STC89C51RC / RD+ 系列单片机中文指南

--- 高可靠
--- 超低价
--- 低功耗
--- 无法解密

STC89C51RC, STC89LE51RC
STC89C52RC, STC89LE52RC
STC89C53RC, STC89LE53RC

STC89C54RD+, STC89LE54RD+
STC89C58RD+, STC89LE58RD+
STC89C516RD+, STC89LE516RD+

附录 A:
为什么少数用户的普通 8051 程序烧录后，不能运行

附录 B: STC89LE516AD，STC89LE516X2

附录 C: STC89C51RC / RD+ 系列单片机 ISP (DIY)

附录 D: ISP Demo(演示版)软件(*.hex)及通信协议

附录E: 如何实现运行中自定义下载,无仿真器时方便调试

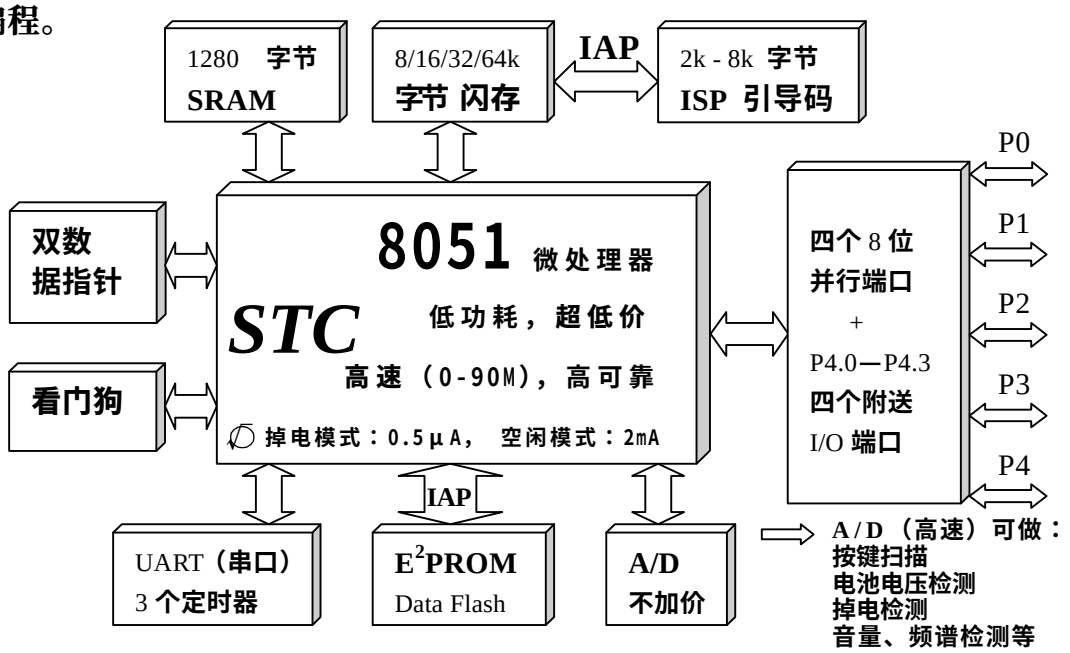
附录 F: Keil C51 高级语言编程的软件如何减少代码长度

国内技术支援：宏晶科技（深圳）
www.MCU-Memory.com support@dsp-memory.com
Update date: 2005-2-16

本应用技术手册是针对有一定 8051 系列（MCS-51）单片机编程基础的用户编写的。

选用 STC 单片机的理由：降低成本，提升性能，原有程序直接使用,硬件无需改动。

STC 公司鼓励您放心大胆选用 PLCC，PQFP 小型封装, 3.3V 工作电压单片机，使您的产品更小，更轻，功耗更低。如果相关新增功能没有用到，则不需看相应部分。用 STC 提供的 STC-ISP.exe 工具将您原有的代码下载进 STC 相关的单片机即可,或用通用编程器编程。



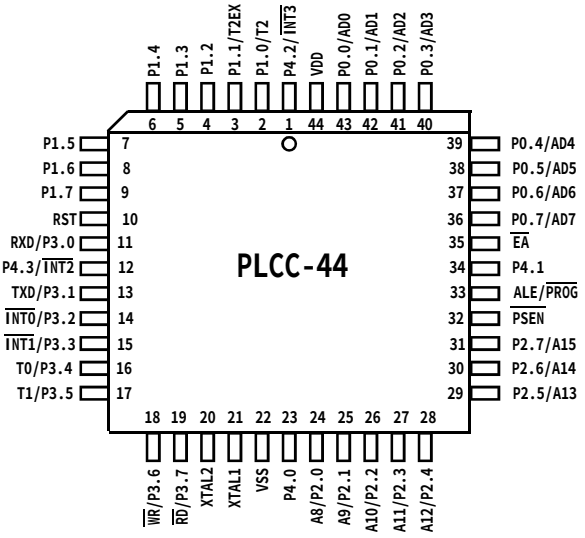
5V: 5.5 - 3.8V,乃至 3.4V； 3V: 3.6 - 2.4V,乃至 1.9V

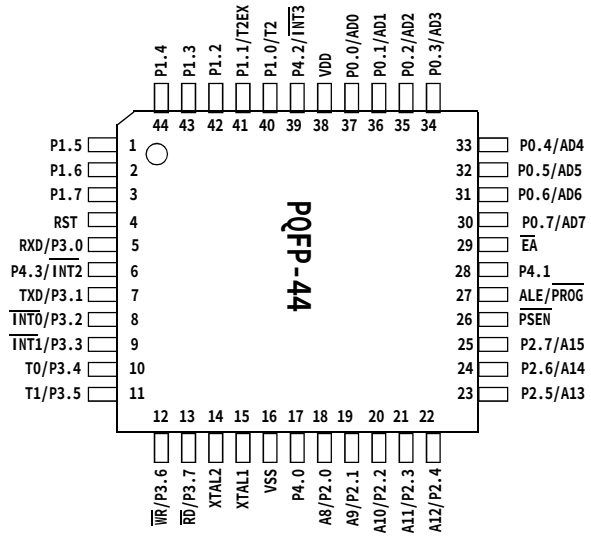
STC89 系列单片机选型一览表　　IAP / ISP 美国技术， 超低价格，15 分钟学会

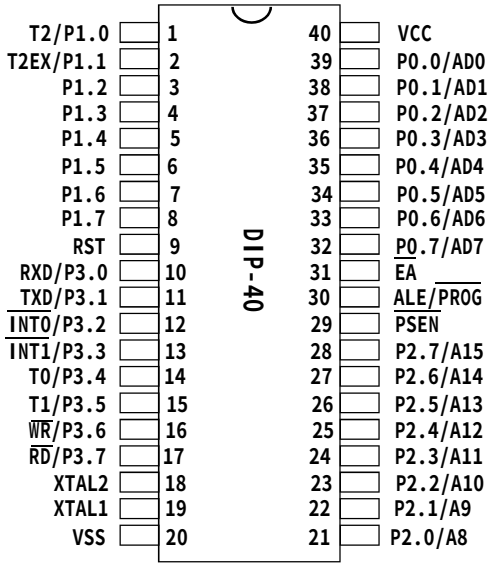
型 号	最高时钟频率Hz 5V	最高时钟频率Hz 3V	Flash 程序存储器	RAM 数据存储器	降低 EMI	看门狗	双倍速	P4 口	ISP	IAP	EEPROM	数据指针	串口 UART	中断源	优先级	定时器	A/D	向下兼容 Winbond	向下兼容 Philips	向下兼容 Atmel
STC89C51 RC	0-80M		4K	512	√	√	√	√	√	√	1K+	2	1ch+	8	4	3		W78E51	P89C51	
STC89C52 RC	0-80M		8K	512	√	√	√	√	√	√	1K+	2	1ch+	8	4	3		W78E52	P89C52	
STC89C53 RC	0-80M		15K	512	√	√	√	√	√	√		2	1ch+	8	4	3		W78E54	P89C54	AT89C55
STC89C54 RD+	0-80M		16K	1280	√	√	√	√	√	√	8K+	2	1ch+	8	4	3		W78E54	P89C54	AT89C55
STC89C58 RD+	0-80M		32K	1280	√	√	√	√	√	√	8K+	2	1ch+	8	4	3		W78E58	P89C58	AT89C51RC
STC89C516 RD+	0-80M		63K	1280	√	√	√	√	√	√		2	1ch+	8	4	3		W78E516	P89C51RD2	AT89C51RD2
STC89LE51 RC		0-80M	4K	512	√	√	√	√	√	√	1K+	2	1ch+	8	4	3		W78LE51		AT89LV51
STC89LE52 RC		0-80M	8K	512	√	√	√	√	√	√	1K+	2	1ch+	8	4	3		W78LE52		AT89LV52
STC89LE53 RC		0-80M	14K	512	√	√	√	√	√	√		2	1ch+	8	4	3		W78LE54		AT89LV55
STC89LE54 RD+		0-80M	16K	1280	√	√	√	√	√	√	8K+	2	1ch+	8	4	3		W78LE54		AT89LV55
STC89LE58 RD+		0-80M	32K	1280	√	√	√	√	√	√	8K+	2	1ch+	8	4	3		W78LE58		AT89LV51RC
STC89LE516RD+		0-80M	63K	1280	√	√	√	√	√	√		2	1ch+	8	4	3		W78LE516	P89LV51RD2	AT89LV51RD2
STC89LE516AD		0-90M	64K	512	√			√	√			2	1ch+	6	4	3	√	需要A/D转换时才选用，8路8位精度		
STC89LE516X2		0-90M	64K	512	√		√	√	√			2	1ch+	6	4	3	√	在P1.0 - P1.7口，17 个机器周期一次		

