如 ISP51, Pro-06, 则需各自安装库文件到各自目录下(例如: c:\Program Files\SinoWealth\Pro06\).

2.3. 硬件的安装

安装步骤

1. 将 SH99F01 演示板上的 JTAG 接口(VDD、TCK、TDI、TMS、TDO、GND)与 JET51 仿真器的 JTAG 仿真接口通过 10 芯扁平电缆相连即可。
2. 当使用 JET51 仿真、下载时, JTAG 仿真口如果有复接大电容电路, 需要与演示板上的外接线路断开以确保通讯正常。

注: 设计用户目标板时, 如果 4 个复用引脚需接大电容电路或其他复杂外部电路, 应将 JTAG 仿真口通过跳线与应用电路连接, 以便烧写或者仿真时可以断开跳线与仿真器连接。

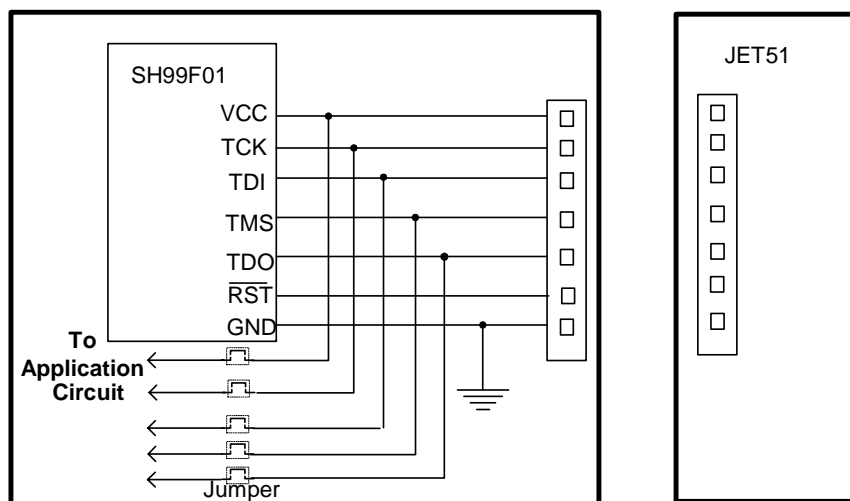


图2-1 SH99F01 JTAG 接口

3. 用户目标板电源上的负载电容不得超过 220 μ F，否则可能导致连接失败。

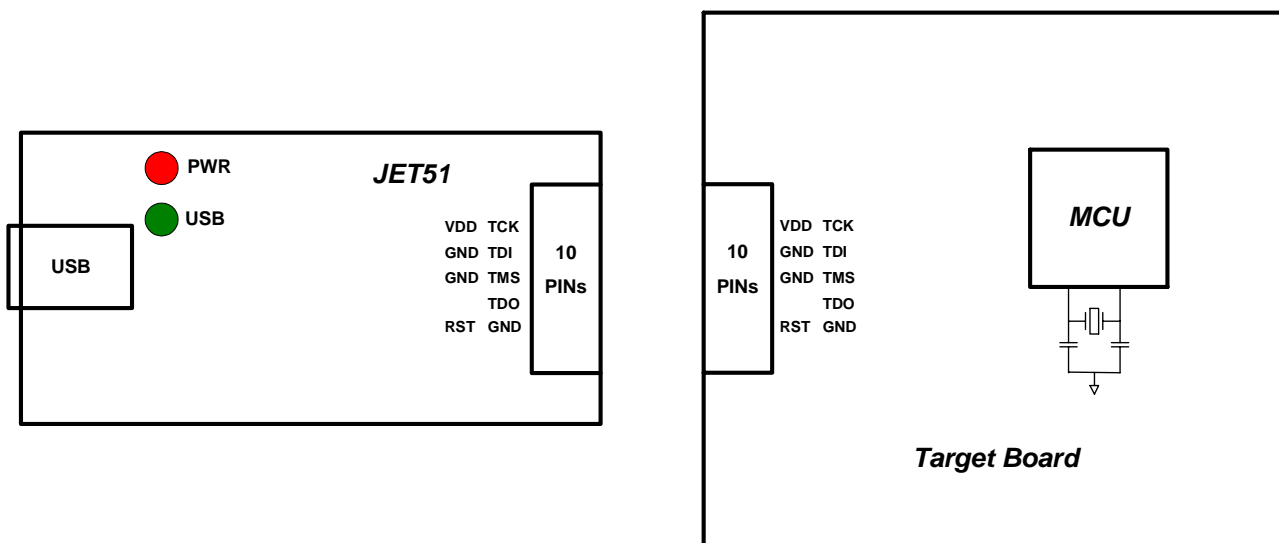


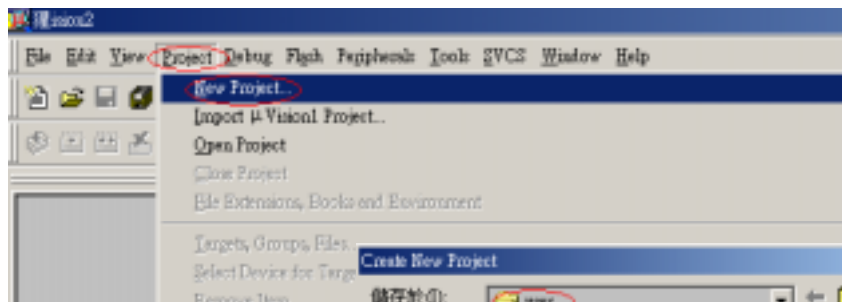
图2-2 JET51 与 Target Board 连接

注：早期版本的 JET51 和 JTAG 接口没有拉出 RST pin，在 Options for Target 中设置仿真选项时，在电源选择中不能选中“使用 RESET 引脚进入烧写模式”，详见 2.5.2.1 节。

2.4. 新建一个 project

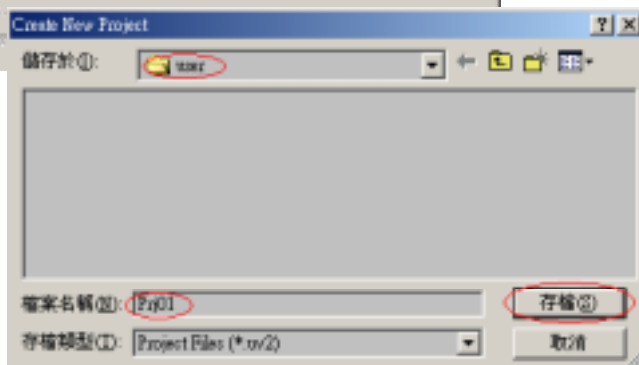
示例：用户的源程序在目录 User 下，文件名为 aa.asm。

现在需要对此建立一个新的 Project，操作如下：

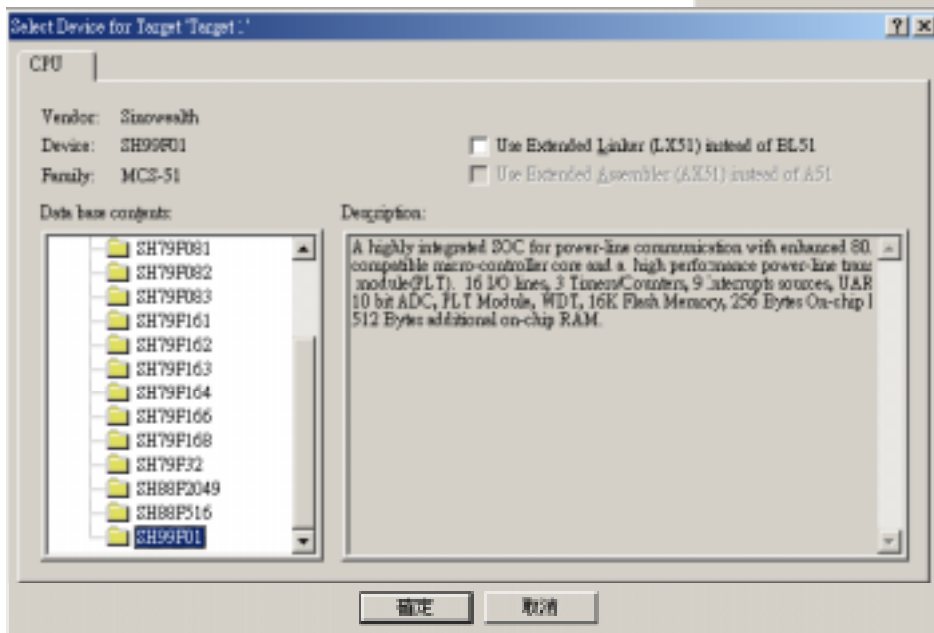
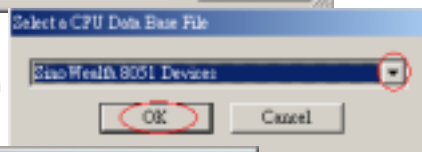