RC/RD+ 系列为真正的看门狗，缺省为关闭（冷启动），启动后无法关闭，可放心省去外部看门狗。内部 Flash 擦写次数为100,000次以上，STC89C51RC/RD+ 系列单片机出厂时就已完全加密，无法解密。用户程序是用 ISP/IAP 机制写入，一边校验一边写，无读出命令，彻底无法解密。DIP-40，PLCC-44，PQFP-44 三种封装（PLCC、QFP 有 P4 口），RC/RD+ 系列单片机 P4 口地址为 E8H,并有 2 个附加外部中断，P4.2/INT3,P4.3/INT2。STC89LE516AD/X2 系列单片机 P4 口地址为 C0H,无附加外部中断。

STC89C51RC / RD+ 系列单片机 管脚图







关于编译器 / 汇编器：

1. 任何老的编译器 / 汇编器均可使用
   Keil C51 中：Device 选择标准的 Intel8052
   头文件包含标准的 <reg52.h>
2. 新增特殊功能寄存器用
   “sfr”及“sbit”声明地址即可
3. 汇编中用“data”,或“EQU”声明地址

关于仿真及仿真器：

1. 任何老的仿真器均可使用
2. 老的仿真器仿真他可仿真的基本功能
3. 新增特殊功能用 ISP 下载看结果即可
4. STC8051 专用仿真器也已推出，人民币 1950

关于工作电压 / 时钟频率:RC/RD+ 系列是真正的 6T 单片机，兼容普通的 12 时钟 / 机器周期

内核实际6T	现有老版5V单片机,无版本号					现有新版5V单片机（B版）					
工作电压	外部时钟	单倍速相当于普通8052	实际内核运行时钟	双倍速相当于普通8052	实际内核运行时钟	外部时钟	单倍速相当于普通8052	实际内核运行时钟	双倍速相当于普通8052	实际内核运行时钟	IAP/ISP可以
5.5V - 4.5V	0-24MHz	0-24MHz	0-12MHz	0-48MHz	0-24MHz	0-40MHz	0-40MHz	0-20MHz	0-80MHz	0-40MHz	读，编程，擦除
5.5V - 3.8V	0-20MHz	0-20MHz	0-10MHz	0-40MHz	0-20MHz	0-33MHz	0-33MHz	0-16.5M	0-66MHz	0-33MHz	读，编程，擦除
5.5V - 3.6V	0-18MHz	0-18MHz	0-9MHz	0-36MHz	0-18MHz	0-24MHz	0-24MHz	0-12MHz	0-48MHz	0-24MHz	读，编程，擦除
5.5V - 3.4V	0-12MHz	0-12MHz	0-6MHz	0-24MHz	0-12MHz	0-20MHz	0-20MHz	0-10MHz	0-40MHz	0-20MHz	读（不要编程/擦除）

3V：3.6 - 2.4V(可外部 24MHz,双倍速 48MHz),2.3-1.9V 时不要进行 IAP 擦除 / 编程

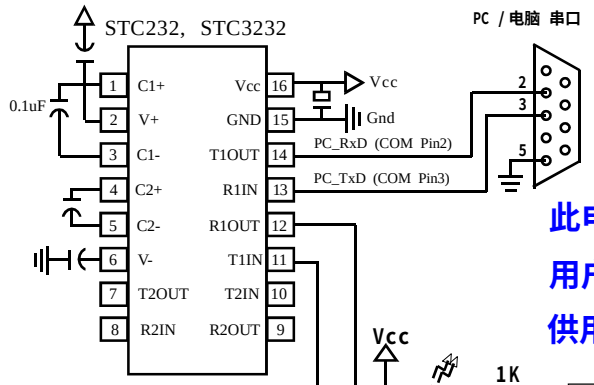
关于看门狗： RC/RD+ 系列为真正的看门狗，缺省为关闭（冷启动），启动后无法关闭。

A. 看门狗溢出复位无法关看门狗(C 版)；
B. 单片机软复位无法关看门狗(C版)
C. 带电工作时，外部复位无法关看门狗(C 版)
D. 软件无法关看门狗
E. 外部干扰无法关看门狗
F. 只有给单片机彻底断电，才可以

STC 单片机 ISP 编程 典型应用电路(89C51RC/RD+ 系列)