步骤 1: 进入仿真界面，
选择 Project→New Project。

步骤 2:
弹出右图窗口，进入源程序所在
目录 User，并输入所要建立项目
的项目名称(如 Proj01.Uv2)。



步骤 3:
点击后弹出对话框，选择“Sino Wealth
8051 Devices”，按“OK”按钮。



步骤 4:
选择芯片：SH99F01，
按“确认”按钮。

步骤 5:
选择“否”。

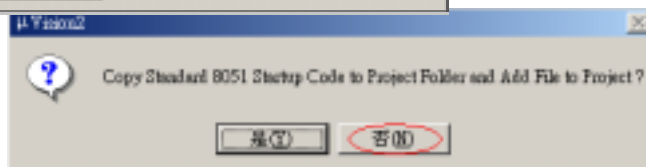
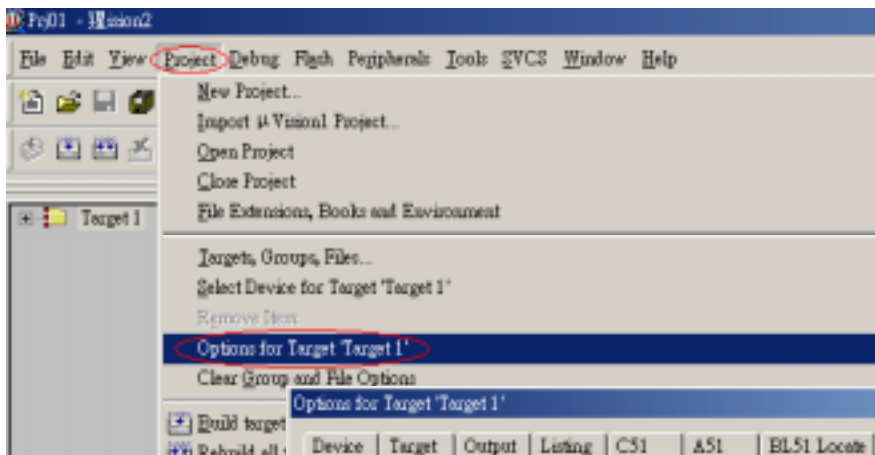


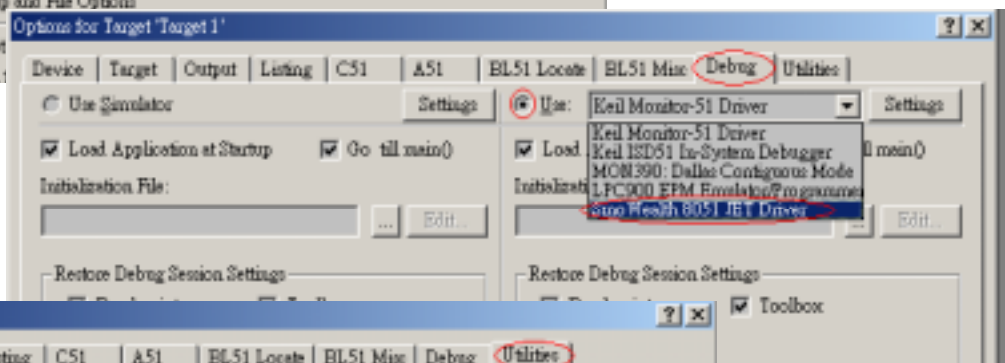
图2-3 操作示意图

至此，已完成建立一个空的项目。下面设置项目属性。

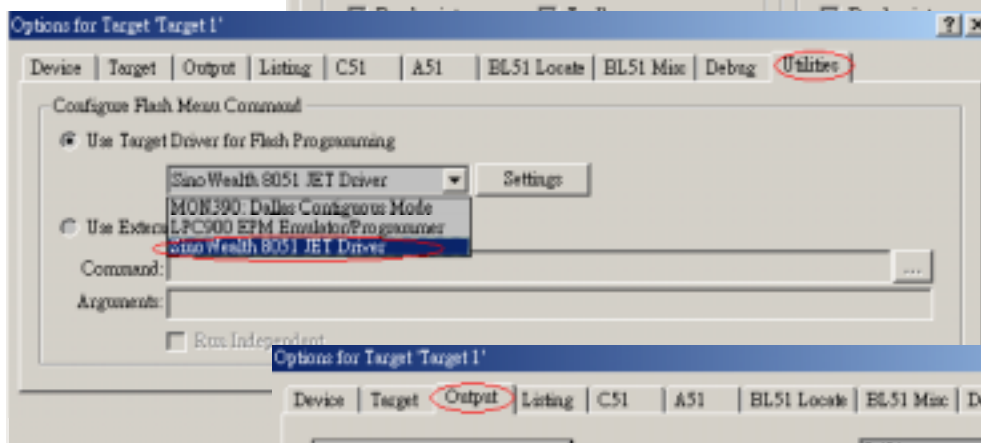


步骤 6:
选择 Project
→Options for Target 'Target1'。

步骤 7:
选择 Debug→Use
→Sino Wealth 8051
JET Driver。



步骤 8:
选择 Utilities
→Sino Wealth 8051
JET Driver
→确认。



步骤 9:
选择 Output
→选择“ Create HEX
File ” →确认。

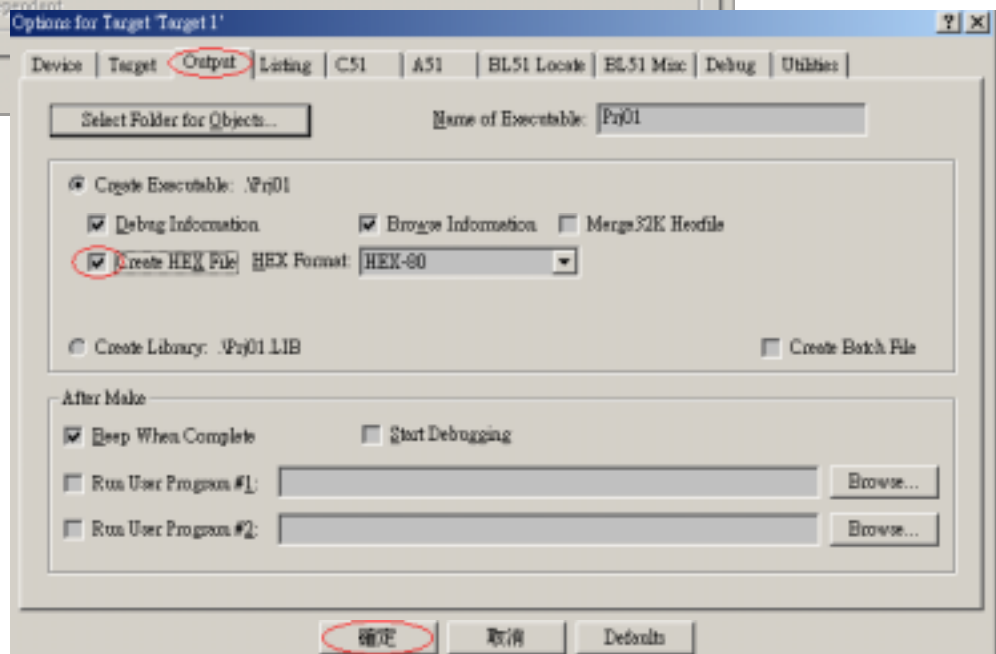
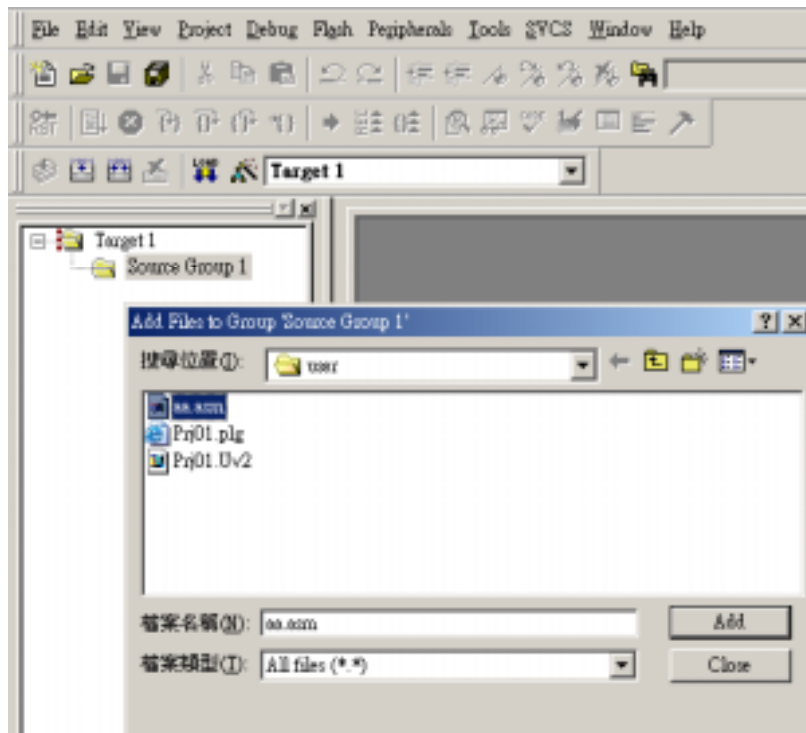