--- 出厂就加密(超级加密)，利用 ISP 技术写入程序，无读出命令，无法解密

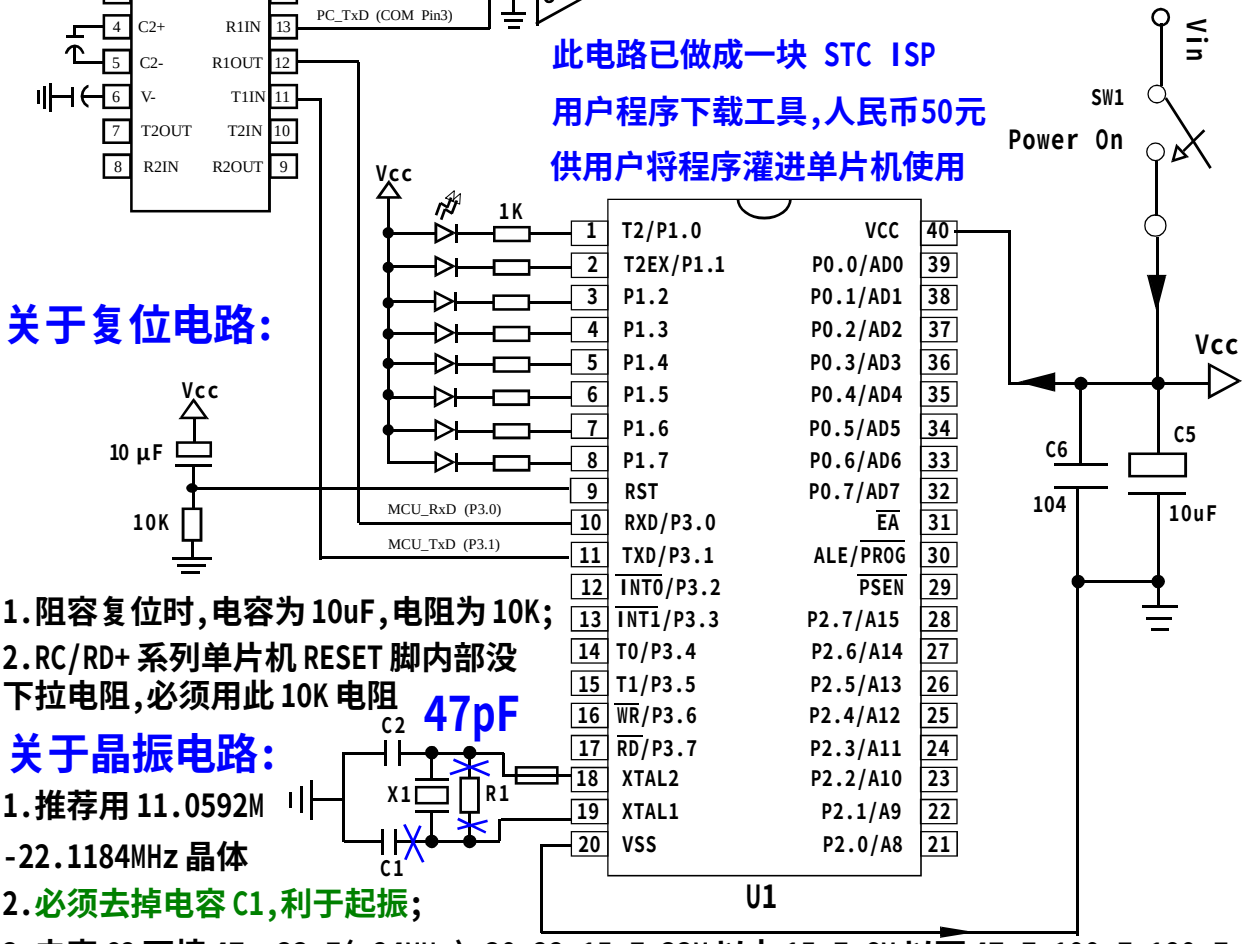
STC 单片机在线编程线路， *STC* RS-232 转换器



关于 /EA(/EA 管脚已内部上拉到 Vcc)：

1. 如外部不加上拉，或外部上拉到 Vcc，上电复位后单片机从内部开始执行程序；
2. 如外部下拉到地，上电复位后单片机从外部开始执行程序

此电路已做成一块 STC ISP 用户程序下载工具，人民币50元供用户将程序灌进单片机使用



关于复位电路：

1. 阻容复位时，电容为 10uF，电阻为 10K；
2. RC/RD+ 系列单片机 RESET 脚内部没下拉电阻，必须用此 10K 电阻

关于晶振电路：

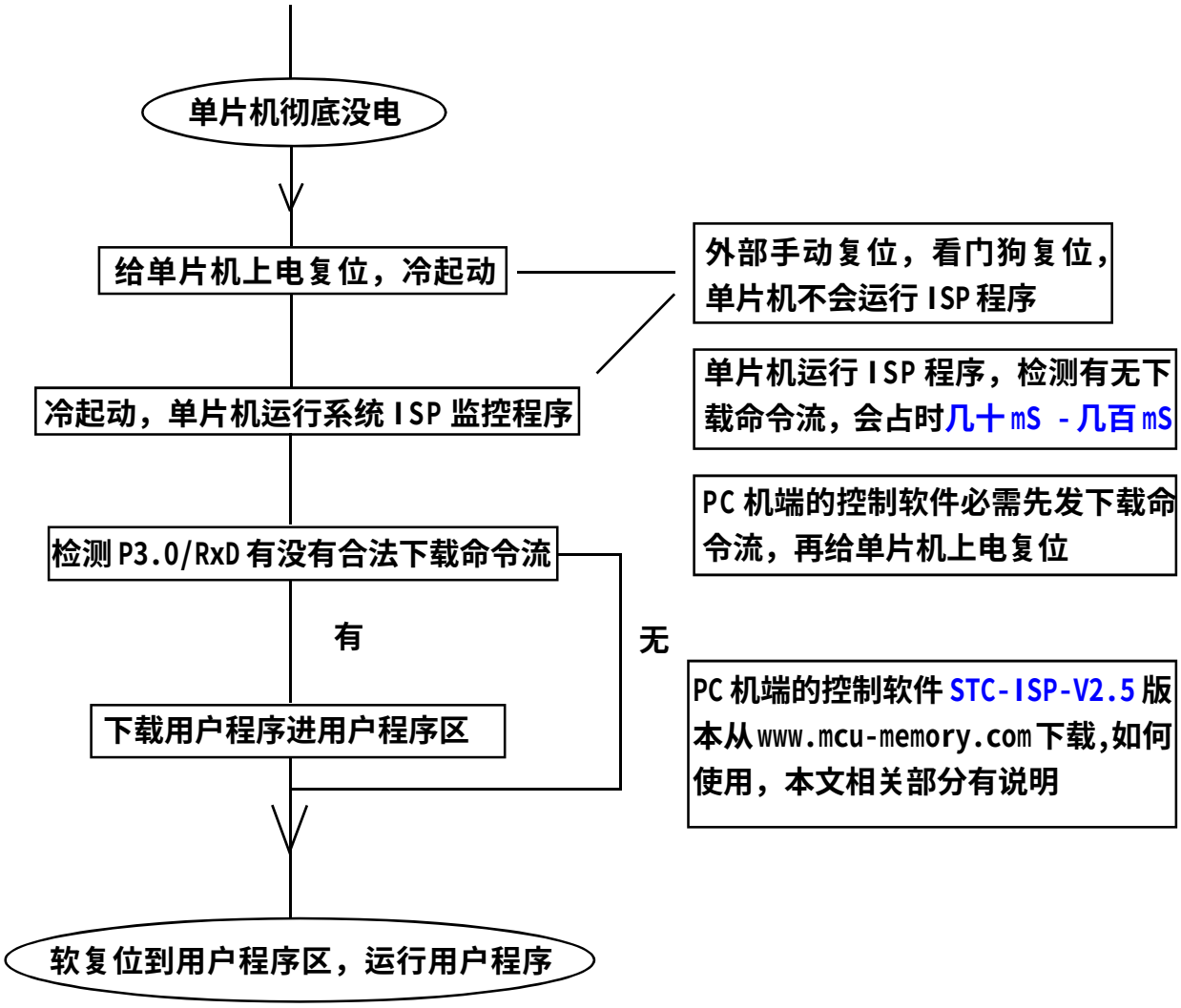
1. 推荐用 11.0592M -22.1184MHz 晶体
2. 必须去掉电容 C1，利于起振；
3. 电容 C2 可接 47～33pF(<24MHz)，30，22，15pF，33M 以上 15pF；6M 以下 47pF，100pF-180pF；

CRYSTAL(晶振)	C1(坚决不用)	C2(47pF - 15pF)	R1(不用)	C3	R2
6MHz	Don't use	47pF, 100pF	Don't use	10uF	10K/8.2K
11.0592MHz	Don't use	30pF,33pF,47pF	Don't use	10uF	10K/8.2K
22.1184MHz	Don't use	27pF,33pF,47pF	Don't use	10uF-22uF	15K,10K,8.2K
24MHz	Don't use	27pF,33pF,47pF	Don't use	10uF-22uF	15K,10K,8.2K
33MHz(Don't use)	Don't use	15pF	15k - 6.8k	10uF-22uF	15K,10K,8.2K
40MHz(Don't use)	Don't use	15pF	15k - 6.8k	10uF-22uF	15K,10K,8.2K

4. 如需工作在 24M 以上，尽量选择双倍速，外部晶振用低频率(<24MHz)，这样 EMI 会小很多；
5. 如果外部晶振频率在 33M 以上，焊 R1 电阻(15K 左右)，33MHz 以上，建议用有源时钟输入。
6. 推荐在 XTAL2 管脚串一个 120 - 160 欧姆的电阻再到晶振的管脚，如果这样 XTAL1 管脚旁的 C1 可以上（并且是推荐上）
7. 晶体振荡器(内部时钟反向放大器增益)软件设置(在用 STC-ISP 工具软件烧录时设置)：

   5V 单片机 24M 以下时，3.3V 单片机 16M 以下时，选择 1/2gain

STC89C51RC/RD+系列单片机 ISP编程 原理 注意事项



为什么有些用户下载程序不成功(在宏晶提供的下载板上)

1.可能电脑端的STC-ISP控制软件要升级，现在必须升级到正式版本STC-ISP-V2.5

2.现在单片机端(下位机)的ISP软件是V2.5C,解决了少数电脑慢，通信连不上的问题.

3.运行用户程序时，可到40M/80MHz,但ISP下载程序现版本软件只能到33M/66MHz

4.少数客户的PLCC-44,PQFP-44转DIP-40的转换座走线过长，造成时钟振荡不稳定，下载不成功，可将XTAL1脚的电容去掉，XTAL2脚的电容加大到47pF以上。

5.可能单片机内部没有ISP引导码，或者要升级PC端的STC-ISP控制软件

6.电脑端的ISP控制软件STC-ISP-V2.9测试版加了一些功能，欢迎测试。

7.电脑端的ISP控制软件STC-ISP-V2.6测试版不要用，有误

8.新的单片机端(下位机)的ISP软件是V2.7C,加了一些抗干扰措施，主要解决冷起动运行ISP程序时间过长的问题，以免客户感觉“复位慢”，实为ISP程序在检测要不要下载程序

为什么有些用户下载程序不成功(在用户自己的系统上)

1.可能用户板上有外部看门狗，需不让其起作用

2.可能用户板上P3.0/RxD，P3.1/Txd除了接RS-232转换器外，还接了RS-485等电路，需要将其断开。

超低功耗 ---- STC89C51RC / RD+ 系列单片机

1.掉电模式：

典型功耗 0.5uA， 可由外部中断唤醒，中断返回后，继续执行原程序

2.空闲模式：

典型功耗 2mA

3.正常工作模式：

典型功耗 4mA - 7mA

超强抗干扰 ---- STC89C51RC / RD+ 系列单片机

1. I/O口

输入/输出口经过特殊处理，很多干扰是从I/O进去的,每个I/O均有对VCC/对GND二级管箝位保护。

2. 电源

单片机内部的电源供电系统经过特殊处理，很多干扰是从电源进去的

3. 时钟

单片机内部的时钟电路经过特殊处理，很多干扰是从时钟部分进去的

4. 看门狗

单片机内部的看门狗电路经过特殊处理，打开后无法关闭，可放心省去外部看门狗

5. 复位电路

单片机内部的复位电路经过特殊处理，很多干扰是从复位电路部分进去的，STC89C51RC/RD+系列单片机为高电平复位。推荐外置复位电路为MAX810/STC810,STC6344,STC6345,813L,706P；也可用R/C复位，10uF电容/10k电阻,22uF/8.2k等。

6.宽电压，不怕电源抖动 5V: 6v - 3.4v 3V: 4v - 1.9v

降低单片机对外部的电磁辐射（EMI)--- 三大措施

1. 禁止ALE时钟信号输出:

RC/RD+系列8051 单片机 扩展RAM管理及禁止ALE输出 特殊功能寄存器 只写

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset value
AUXR	8Eh	Auxiliary Register 0	-	-	-	-	-	-	EXTRAM	ALEOFF	xxxx,xx00

禁止ALE信号输出(应用示例供参考,汇编语言):

MOV AUXR, #00000001B; ALEOFF位置“1”,禁止ALE时钟输出

2. 外部时钟频率降一半，6T模式: 传统的8051为每个机器周期12时钟，如将STC的增强型8051单片机在ISP烧录程序时设为双倍速（即6T模式，每个机器周期6时钟），则可将单片机外部时钟频率降低一半，有效的降低单片机时钟对外界的辐射

3. 单片机内部时钟振荡器增益降低一半: 在ISP烧录程序时将OSCDN设为1/2 gain 可以有效的降低单片机时钟高频部分对外界的辐射,5V单片机外部晶振频率<24MHz时,3V单片机外部晶振频率<16MHz时，将OSCDN设为1/2 gain。

特殊功能寄存器映像 SFR Mapping

STC89C51RC, STC89C52RC, STC89C53RC, STC89C54RD+,STC89C58RD+,STC89C516RD+
STC89LE51RC,STC89LE52RC,STC89LE53RC,STC89LE54RD+,STC89LE58RD+,STC89LE516RD+

	Bit Addressable	Non Bit Addressable							
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8h									FFh
F0h	B 0000,0000								F7h
E8h	P4 xxxx,1111								EFh
E0h	ACC 0000,0000	WDT_CONTR xx00,0000	ISP_DATA 1111,1111	ISP_ADDRH 0000,0000	ISP_ADDRL 0000,0000	ISP_CMD 1111,1000	ISP_TRIG xxxx,xxxx	ISP_CONTR 000x,x000	E7h
D8h									DFh
D0h	PSW 0000,0000								D7h
C8h	T2CON 0000,0000	T2MOD xxxx,xx00	RCAP2L 0000,0000	RCAP2H 0000,0000	TL2 0000,0000	TH2 0000,0000			CFh
C0h	XICON 0000,0000								C7h
B8h	IP xx00,0000	SADEN 0000,0000							BFh
B0h	P3 1111,1111							IPH 0000,0000	B7h
A8h	IE 0000,0000	SADDR 0000,0000							AFh
A0h	P2 1111,1111		AUXR1 xxxx,0xx0						A7h
98h	SCON 0000,0000	SBUF xxxx,xxxx							9Fh
90h	P1 1111,1111								97h
88h	TCON 0000,0000	TMOD 0000,0000	TL0 0000,0000	TL1 0000,0000	TH0 0000,0000	TH1 0000,0000	AUXR xxxx,xx00		8Fh
80h	P0 1111,1111	SP 0000,0111	DPL 0000,0000	DPH 0000,0000				PCON 0xx1,0000	87h
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

RC/RD+ 系列 8051 单片机内核特殊功能寄存器 C51 Core SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
ACC	E0h	Accumulator									0000,0000
B	F0h	B Register									0000,0000
PSW	D0h	Program Status Word	CY	AC	F0	RS1	RS0	OV	-	P	0000,0000
SP	81h	Stack Pointer									0000,0111
DPL	82h	Data Pointer Low Byte									0000,0000
DPH	83h	Data Pointer High Byte									0000,0000

RC/RD+ 系列 8051 单片机系统管理特殊功能寄存器 System Management SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset value
PCON	87h	Power Control	SMOD	-	-	POF	GF1	GF0	PD	IDL	0xx1,0000
AUXR	8Eh	Auxiliary Register 0	-	-	-	-	-	-	EXTRAM	ALEOFF	xxxx,xx00
AUXR1	A2h	Auxiliary Register 1	-	-	-	-	GF2	-	-	DPS	xxxx,0xx0

不同：STC89LE516AD / 89LE516X2 系列单片机没有 EXTRAM 控制位.

RC/RD+ 系列 8051 单片机 中断 特殊功能寄存器 Interrupt SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
IE	A8h	Interrupt Enable	EA	-	ET2	ES	ET1	EX1	ET0	EX0	0000,0000
IP	B8h	Interrupt Priority Low	-	-	PT2	PS	PT1	PX1	PT0	PX0	xx00,0000
IPH	B7h	Interrupt Priority High	PX3H	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	0000,0000
XICON	C0h	Auxiliary Interrupt Control	PX3	EX3	IE3	IT3	PX2	EX2	IE2	IT2	0000,0000

不同：STC89LE516AD 系列单片机没有(XICON,PX3H,PX2H),因为 P4.2/P4.3 无中断.

RC/RD+ 系列 8051 单片机 I/O 口 特殊功能寄存器 Port SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
P0	80h	8-bit Port 0	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	1111,1111
P1	90h	8-bit Port 1	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	1111,1111
P2	A0h	8-bit Port 2	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	1111,1111
P3	B0h	8-bit Port 3	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	1111,1111
P4	E8h	4-bit Port 4	-	-	-	-	P4.3	P4.2	P4.1	P4.0	xxxx,1111

不同：STC89LE516AD / 89LE516X2 系列单片机 P4 口地址为 C0h,而不是 E8h.