图2-4 操作示意图



至此，项目属性设置完毕。下面将已经编写好的程序添加到项目中。



步骤 10:

双击 Target1

→选择“Source Group1”，按右键

→选择目标目录，例如“User”

→选择目标文件，例如“aa.asm”

→点击“Add”按钮。

步骤 11:

关闭步骤 10 中“Add Files to Group ‘Source Group1’”窗口

→双击“Source Group1”

→双击“aa.asm”。

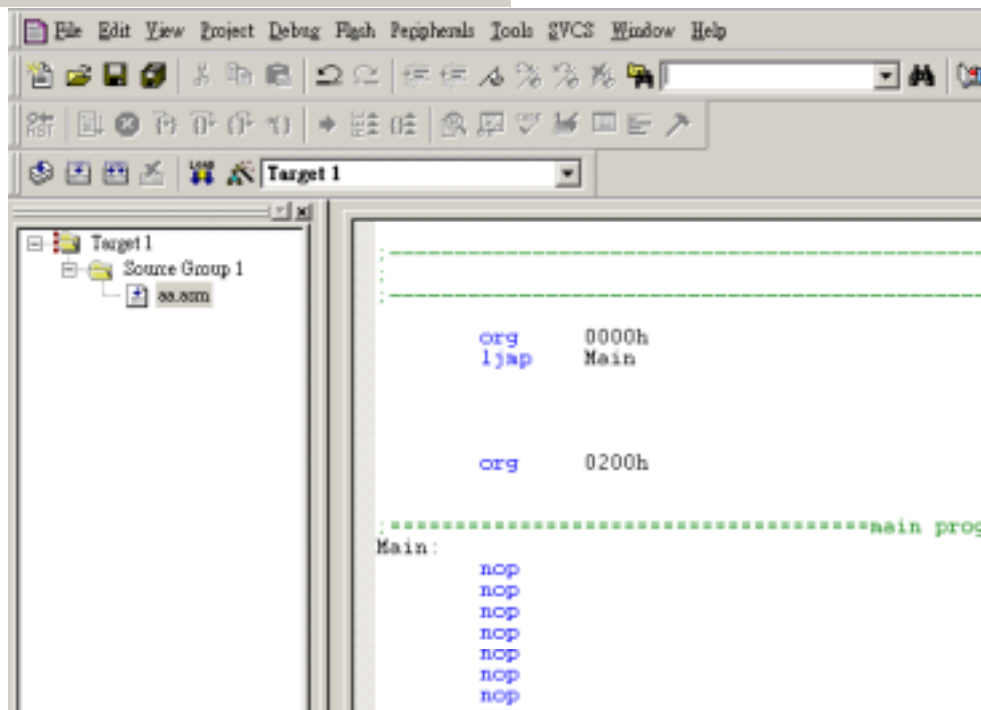
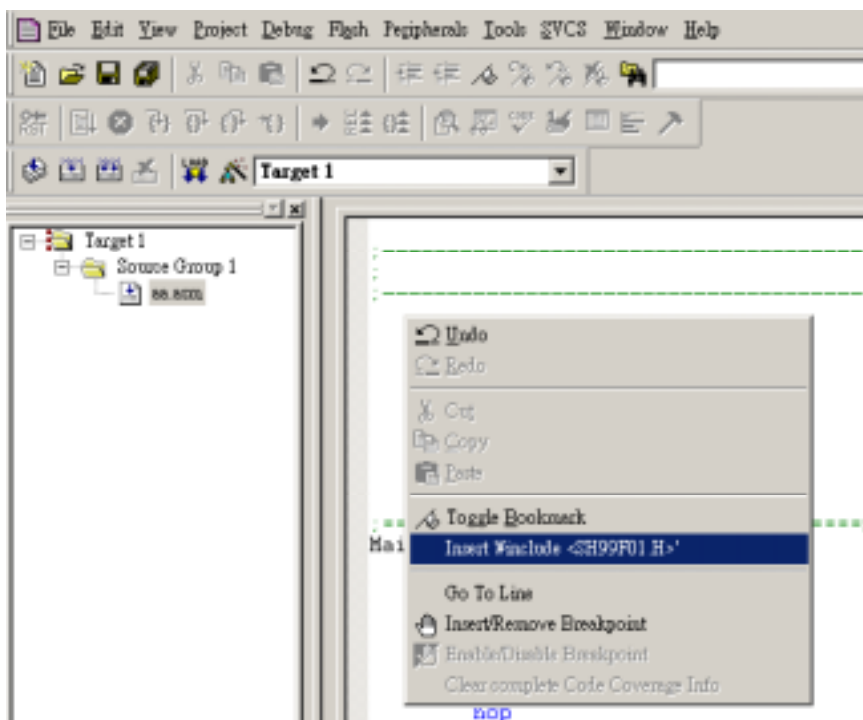


图2-5 操作示意图



步骤 12:

将鼠标移至程序开始处,按鼠标右键,选择头文件,头文件自动加入程序中。

头文件加入后,将鼠标移至头文件处,按鼠标右键,可以打开头文件。

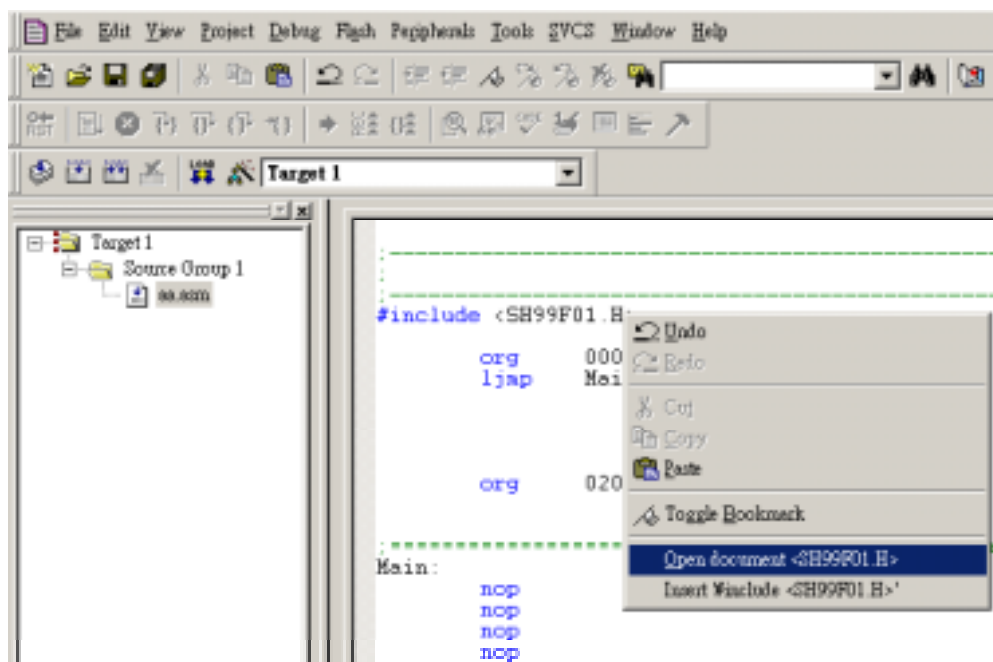


图2-6 操作示意图

2.5. 程序的编译及下载

2.5.1. 程序的编译



当程序编写完成后，点击菜单 Project\Build target(或编译按钮)，完成编译，生成 OBJ 文件和 HEX 文件。当存在编译错误(Error)，则不会生成 OBJ 文件和 HEX 文件。