RC/RD+系列 8051 单片机 串行口 特殊功能寄存器 Serial I/O Port SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
SCON	98h	Serial Control	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	0000,0000
SBUF	99h	Serial Data Buffer									xxxx,xxxx
SADEN	B9h	Slave Address Mask									0000,0000
SADDR	A9h	Slave Address									0000,0000

RC/RD+系列 8051 单片机 定时器 特殊功能寄存器 Timer SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
TCON	88h	Timer / Counter 0 and 1 Control	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000,0000
TMOD	89h	Timer / Counter 0 and 1 Modes	GATE GATE1	C/T# C/T1#	M1 M1_1	M0 M1_0	GATE GATE0	C/T# C/T0#	M1 M0_1	M0 M0_0	0000,0000
TL0	8Ah	Timer / Counter 0 Low Byte									0000,0000
TH0	8Ch	Timer / Counter 0 High Byte									0000,0000
TL1	8Bh	Timer / Counter 1 Low Byte									0000,0000
TH1	8Dh	Timer / Counter 1 High Byte									0000,0000
T2CON	C8h	Timer / Counter 2 Control	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2#	CP/RL2#	0000,0000
T2MOD	C9h	Timer / Counter 2 Mode	-	-	-	-	-	-	T2OE	DCEN	xxxx,xx00
RCAP2L	CAh	Timer / Counter 2 Reload/Capture Low Byte									0000,0000
RCAP2H	CBh	Timer / Counter 2 Reload/Capture High Byte									0000,0000
TL2	CCh	Timer / Counter 2 Low Byte									0000,0000
TH2	CDh	Timer / Counter 2 High Byte									0000,0000

RC/RD+系列 8051单片机 看门狗定时器 特殊功能寄存器 Watch Dog Timer SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
WDT_CONTR	E1h	Watch-Dog-Timer Control register	-	-	EN_WDT	CLR_WDT	IDLE_WDT	PS2	PS1	PS0	xx00,0000

RC/RD+系列 8051 单片机 ISP/IAP 特殊功能寄存器 ISP/IAP SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
ISP_DATA	E2h	ISP/IAP Flash Data Register									1111,1111
ISP_ADDRH	E3h	ISP/IAP Flash Address High									0000,0000
ISP_ADDRL	E4h	ISP/IAP Flash Address Low									0000,0000
ISP_CMD	E5h	ISP/IAP Flash Command Register	-	-	-	-	-	MS2	MS1	MS0	xxxx,x000
ISP_TRIG	E6h	ISP/IAP Flash Command Trigger									xxxx,xxxx
ISP_CONTR	E7h	ISP/IAP Control Register	ISPEN	SWBS	SWRST	-	-	WT2	WT1	WT0	000x,x000

STC89C51RC/RD+ 系列单片机扩展 RAM 的使用
STC89C51RC/RD+ 系列单片机扩展 RAM 的禁止

适用型号:

STC89C51RC,STC89C52RC,STC89C53RC,STC89LE51RC,STC89LE52RC,STC89LE53RC

STC89C54RD+,STC89C58RD+,STC89C516RD+,STC89LE54RD+,STC89LE58RD+,STC89LE516RD+

普通 89C51,89C52 系列单片机的内部 RAM 只有 128(89C51)/256(89C52)供用户使用

(1).低 128 字节的内部 RAM（地址:00H-7FH),可直接寻址或间接寻址,(data/idata)

(2).高 128 字节的内部 RAM（地址:80H-FFH),只能间接寻址(普通 89C51 没有),(idata)

(3).特殊功能寄存器 SFR（地址:80H-FFH)，只能直接寻址,(data)

特殊功能寄存器 SFR 和高 128 字节的内部 RAM 是通过寻址方式来区分的,传统的 8051 系列单片机只有 128-256 字节 RAM 供用户使用，在此情况下 STC 公司响应广大用户的呼声，在一些单片机内部增加了 RAM。 STC89C58RD+ 系列单片机扩展了 1024 个字节 RAM，STC89C52RC 系列单片机扩展了 256 个字节 RAM。

RC/RD+ 系列 8051 单片机 扩展 RAM 管理及禁止 ALE 输出 特殊功能寄存器 只写

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset value
AUXR	8Eh	Auxiliary Register 0	-	-	-	-	-	-	EXTRAM	ALEOFF	xxxx,xx00

Symbol 符号 Function 功能

EXTRAM Internal/External RAM access 内部 / 外部 RAM 存取

0: 内部扩展的 EXT_RAM 可以存取.

RD+ 系列单片机

在 00H 到 3FFH 单元(1024 字节),使用 MOVX @DPTR 指令访问,超过 400H 的地址空间总是访问外部数据存储器（含 400H 单元)，MOVX @Ri 只能访问 00H 到 FFH 单元

RC 系列单片机

在 00H 到 FFH 单元(256 字节),使用 MOVX @DPTR 指令访问,超过 100H 的地址空间总是访问外部数据存储器（含 100H 单元)，MOVX @Ri 只能访问 00H 到 FFH 单元

1: External data memory access.

外部数据存储器存取,禁止访问内部扩展 RAM，此时 MOVX @DPTR / MOVX @Ri 的使用同普通 8052 单片机

ALEOFF Disable/enable ALE.

0: ALE is emitted at a constant rate of 1/3 the oscillator frequency in 6 clock mode, 1/6 fosc in 12 clock mode

ALE 脚输出固定的 1/6 晶振频率信号在 12 时钟模式时,在 6 时钟模式时输出固定的 1/3 晶振频率信号.

1: ALE is active only during a MOVX or MOVC instruction.

ALE 脚仅在执行 MOVX or MOVC 指令时才输出信号,好处是：降低了系统对外界的 EMI.

注解：STC89LE516AD,STC89LE516X2 系列无 EXTRAM 控制位,仅有 ALEOFF 控制位。

STC89LE516AD/X2 系列用 MOVX A，@Ri / MOVX @Ri，A 指令固定访问内部扩展的 EXTRAM，用 MOVX A，@DPTR / MOVX @DPTR，A 指令固定访问外部 RAM.

应用示例供参考（汇编）：

访问内部扩展的 EXTRAM

```
;新增特殊功能寄存器声明(汇编方式)
AUXR    DATA    8EH;        或者用    AUXR  EQU   8EH    定义
MOV     AUXR,   #00000000B;   EXTRAM 位清为 ”0 ”, 其实上电复位时此位就为 ”0 ”.
    ;MOVX  A,  @DPTR / MOVX  @DPTR, A 指令可访问内部扩展的 EXTRAM
        ;RD+ 系列为(00H - 3FFH,共 1024 字节)
        ;RC 系列为(00H - FFH,共 256 字节)
    ;MOVX  A,  @Ri / MOVX  A, @Ri 指令可直接访问内部扩展的 EXTRAM
        ;使用此指令 RD+ 系列 只能访问内部扩展的 EXTRAM(00H - FFH,共 256 字节)
;写芯片内部扩展的 EXTRAM
    MOV     DPTR,   #address
    MOV     A,  #value
    MOVX    @DPTR,  A
;读芯片内部扩展的 EXTRAM
    MOV     DPTR,   #address
    MOVX    A,  @DPTR
RD+ 系列
; 如果 #address < 400H，则在 EXTRAM 位为 ”0 ”时，访问物理上在内部，逻辑上在外部的
        此 EXTRAM
; 如果 #address >= 400H，则总是访问物理上外部扩展的 RAM 或 I/O 空间（400H--FFFFH）
RC 系列
; 如果 #address < 100H，则在 EXTRAM 位为 ”0 ”时，访问物理上在内部，逻辑上在外部的
        此 EXTRAM
; 如果 #address >= 100H，则总是访问物理上外部扩展的 RAM 或 I/O 空间（100H--FFFFH）
```

禁止访问内部扩展的 EXTRAM ,以防冲突

```
MOV     AUXR,   #00000010B;   EXTRAM 控制位设置为 ”1 ”, 禁止访问 EXTRAM,以防冲突
```

有些用户系统因为外部扩展了 I/O 或者用片选去选多个 RAM 区,有时与此内部扩展的 EXTRAM 逻辑地址上有冲突，将此位设置为 ”1 ”, 禁止访问此内部扩展的 EXTRAM 就可以了.

大实话 ： 其实不用设置 AUXR 寄存器即可直接用 MOVX @DPTR 指令访问此内部扩展的 EXTRAM,超过此 RAM空间,将访问片外单元.如果系统外扩了SRAM,而实际使用的空间小于1024/256字节,则可直接将此 SRAM 省去,比如省去 STC62WV256,IS62C256,UT6264 等. 另外尽量用 MOVX A, @Ri / MOVX @Ri, A 指令访问此内部扩展的 EXTRAM,这样只能访问 256 字节的扩展 EXTRAM,但可与很多单片机兼容。如 STC89LE516AD/X2 系列 MOVX A, @Ri / MOVX @Ri, A 指令只能固定访问内部扩展的 EXTRAM, MOVX A, @DPTR / MOVX @DPTR, A 指令固定访问外部 RAM.