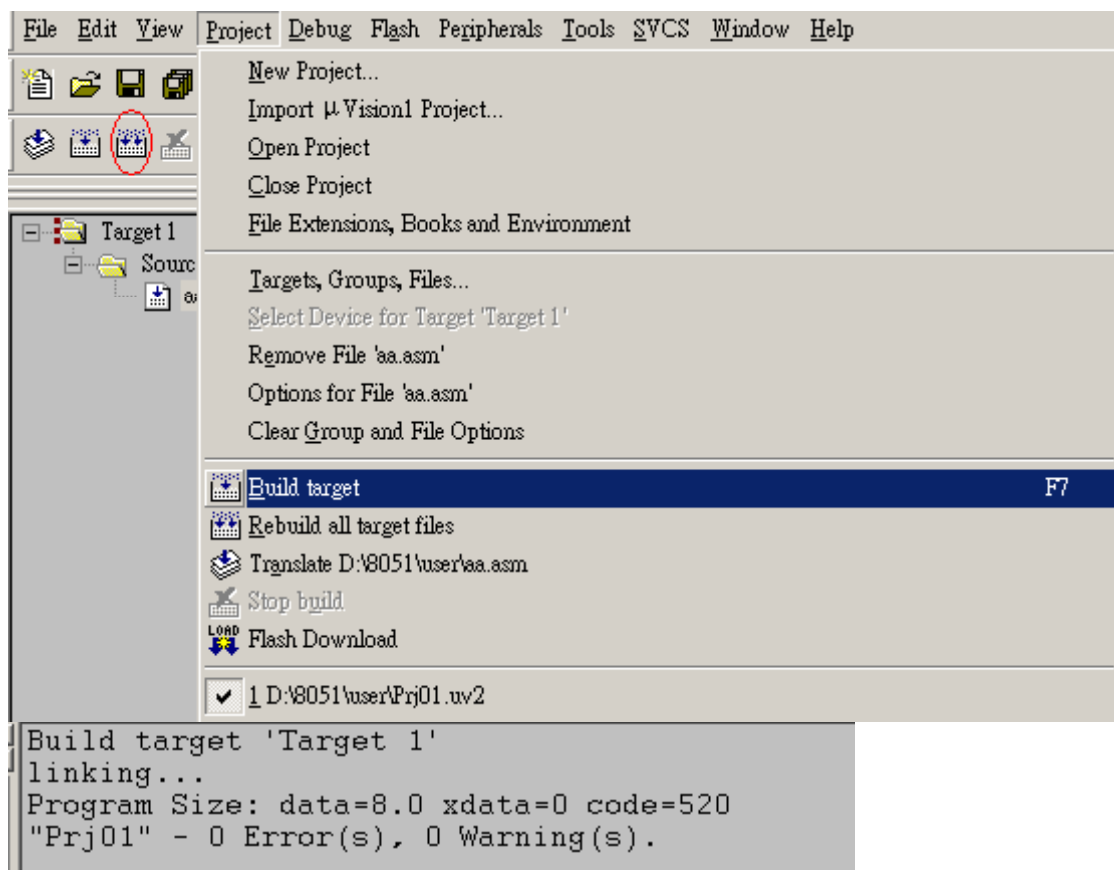


图2-7 操作示意图

注：对于使用汇编语言开发的 Project，由于此前已经在项目中包含了“SH99F01.H”，在编译之前要在 A51 中反选“Define 8051 SFR Names”选项(默认是选中的)，如下图所示，否则会出现部分 SFR 重复定义的错误，也可在程序中添加“\$NOMOD51”汇编控制语句。

对于使用 C 语言开发的 Project，因其使用 C51 编译器，无需上述操作。

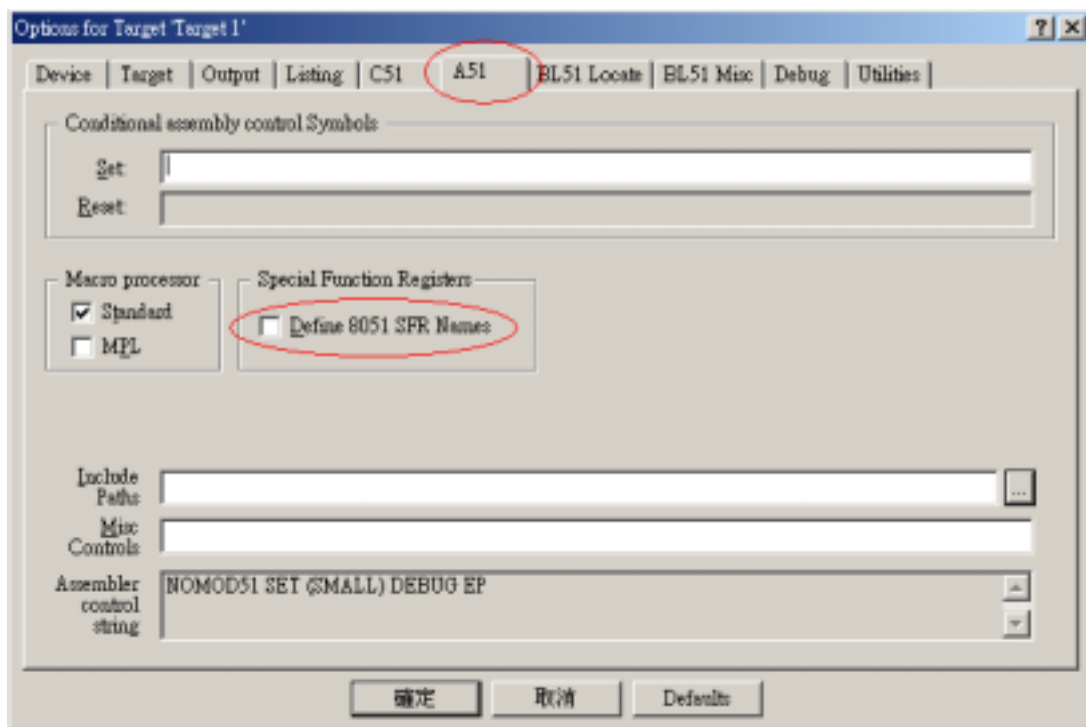
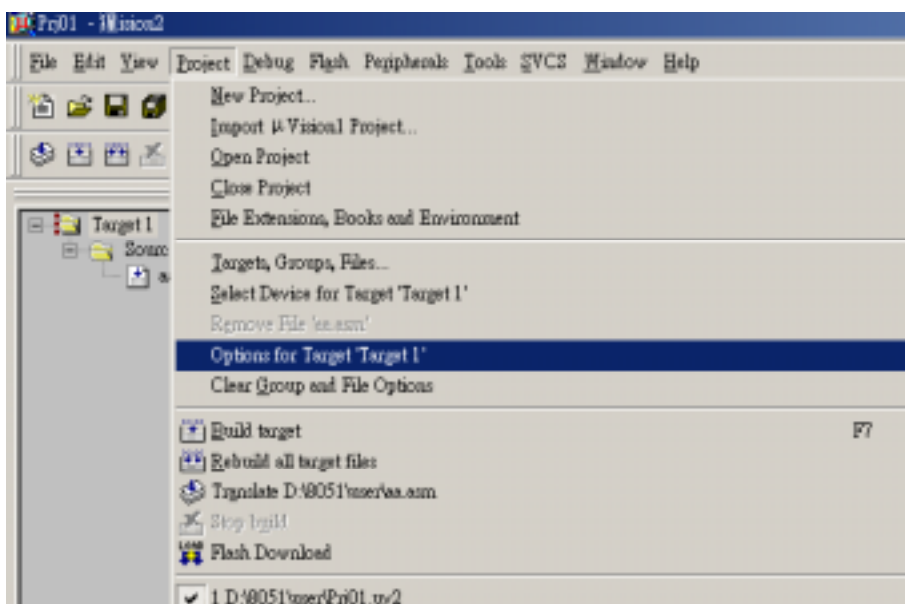