应用示例供参考（C 语言）：

```
/*  访问内部扩展的 EXTRAM */
/*  RD+ 系列为(00H - 3FFH, 共 1024 字节扩展的 EXTRAM) */
/*  RC 系列为(00H - FFH, 共 256 字节扩展的 EXTRAM) */
    /*  新增特殊功能寄存器声明(C 语言方式) */
    sfr AUXR    =   0x8e    /*  如果不需设置 AUXR 就不用声明 AUXR    */
    AUXR    =   0x00;   /*  0000,0000  EXTRAM 位清 0, 其实上电复位时此位就为 0 */
    unsigned char  xdata  sum,  loop_counter, test_array[128];
    /*  将变量声明成 xdata 即可直接访问此内部扩展的 EXTRAM  */
```

```
/*  写芯片内部扩展的 EXTRAM */
        sum =   0;
        loop_counter    =   128;
        test_array[0]   =   5;
/*  读芯片内部扩展的 EXTRAM */
        sum =   test_array[0];
    /*  RD+ 系列:
                如果 #address < 400H，则在 EXTRAM 位为 ”0 ” 时，访问物理上在内部，逻辑
                    上在外部的此 EXTRAM
                如果 #address>=400H，则总是访问物理上外部扩展的 RAM 或 I/O 空间（400H-FFFFH）
        RC 系列:
                如果 #address < 100H，则在 EXTRAM 位为 ”0 ” 时，访问物理上在内部，逻辑
                    上在外部的此 EXTRAM
                如果 #address>=100H，总是访问物理上外部扩展的 RAM 或 I/O 空间（100H--FFFFH）
    */
```

禁止访问内部扩展的 EXTRAM，以防冲突

```
    AUXR    =   0x02;   /*  0000,0010，EXTRAM 位设为 ”1 ”，禁止访问 EXTRAM,以防冲突 */
```

有些用户系统因为外部扩展了 I/O 或者用片选去选多个 RAM 区,有时与此内部扩展的 EXTRAM 逻辑上有冲突，将此位设置为 ”1 ”， 禁止访问此内部扩展的 EXTRAM 就可以了.

AUXR 是只写寄存器

所谓只写，就是直接用“ MOV AUXR， #data ”去写,而不要用含读的操作如“ 或，与，入栈 ”因为他不让你读，如去读，读出的数值不确定，用含读的操作如“ 或，与，入栈 ”，会达不到需要的效果。

中断

RC/RD+系列 8051 单片机 中断 特殊功能寄存器 Interrupt SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
IE	A8h	Interrupt Enable	EA	-	ET2	ES	ET1	EX1	ET0	EX0	0000,0000
IP	B8h	Interrupt Priority Low	-	-	PT2	PS	PT1	PX1	PT0	PX0	xx00,0000
IPH	B7h	Interrupt Priority High	PX3H	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	0000,0000
XICON	C0h	Auxiliary Interrupt Control	PX3	EX3	IE3	IT3	PX2	EX2	IE2	IT2	0000,0000

中断与普通 8052 完全兼容，优先级可设为 4 级，另增加 2 个外部中断 INT2/P4.3，INT3/P4.2。

Interrupt Source 中断源	Vector Address 中断向量地址	Polling Sequence 中断查询次序	中断优先级设置	优先级0 最低	优先级1	优先级2	优先级3 最高	Interrupt Request 中断请求
/INT0	0003H	0(最优先)	PX0H,PX0	0,0	0,1	1,0	1,1	IE0
Timer 0	000BH	1	PT0H,PT0	0,0	0,1	1,0	1,1	TF0
/INT1	0013H	2	PX1H,PX1	0,0	0,1	1,0	1,1	IE1
Timer 1	001BH	3	PT1H,PT1	0,0	0,1	1,0	1,1	IF1
UART	0023H	4	PSH, PS	0,0	0,1	1,0	1,1	RI + TI
Timer 2	002BH	5	PT2H,PT2	0,0	0,1	1,0	1,1	TF2 + EXF2
/INT2	0033H	6	PX2H,PX2	0,0	0,1	1,0	1,1	IE2
/INT3	003BH	7(最低)	PX3H,PX3	0,0	0,1	1,0	1,1	IE3

Name	Function
PX3	External interrupt 3 priority high if set
EX3	External interrupt 3 enable if set
IE3	IE3 is set/cleared automatically by hardware when interrupt is detected/serviced
IT3	External interrupt 3 is falling-edge/low-level triggered when this bit is set/cleared by software
PX2	External interrupt 2 priority high if set
EX2	External interrupt 2 enable if set
IE2	IE2 is set/cleared automatically by hardware when interrupt is detected/serviced
IT2	External interrupt 2 is falling-edge/low-level triggered when this bit is set/cleared by software
PXH3	External interrupt 3 priority highest if set
PXH2	External interrupt 2 priority highest if set

降低单片机对系统的电磁干扰（EMI)--- 三大措施

1. 禁止 ALE 信号输出，适用型号：

STC89C51RC,STC89C52RC,STC89C53RC,STC89LE51RC,STC89LE52RC,STC89LE53RC
STC89C54RD+,STC89C58RD+,STC89C516RD+,STC89LE54RD+,STC89LE58RD+,STC89LE516RD+
STC89LE516AD / X2 系列 （注：此系列单片机无 EXTRAM 控制位）

RC/RD+ 系列 8051 单片机 扩展 RAM 管理及禁止 ALE 输出 特殊功能寄存器 只写

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset value
AUXR	8Eh	Auxiliary Register 0	-	-	-	-	-	-	EXTRAM	ALEOFF	xxxx,xx00

禁止 ALE 信号输出(应用示例供参考,C 语言):

```
sfr     AUXR =    0x8e;     /* 声明 AUXR 寄存器的地址 */
AUXR =    0x01;
/* ALEOFF 位置 1,禁止 ALE 信号输出,提升系统的 EMI 性能,复位后为 0,ALE 信号正常输出 */
```

禁止 ALE 信号输出(应用示例供参考,汇编语言):

```
AUXR  EQU   8Eh   ;      或      AUXR  DATA  8Eh
MOV   AUXR,#00000001B;   ALEOFF 位置“1”,禁止 ALE 信号输出,提升了系统的 EMI 性能
```

2. 外部时钟频率降一半，6T 模式：传统的 8051 为每个机器周期 12 时钟，如将 STC 的增强型 8051 单片机在 ISP 烧录程序时设为双倍速（及 6T 模式，每个机器周期 6 时钟），则可将单片机外部时钟频率降低一半，有效的降低单片机时钟对外界的干扰

3. 单片机内部时钟振荡器增益降低一半：在 ISP 烧录程序时将 OSCDN 设为 1/2 gain 可以有效的降低单片机时钟高频部分对外界的辐射,但此时外部晶振频率尽量不要高于 24MHz。

P4 口

RC/RD+ 系列 8051 单片机 I/O 口 特殊功能寄存器 Port SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
P0	80h	8-bit Port 0	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	1111,1111
P1	90h	8-bit Port 1	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	1111,1111
P2	A0h	8-bit Port 2	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	1111,1111
P3	B0h	8-bit Port 3	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	1111,1111
P4	E8h	4-bit Port 4	-	-	-	-	P4.3	P4.2	P4.1	P4.0	xxxx,1111

汇编语言：

```
P4      DATA    0E8H ;    or   P4   EQU  0E8H
MOV     A,      P4   ;    Read P4 status to Accumulator.
MOV     P4,     #0AH ;    Output data “A”through P4.0 - P4.3
ORL     P4,     #00000001B ;  P4.0 = 1
ANL     P4,     #11111110B ;  P4.0 = 0
MOV     P4,     #0AH ;    Output data “A”through P4.0 - P4.3
```

C 语言:

```
sfr     P4   =   0xe8;    C 语言中声明 P4 口特殊功能寄存器地址
```

注：STC89LE516AD,STC89LE516X2,STC89LE58AD,STC89LE54AD 的 P4 口地址在 C0h。

双数据指针 DPTR0,DPTR1 的使用

适用型号:

STC89C51RC,STC89C52RC,STC89C53RC,STC89LE51RC,STC89LE52RC,STC89LE53RC

STC89C54RD+,STC89C58RD+,STC89C516RD+,STC89LE54RD+,STC89LE58RD+,STC89LE516RD+

STC89LE516AD,STC89LE516X2

RC/RD+/AD/X2 系列 8051 单片机 双数据指针 特殊功能寄存器

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset value
AUXR1	A2h	Auxiliary Register 1	-	-	-	-	GF2	-	-	DPS	xxxx,0xx0

Symbol 符号 Function 功能

GF2 General purpose user-defined flag. 通用功能用户自定义位

DPS DPTR registers select bit. DPTR 寄存器选择位

0: DPTR0 is selected DPTR0 被选择

1: DPTR1 is selected DPTR1 被选择

此系列单片机有两个 16-bit 数据指针,DPTR0,DPTR1.当 DPS 选择位为 0 时,选择 DPTR0, 当 DPS 选择位为 1 时,选择 DPTR1.

AUXR1 特殊功能寄存器,位于 A2H 单元,其中的位不可用布尔指令快速访问.但由于 DPS 位位于 bit0,故对 AUXR1 寄存器用 INC 指令,DPS 位便会反转,由 0 变成 1 或由 1 变成 0,即可实现双数据指针的快速切换.

应用示例供参考:

```
;新增特殊功能寄存器定义
AUXR1   DATA    0A2H
MOV     AUXR1,  #0      ;此时 DPS 为 0,DPTR0 有效

MOV     DPTR,   #1FFH   ;置 DPTR0 为 1FFH
MOV     A,      #55H
MOVX    @DPTR,  A       ;将 1FFH 单元置为 55H

MOV     DPTR,   #2FFH   ; 置 DPTR0 为 2FFH
MOV     A,      #0AAH
MOVX    @DPTR,  A       ;将 2FFH 单元置为 0AAH

INC     AUXR1           ; 此时 DPS 为 1,DPTR1 有效
MOV     DPTR,   #1FFH   ; 置 DPTR1 为 1FFH
MOVX    A,      @DPTR   ; 读 DPTR1 数据指针指向的 1FFH 单元的内容,累加器 A 变为 55H.

INC     AUXR1           ; 此时 DPS 为 0,DPTR0 有效
MOVX    A,      @DPTR   ; 读 DPTR0 数据指针指向的 2FFH 单元的内容,累加器 A 变为 0AAH.

INC     AUXR1           ; 此时 DPS 为 1,DPTR1 有效
MOVX    A,      @DPTR   ; 读 DPTR1 数据指针指向的 1FFH 单元的内容,累加器 A 变为 55H.

INC     AUXR1           ; 此时 DPS 为 0,DPTR0 有效
MOVX    A,      @DPTR   ; 读 DPTR0 数据指针指向的 2FFH 单元的内容,累加器 A 变为 0AAH.
```

结论: 与 Philips 使用方式一致

看门狗应用

适用型号:

STC89C51RC,STC89C52RC,STC89C53RC,STC89LE51RC,STC89LE52RC,STC89LE53RC

STC89C54RD+,STC89C58RD+,STC89C516RD+,STC89LE54RD+,STC89LE58RD+,STC89LE516RD+

宏晶技术支持,请直接向宏晶采购晶片,而不要通过中间商,支持我们更好地服务

RC/RD+系列 8051单片机 看门狗定时器 特殊功能寄存器 Watch Dog Timer SFR

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
WDT_CONTR	E1h	Watch-Dog-Timer Control register	-	-	EN_WDT	CLR_WDT	IDLE_WDT	PS2	PS1	PS0	xx00,0000

Symbol 符号 Function 功能

EN_WDT Enable WDT bit. When set, WDT is started
看门狗允许位， 当设置为“1”时，看门狗启动。

CLR_WDT WDT clear bit. When set, WDT will recount. Hardware will automatically clear this bit.
看门狗清“0”位，当设为“1”时，看门狗将重新计数。硬件将自动清“0”此位。

IDLE_WDT When set, WDT is enabled in IDLE mode. When clear, WDT is disabled in IDLE mode
看门狗“IDLE”模式位，当设置为“1”时，看门狗定时器在“空闲模式”计数
当清“0”该位时，看门狗定时器在“空闲模式”时不计数

PS2, PS1, PS0 Pre-scale value of Watchdog timer is shown as the bellowed table:
看门狗定时器预分频值，如下表所示

PS2	PS1	PS0	Pre-scale 预分频	WDT Period @20MHz and 12 clocks mode
0	0	0	2	39.3 mS
0	0	1	4	78.6 mS
0	1	0	8	157.3 mS
0	1	1	16	314.6 mS
1	0	0	32	629.1 mS
1	0	1	64	1.25S
1	1	0	128	2.5S
1	1	1	256	5S

The WDT period is determined by the following equation 看门狗溢出时间计算

看门狗溢出时间 = (N x Pre-scale x 32768) / Oscillator frequency

N = 12，当在12 clock mode时，N = 6，当在6 clock mode时

设时钟为12MHz，12时钟模式

看门狗溢出时间 = (12 x Pre-scale x 32768) / 12000000 = Pre-scale x 393216 / 12000000

PS2	PS1	PS0	Pre-scale 预分频	WDT Period @12MHz and 12 clocks mode
0	0	0	2	65.5 mS
0	0	1	4	131.0 mS
0	1	0	8	262.1 mS
0	1	1	16	524.2 mS
1	0	0	32	1.0485S
1	0	1	64	2.0971S
1	1	0	128	4.1943S
1	1	1	256	8.3886S

设时钟为 11.0592MHz，12 时钟模式

看门狗溢出时间 = (12 x Pre-scale x 32768) / 11059200 = Pre-scale x 393216 / 11059200

PS2	PS1	PS0	Pre-scale 预分频	WDT Period @11.0592MHz and 12 clocks mode
0	0	0	2	71.1 mS
0	0	1	4	142.2 mS
0	1	0	8	284.4 mS
0	1	1	16	568.8 mS
1	0	0	32	1.1377S
1	0	1	64	2.2755S
1	1	0	128	4.5511S
1	1	1	256	9.1022S

汇编语言程序示例

```
WDT_CONTR   DATA    0E1H ;    或者   WDT_CONTR  EQU   0E1H
;复位入口
    ORG     0000H
    LJMP    Initial
    …
    ORG     0060H
Initial:
    MOV     WDT_CONTR,  #00110100B; Load initial value 看门狗定时器控制寄存器初始化
                ; EN_WDT = 1, CLR_WDT = 1, IDLE_WDT = 0, PS2 = 1, PS1 = 0, PS0 = 0
    …
Main_Loop:
    LCALL   Display_Loop
    LCALL   Keyboard_Loop
    …
    MOV     WDT_CONTR,  #00110100B ;  喂狗，  不要用 ORL    WDT_CONTR,  #00010000B
    …
    LJMP    Main_Loop
```

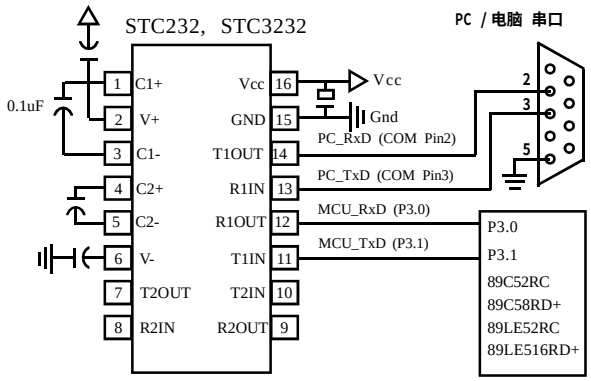
C 语言程序示例

```
#include<reg52.h>
sfr     WDT_CONTR   =   0xe1;
void main()
{   …
    while(1){
        WDT_CONTR   =   0x34;
        /* 0011,0100 EN_WDT = 1,CLR_WDT = 1, IDLE_WDT = 0, PS2 = 1, PS1 = 0, PS0 = 0 */
        display();
        keyboard();
        …
        WDT_CONTR   =   0x34;  /* 喂狗，  不要用  WDT_CONTR  =   WDT_CONTR | 0x10;*/
    }
}
```