图2-8 操作示意图

2.5.2. 程序的下载

2.5.2.1. 下载前的选项设置

程序下载前的选项设置包括选用芯片设置、上电方式设置、CODE OPTION 设置等。所有设置系统会自动保存。之后再程序下载时，可以跳过此步骤。



步骤 1:
点击 Project
→Option for Target 'Target 1'
(或直接点击快捷按钮)。



图2-9 操作示意图

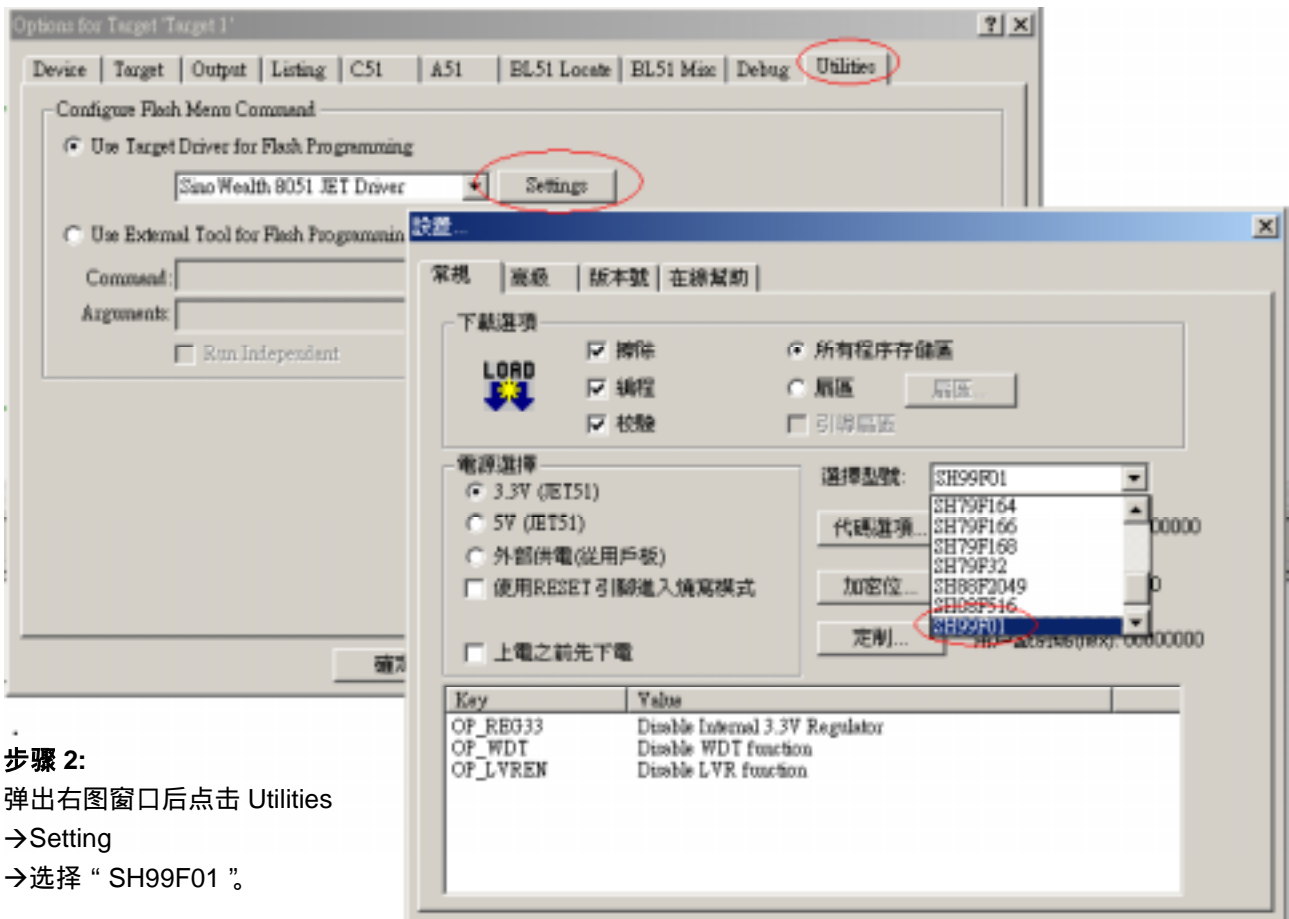
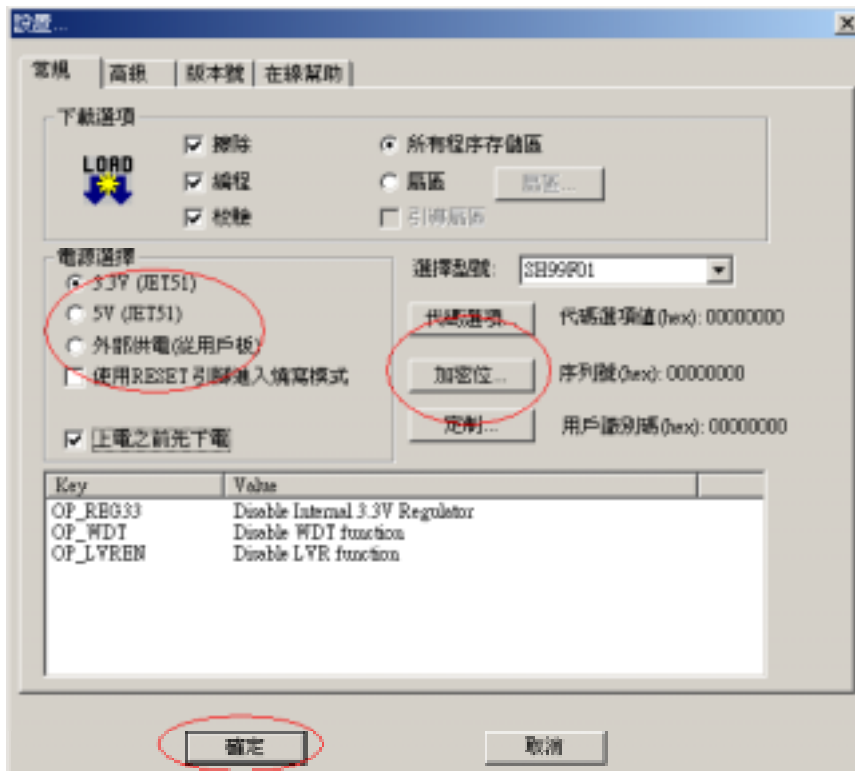


图2-10 操作示意图



步骤 3:

用户根据需要选择上电方式及 option 等选项，全部选择完成后点击“OK”按钮。

步骤 4:

按“确认”按钮完成设置。

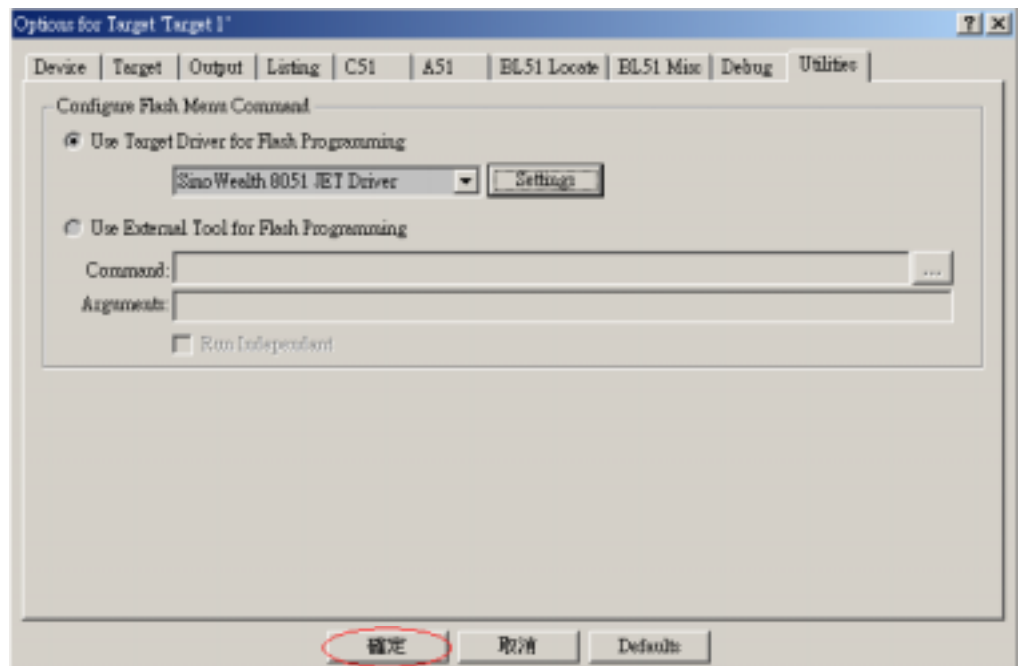


图2-11 操作示意图



注: JET51 仿真器提供了 2 种不同的上电方式, 在步骤 3 中设置。

1) 方式 1: JET51 仿真器供电

选择“3.3V(JET51)或 5V(JET51)”和“上电之前先下电”。其中,“上电之前先下电”是指“下电/上电后重进 JTAG 模式”,通常在仿真或者下载出错时选用此方式。

在程序下载和仿真的时候,用户目标板的供电由 JET51 仿真器提供。此时,用户目标板不需要外接任何电源,电源会通过 JET51 仿真器电源直接供给用户目标板。

如若用户目标板耗电小于 20mA,推荐用方式 1。

2) 方式 2: 外部供电

选择“外部供电(从用户板)”。

在程序下载和仿真的时候,用户目标板通过外接电源供电。此时,上电顺序有严格要求:

步骤 1: 确认用户目标板处于断电状态。

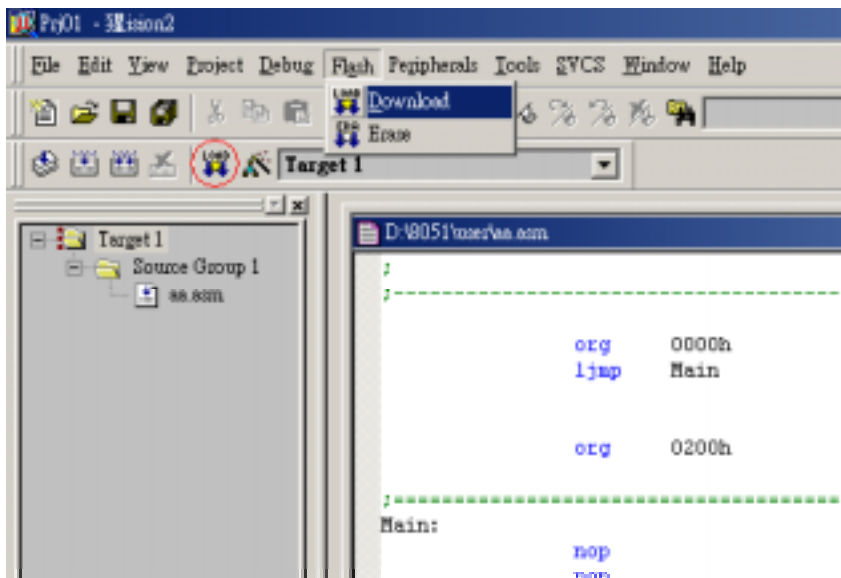
步骤 2: 连接 JET51 仿真器和用户目标板。

步骤 3: JET51 仿真器和 PC 通过 USB 连接,红色电源灯和绿色 USB 灯全部点亮。

步骤 4: 用户目标板接入外部电源。

此后调试可以反复下载和仿真,无须下电及重新连接。

2.5.2.2. 程序下载



按“Download”快捷按钮
或点击 Flash→Download。

Download成功

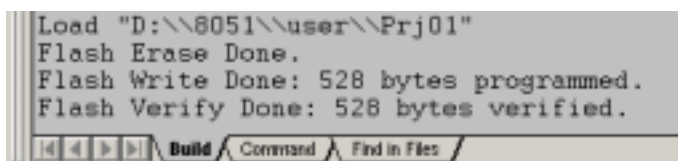


图2-12 操作示意图



2.6. 程序仿真

当程序已经下载到芯片中，点击菜单 Debug\ Start/Stop Debug Session 或快捷按钮进入程序仿真模式。

当 JTAG 完全连接上以后，会有黄色箭头指向 0000H 处，同时左边寄存器窗口 SP 的值为 07H，表示成功进入调试模式可以进行仿真了。用户可以通过选择 Debug 菜单下的命令来执行 Step、Step Over、Run、Stop 等操作，也可以执行增加断点、减少断点的操作。

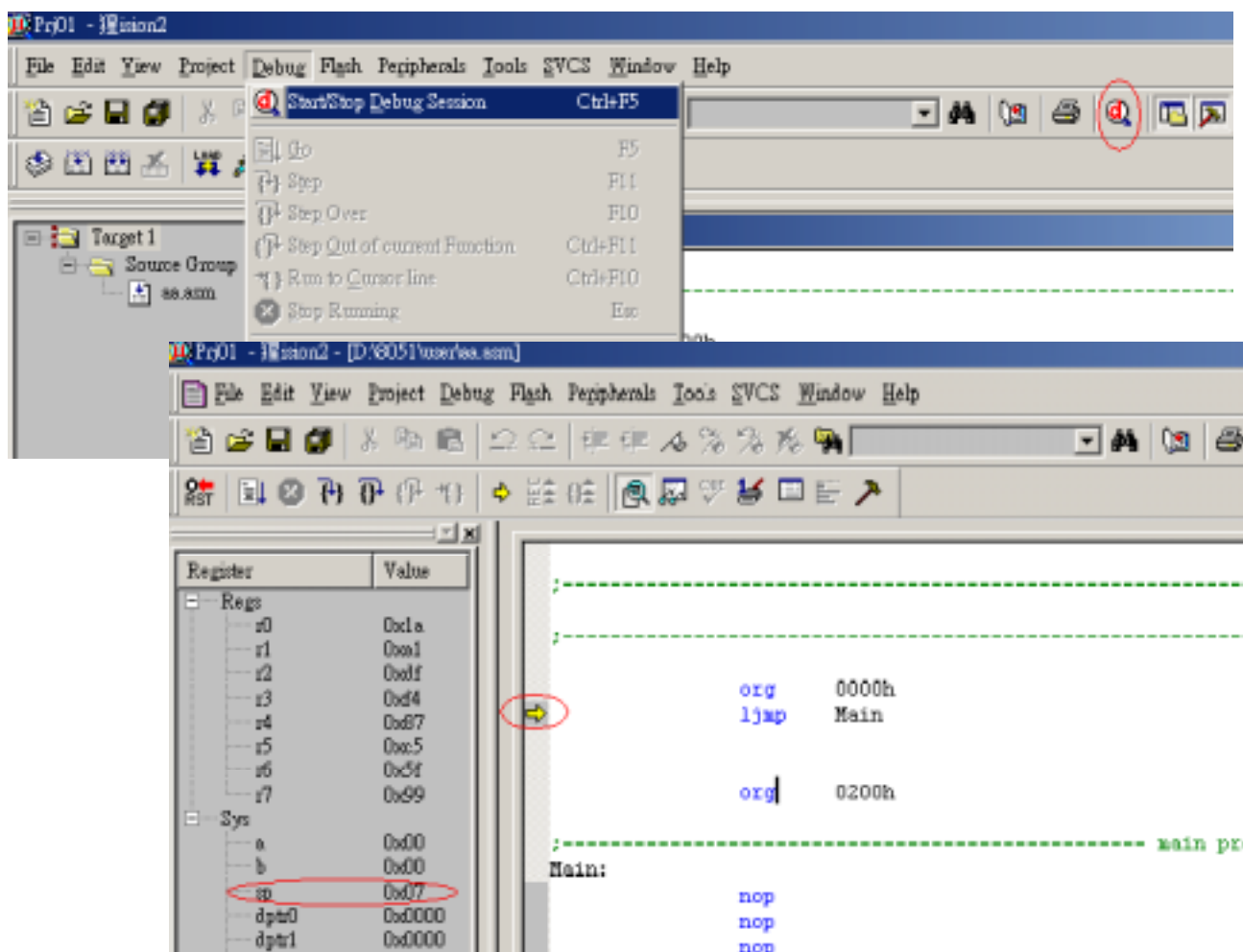


图2-13 操作示意图

注意:

- 1) 断点为将该指令执行完以后，再停止。
- 2) 执行 Step Over 指令，在执行到 ACALL、LCALL 等语句时，会将 ACALL、LCALL 下一条语句执行完才停止。



- 3) 仿真器只支持 7 个断点。若超过 7 个断点，则只支持最后设置的 7 个断点。以前的断点将被忽略。
- 4) 程序仿真无误后，断开外加电源，拔掉 JTAG 连线。
- 5) 用户目标板接入外部电源，开始独立运行(脱机模式)。



3. 应用注意事项

3.1. 和传统 8051 差异

SH99F01 是一颗采用增强型 8051 内核的单片机，比标准 8051 扩展了很多功能,考虑到和标准 8051 的兼容性(尤其在用 keil c 来开发程式时)，建议如下：

3.1.1 SH99F01 机器周期为 1 个振荡器周期，而传统型 51 内核机器周期为 12 个振荡器周期。其指令执行效率约为同频率的传统 8051 的 8~10 倍。详细指令执行时间参见 DATASHEET。

3.1.2 SH99F01 提供增强型的乘/除法指令，支持 16bit × 8bit 和 16bit ÷ 8bit 运算。

MUL(INSCON.2)和 DIV(INSCON.3)位控制选择执行 8 位或 16 位模式乘/除法运算，建议操作完成后将此两位清 0。

3.1.3 SH99F01 提供双 DPTR 指针，用户在使用查表，数据搬移等操作时会更方便。

DPS(INSCON.0)位控制选择使用 DPTR0 或 DPTR1 指针，建议操作完成后将 DPS 位清 0。

3.1.4 SH99F01 扩展了间接寻址指令 MOVX @Ri,A 或 MOVX A, @Ri 的高 8 位地址寄存器 XPAGE，可以访问到超过 256 BYTE 的外部 RAM，提高访问效率。建议操作完成后将 XPAGE 清 0。

3.1.5 SH99F01 复位脚采用低电平复位，不同于传统 8051 的高电平复位。

3.1.6 SH99F01 由 FAC(FLASHCON.0)位控制选择对程序存储区或客户信息区/类 EEPROM 区的访问，建议操作完成后将此位清 0。

其它功能上的差异请参见 DATASHEET。

3.2. FLASH ROM

3.2.1 SH99F01 提供 16K BYTE FLASH ROM，每 1K BYTE 为 1 个扇区(SECTOR)，每两个扇区可分别进行加密。如果没有加密，每个扇区可分别单独进行编程和擦除。整体擦除(MASS ERASE)可以擦除所有扇区。

3.2.2 SH99F01 的全部 16K BYTE 都可用于存储程序和数据，另外提供 2K BYTE 的客户信息区 / 类 EEPROM 块区用于存放用户数据，后者 1 个扇区(SECTOR)为 256 BYTE。

3.2.3 存放在 FLASH ROM 中的程序可以对不在同一扇区的 FLASH ROM 编程(SSP 模式)，因此可以当作 EEROM 来使用。扇区中的数据从 0 改为 1 时，可以直接烧写，但从 1 改为 0 时，必需通过扇区擦除后才能实现。对 2K BYTE 的客户信息区 / 类 EEPROM 块区也是同样操作。

FLASH ROM 每一个 BYTE 的烧写时间与系统时钟无关,但不会大于 50us，擦除 1 个扇区 (SECTOR)的时间不大于 40ms。

3.2.4 SH99F01 支持 ICP 功能，即在客户应用电路板完成后，在客户电路板上通过烧写器烧写程式代码。但必须注意，P1.0~P1.3 四个烧写口不能有大电容负载，VDD 管脚上的负载电容不得超过 220μF，否则可能导致烧写失败。同时注意，CREG 脚需要外接 47uF 电容。

3.2.5 FLASH 烧写/擦除步骤

3.2.5.1 烧写



1) 关闭中断。在整个烧写期间无法响应中断，但中断申请标志位照常会置 1。因此，如果烧写期间来了中断，烧写完成后，打开中断允许位，中断服务程序仍将执行。

2) SH99F01 无需设置烧写时钟寄存器。如果编程地址在程序存储区，将 FAC 位清 0；如果编程地址在客户信息区或类 EEPROM 区，将 FAC 位置 1。

3) 设置 XPAGE、IB_OFFSET 寄存器。此 2 个寄存器对应烧写地址。其中 IB_OFFSET 对应地址低位，XPAGE 对应地址高位。

4) 将准备填充值写入 IB_DATA。此寄存器中填写的值将被写入到 FLASH 中指定地址。

5) 按照顺序设置 IB_CON1~5，其中数值如下所列：

IB_CON1 → #6EH ;表示烧写。

IB_CON2 → #05H

IB_CON3 → #0AH

IB_CON4 → #09H

IB_CON5 → #06H

注意顺序不能有错，数值也必须正确。烧写完成后这 5 个寄存器硬件自动清零，防止误写。因此下次写操作必须重新填充这 5 个寄存器。

6) 软件写入 IB_CON5 后启动硬件写操作，CPU 将进入 IDLE 模式。烧写完成后自动唤醒。此步骤由 CPU 自动完成，用户无须干预。

7) 软件设置完上述寄存器后插入 4 个 NOP 指令，保证烧写完成后 CPU 从 IDLE 模式可靠唤醒。

8) 如需继续写入数据，跳转至第 3 步。

9) 将 XPAGE 寄存器清除为零；根据后续程序需要置 1 或者清 0 FAC 位（FLASHCON.0）。

10) 恢复中断。

11) 程序示例：注意示例中各数据均为示意性数据，应用时请根据具体情况设置。

CLR EA ;步骤 1

ORL FLASHCON,#01H ;步骤 2

(或 ANL FLASHCON,#0FEH)

MOV XPAGE,#18H ;步骤 3

MOV IB_OFFSET,#00H

MOV IB_DATA,#05H ;步骤 4

MOV IB_CON1,#6EH ;步骤 5

MOV IB_CON2,#05H

MOV IB_CON3,#0AH

MOV IB_CON4,#09H

MOV IB_CON5,#06H ;步骤 6: CPU 进行烧写，自动进入 IDLE 模式，完成后自

动唤醒

NOP ;步骤 7

NOP



NOP

NOP

;步骤 8: 如果还需要继续写入数据, 重复上述步骤 3~8

MOV XPAGE,#00H ;步骤 9

ANL FLASHCON,#0FEH

(或 ORL FLASHCON,#01H)

SETB EA ;步骤 10

3.2.5.2 擦除

1) 关闭中断。在整个烧写期间无法响应中断, 但中断申请标志位照常会置 1。因此, 如果擦除期间来了中断, 擦除完成后, 打开中断允许位, 中断服务程序仍将执行。

2) SH99F01 无需设置擦除时钟寄存器。如果编程地址在程序存储区, 将 FAC 位清 0; 如果编程地址在客户信息区或类 EEPROM 区, 将 FAC 位置 1。

3) 根据需要擦除的扇区设置 XPAGE。1K BYTE 为一个扇区, SH99F01 ROM 为 16K BYTE, XPAGE 的 bit5~2 对应扇区号。

4) 按照顺序设置 IB_CON1~5。

IB_CON1----→ #0E6H ;表示擦除。

IB_CON2 ---→ #05H

IB_CON3----→ #0AH

IB_CON4 ----→ #09H

IB_CON5 ----→ #06H

注意顺序不能有错, 数值也必须正确。擦除完成后, 这 5 个寄存器硬件自动清零, 防止误擦除。因此下次擦除操作必须重新填充这 5 个寄存器。

5) 软件写入 IB_CON5 后启动硬件擦除操作, CPU 将进入 IDLE 模式。擦除完成后自动唤醒。此步骤由 CPU 自动完成, 用户无须干预。

6) 软件设置完上述寄存器后插入 4 个 NOP 指令, 保证擦除完成后 CPU 从 IDLE 模式可靠唤醒。

7) 如果还需要继续擦除扇区, 跳转至第 3 步。

8) 将 XPAGE 寄存器清除为零; 根据后续程序需要置 1 或者清 0 FAC 位 (FLASHCON.0)。

9) 恢复中断。

10) 程序示例: 注意示例中各数据均为示意性数据, 应用时请根据具体情况设置。

CLR EA ;步骤 1

ORL FLASHCON,#01H ;步骤 2

(或 ANL FLASHCON,#0FEH)

MOV XPAGE,#18H ;步骤 3

MOV IB_CON1,#0E6H ;步骤 4

MOV IB_CON2,#05H



```
MOV    IB_CON3,#0AH
MOV    IB_CON4,#09H
MOV    IB_CON5,#06H
;步骤 5: CPU 进行扇区擦除，自动进入 IDLE 模式，完成后自动唤醒
NOP                                ;步骤 6
NOP
NOP
NOP
;步骤 7: 如果需要继续写入数据，重复上述步骤
MOV    XPAGE,#00H                ;步骤 8
ANL     FLASHCON,#0FEH
      (或 ORL FLASHCON,#01H)
SETB   EA                        ;步骤 9
```

3.3. RAM

3.3.1 SH99F01 有 256 BYTE 内部 RAM，512 BYTE 外部 RAM。

3.3.2 内部 RAM 地址范围为 0~FFH，其中 0~7FH 既可以直接寻址，也可以间接寻址。80~FFH 只能用间接寻址。若用直接寻址访问此地址范围，访问到的是系统寄存器。

3.3.3 外部 RAM 地址范围为 0~1FFH。传统的 8051 用 MOVX A, @Ri (i=0 或 1) 只能访问 0~FFH 地址范围内的外部 RAM。在 SH99F01 中，扩充了 XPAGE 寄存器(\$F7)用作高 8 位地址。用此寄存器配合可以使 MOVX A, @Ri 指令可以访问超过 0~FFH 地址范围的外部 RAM。

3.3.4 外部 RAM 也可以使用 MOVX A, @DPTR 或 MOVX @DPTR, A 来访问外部 512 字节 RAM。

3.4. Idle & Power-down

程序编写时注意 IDLE 或 Power-down 指令之后务必加至少 3 条 NOP 指令，以保证系统能够稳定进入 IDLE 或 Power-down 模式。

```
MOV    8EH, #55H
ORL     PCON, #01H                ;idle mode (or #02H: Power-down mode)
NOP                                ;必须添加
NOP                                ;必须添加
NOP                                ;必须添加
.....                            ;后续程序。调试程序时如果需要在 IDLE 唤醒处设置断
```

点

;断点位置要放在 3 个 NOP 之后。



3.5. 中断

3.5.1. OVL 中断

当 CPU 取址后译码发现操作码为 A5H 时(要求填充),或者 PC 指针超出寻址范围,CPU 会认为程序已跑飞,会自动产生一个不可屏蔽中断,在中断服务程序中,建议写法为:

OVL_NMI_SERVICE:

```
MOV DPTR, #PROTECT_CODE_ENTRANCE ;设置跳转地址
POP A
POP A
PUSH DPL ;用户保护程序入口地址压栈
PUSH DPH
RETI ;以 RETI 退出中断,避免对后续中断响应产生影响
```

其中, #PROTECT_CODE_ENTRANCE 是出错后,用户希望所跳转到的程序入口地址。

注意:

3.5.1.1 在 OVL 中断服务程序中绝对不能用 JMP 指令替换 RETI 指令直接跳转到出错后处理程序入口地址,这样 CPU 会认为一直处在 OVL 中断服务程序中,由于 OVL 中断程序优先级最高,会屏蔽掉所有中断。

3.5.1.2 若希望程序跑飞后能产生 OVL 中断,建议将程序中所有未用到的 ROM 地址全部用 A5H 填充,以进一步提高 CPU 的抗干扰能力。

3.5.1.3 若程序 PC 超过 3FF8H(16376 Bytes),则同样会产生 OVL 中断。

注: 由于 SH99F01 指令系统为流水线结构,PC 在到达 3FFFH 时预取指地址已经超出 3FFFH,从而引发 OVL 中断,因此需限制 PC 在 3FFFH 以内的一个地址,考虑到不同指令,应限制在 3FF8H 以内。

技巧: 如果开启 Watch Dog,则可在 OVL 中断服务程序中写一个死循环,通过 Watch Dog 复位退出异常状态,重新装载系统。

附: 在程序中用 A5H 填充未用到 ROM 地址的方法:

首先定义宏:

```
Fill_A5H MACRO N
                REPT      N
                DB        0A5H
                ENDM
ENDM
```

然后在没有用到的 ROM 空间调用宏,比如 1000H 到 11FFH 没有用到,可以用如下方法填充:

```
CSEG AT 1000H
Fill_A5H 512
```



3.5.2. Port 口外部中断

3.5.2.1 由于 I/O 复用功能比较多,只有当相应的 PORT 口设置为 I/O 功能而且设置为输入状态,才有可能产生外部中断。当 EXx(x=0-1)位置 1 时内部上拉功能自动开启。注意,内部上拉电阻大概在 30kΩ 左右,如果刚开始 PORT 口输出低电平,中断初始化程序中将 PORT 口设为输入,然后马上开启 PORT 口中断控制位,而且设置为低电平触发,由于引脚上有电容负载,内部上拉比较慢,内部中断低电平检测电路可能会检测到此低电平,产生一次中断请求。因此,建议此情况下,内部上拉打开后,等待一段时间(比如 5 个 NOP 指令)后,再打开中断允许控制位。

3.5.2.2 当外部中断口进行第二功能切换时,应先将 EXy(y=0-1)位清 0,屏蔽外部中断功能,否则可能会因端口的功能切换而响应一次外部中断。

3.5.2.3 Port 口中断设置请按以下步骤进行(以 INT0/P1.2 口为例):

- 1) 将 INT0 口设置为输入状态,选择内部上拉;
- 2) 5 个 NOP 指令(建议);
- 3) 设置 INT0 口触发方式(\$88.0),清 INT0 中断请求标志(\$88.1);
- 4) EX0\$A8.0)位置 1,设置中断总控制位(\$A8.7)为允许状态。

3.5.2.4 IDLE、Power-down 唤醒(以 INT0/P1.2 口为例):

按 3.5.2.3 设置好 INT0 中断口之后,进入 IDLE 或 Power-down,当 INT0 口有下降沿或低电平(根据 INT0 触发方式的设定)产生时,即可唤醒 IDLE 或 Power-down。

唤醒后,程序先执行 INT0 中断服务子程序,并以 RETI 返回。

之后再执行 IDLE 或 Power-down 指令后面 3 个 NOP 之后的指令。

3.6. ADC

3.6.1 SH99F01 内部集成 10bit 分辨率 ADC,ADC 时钟来源可由寄存器 TADC[2:0]设置为系统时钟的 2 分频、4 分频、6 分频、8 分频、12 分频、16 分频、24 分频、32 分频。不论使用多少频率的系统时钟,都必须保证 ADC 时钟周期大于或等于 1us,否则会影响精度。如果实际使用中只需要 8bit 精度,ADC 时钟周期可以更短一些。

3.6.2 完成一次 A/D 转换需要的时间分为 2 部分:首先是采样时间,可由寄存器 TS[3:0]设置,最低为 2 个 ADC 时钟周期;然后是转换时间,固定为 12 个 ADC 时钟周期,因此完成一次 A/D 转换至少要 14us。外部仿真输入信号必须保证在采样期间固定,而在转换期间输入信号发生变化不会影响到最终 A/D 转换结果。

3.6.3 ADC 内部集成了数字比较器功能,可以将数字比较器参考值写入数值寄存器。将此功能打开(ADCON=101X XXX1B)后,ADC 一直处在转换状态,但每次转换完后不会产生中断标志。如果发现转换结果比参考值大,则会产生中断标志。因为占用了内部 A/D 资源,因此 ADC 功能和数字比较器功能无法同时使用。



3.7. 看门狗

3.7.1 内置看门狗电路，溢出时间从 1ms 到 4s 可调整。如果在设定时间内没有操作看门狗寄存器(读或写)，看门狗电路会产生复位信号，同时将看门狗溢出标志位置 1。

注：Watch Dog 定时会有一定偏差，需留有一定裕量。

3.7.2 看门狗电路耗电约 1~2uA，如果 option 选项允许，即使进入 Power-down 模式依然有效。因此看门狗会唤醒 Power-down 模式，唤醒 Power-down 模式后同样将看门狗溢出标志置 1。

3.7.3 看门狗电路可通过选项选择有效或无效。

3.7.4 JTAG 仿真模式下 Watch Dog 正常起作用。

3.8. UART

3.8.1 由于 TXD/RXD 引脚与 I/O 功能复用，如果 REN 控制位设为 1，则 RXD 引脚自动设为输入，内部自动拉高。而 TXD 引脚只有在 UART 发送即软件向 SBUF 寄存器写入数据时才会作为 TXD 引脚使用，发送完成后又恢复为 I/O 功能。为了避免接收端误判，在 UART 发送初始化程序中，应该将相应 I/O 状态设为输出高电平。

3.8.2 UART 波特率发生器可以采用 Timer1，也可以采用 Timer2，由 T2CON 寄存器中 TCLK 和 RCLK 控制位来控制：若为 0，则采用 Timer1，为 1 则选择 Timer2。由此可见，UART 发送和接收波特率可以设置为不同值。

3.8.3 UART 接收数据的波特率最大允许偏差为： $\pm 4\%$ 。

3.9. 应用电路

高压应用和低压应用：

3.9.1.1 代码选项的设置

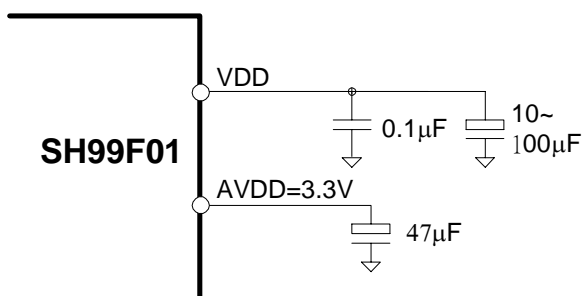
SH99F01 有若干代码选项供用户使用，其中 OP_REG33 与用户系统设计关系较大，见下表：

表3-1 OP_REG33

OP_REG33	内部 3.3V 稳压源	AVDD 引脚	AVDD 电压	应用场合
0	关	接 VDD	VDD	VDD =3.0V~3.6V
1	开	接 47uF 电容到 AGND	3.3V	VDD=3.6V~5.5V

3.9.1.2 VDD 和 AVDD

不同的应用场合 VDD 和 AVDD 选择短接或断开(VDD 上电解电容用于抗干扰)，见下图：



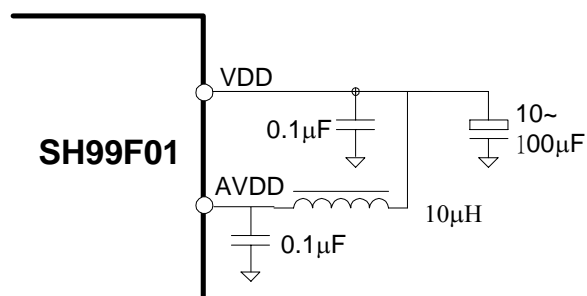
高压应用 (VDD=3.6V~5.5V)

a. 设置代码选项：

设置 OP_REG33 = 1

b. AVDD、VDD 管脚不短接

c. AVDD 电源 = 3.3V



低压应用 (VDD=3.0V~3.6V)

a. 设置代码选项：

设置 OP_REG33 = 0

b. AVDD、VDD 管脚通过低 ESR 电感短接

c. AVDD 电源 = VDD





5. 应用指南版本历史

版本号	版本记录	时间
V1.0	Original	2009-12-31