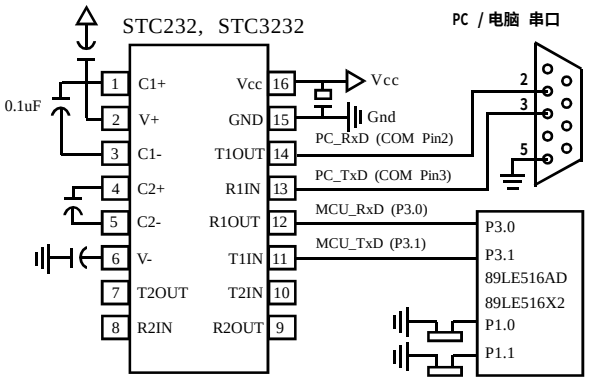
STC 89C51RC/RD+ 系列单片机在系统可编程的使用

--- 将用户代码下载进单片机内部，不用编程器

STC 单片机在线编程线路， *STC* RS-232 转换器



STC 单片机在线编程线路， *STC* RS-232 转换器



上面左图适用如下型号：

STC89C51RC， STC89C52RC， STC89C53RC

STC89LE51RC，STC89LE52RC，STC89LE53RC

STC89C54RD+，STC89C58RD+，STC89C516RD+

STC89LE54RD+,STC89LE58RD+,STC89LE516RD+

STC89C516RD，STC89C58RD （老产品，不要选）

STC89LV516RD,STC89LV58RD （老产品，不要选）

上面右图适用如下型号：

STC89LE516AD，STC89LE516X2，STC89LE58AD， STC89LE54AD， STC89LE52AD

STC89 系列单片机大部分具有在系统可编程（ISP）特性，ISP 的好处是：省去购买通用编程器，单片机在用户系统上即可下载 / 烧录用户程序，而无须将单片机从已生产好的产品上拆下，再用通用编程器将程序代码烧录进单片机内部。有些程序尚未定型的产品可以一边生产，一边完善，加快了产品进入市场的速度，减小了新产品由于软件缺陷带来的风险。由于可以将程序直接下载进单片机看运行结果故也可以不用仿真器。

大部分 STC89 系列单片机在销售给用户之前已在单片机内部固化有 ISP 系统引导程序，配合 PC 端的控制程序即可将用户的程序代码下载进单片机内部，故无须编程器(速度比通用编程器快)。不要用通用编程器编程，否则有可能将单片机内部已固化的 ISP 系统引导程序擦除，造成无法使用 STC 提供的 ISP 软件下载用户的程序代码。

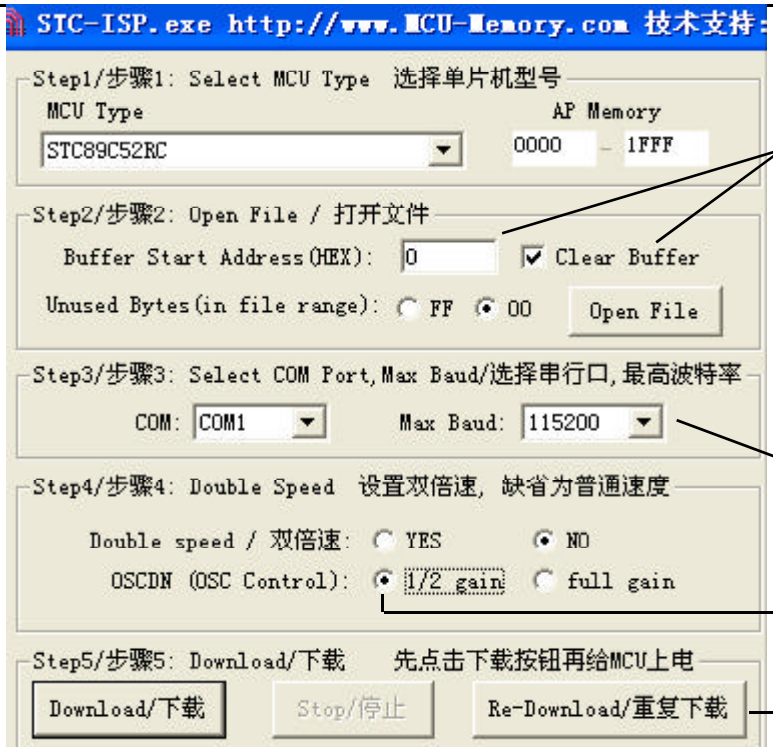
如何获得及使用 STC 提供的 ISP 下载工具（STC-ISP.exe 软件）：

(1). 获得 STC 提供的 ISP 下载工具（软件）

登陆 www.MCU-Memory.com 网站，从 STC 半导体专栏下载 PC（电脑）端的 ISP 程序，然后将其自解压，再安装即可（执行 setup.exe),注意随时更新软件。

(2). 使用 STC-ISP 下载工具（软件),请随时更新，目前已到 Ver2.5 / 2.9 版本(2005/2/2),支持 *.Hex(Intel 16 进制格式)文件,RC/RD+ 系列单片机的底层软件版本为 Ver2.7C(旧版可更换)。
请谁时注意升级 PC（电脑）端的 ISP 程序,现 Ver2.5 是正式版，Ver2.9 测试版欢迎测试。
单片机的底层软件版本为 Ver2.7C 的单片机，PC（电脑）端的 ISP 程序必须是 Ver2.5 以上

(3).已经固化有 ISP 引导码,并设置为上电复位进入 ISP 的 STC89C51RC/RD+ 系列单片机出厂时就已完全加密,需要单片机内部的电放光后上电复位(冷起动)才运行系统 ISP 程序。



第一次调文件进缓冲区，要清缓冲区。

要调几个文件进缓冲区，一次同时烧录进单片机，除每次均要指定缓冲区起始地址外，第二次及以后不能清缓冲区

用户根据需要限制最大通信波特率

5V 单片机，24MHz 以下
3V 单片机，16MHz 以下
选择1/2 gain,指外部晶振

大批量生产时使用

Step1/ 步骤 1：选择你所使用的单片机型号，如 STC89C58RD+， STC89LE516AD 等

Step2/ 步骤 2：打开文件，要烧录用户程序，必须调入用户的程序代码（*.bin， *.hex）

Step3/ 步骤 3：选择串行口，你所使用的电脑串口，如串行口 1--COM1， 串行口 2--COM2,...

有些新式笔记本电脑没有 RS-232 串行口,可买一条 USB-RS232 转接器，人民币 70 元左右。

Step4/ 步骤 4：设置是否双倍速，双倍速选中 Double Speed 即可

STC89C51RC / RD+ 系列可以反复设置 双倍速 / 单倍速

STC89LE516AD 为单倍速，STC89LE516X2 为双倍速,用户自己无法指定 双倍速 / 单倍速

STC89C516RD 系列出厂时为单倍速，用户可指定设为双倍速，如想从双倍速恢复成单倍速，则需用通用编程器擦除整个晶片方可，这会将单片机内部已烧录的 ISP 引导程序擦除。一般使用缺省设置即可，无须设置。

OSCDN: 单片机时钟振荡器增益降一半

选 1/2 gain 为降一半，降低 EMI；选 full gain（全增益）为正常状态。

Step5/ 步骤 5：选择“Download/ 下载”按钮下载用户的程序进单片机内部，可重复执行 Step5/ 步骤 5， 也可选择“Re-Download/ 重复下载”按钮

下载时注意看提示，主要看是否要给单片机上电或复位， 下载速度比一般通用编程器快。

一般先选择“Download/ 下载”按钮，然后再给单片机上电复位(先彻底断电)，而不要先上电

关于硬件连接：

(1). MCU/ 单片机 RXD(P3.0) --- RS-232 转换器 --- PC/ 电脑 TXD(COM Port Pin3)
(2). MCU/ 单片机 TXD(P3.1) --- RS-232 转换器 --- PC/ 电脑 RXD(COM Port Pin2)
(3). MCU/ 单片机 GND ---------- PC/ 电脑 GND(COM Port Pin5)
(4). P1.0, P1.1 = 0, 0 ；上电复位进入 ISP 模式,下载完后释放 P1.0,P1.1, 89LE516AD/X2 运行用户程序.
STC89C51RC/RD+,STC89LE51RC/RD+ 系列单片机不需要 P1.0,P1.1 = 0, 0
STC89LE516AD,STC89LE516X2,STC89LE58AD,STC89LE54AD 需要 P1.0,P1.1 = 0, 0
(5). RS-232 转换器可选用 SP232/MAX232(4.5-5.5V),SP3232/MAX3232(3V-5.5V).
SP232/MAX232 尽量选用 SOP 封装(窄体,SP232EEN)，SP3232 尽量选用 SSOP 封装(SP3232EEA)

如用户系统没有 RS-232 接口，
可使用 STC-ISP Ver 2.0B.PCB 演示板作为编程工具

STC-ISP Ver 2.0B PCB 板可完成下载 / 烧录用户程序的功能。

在 STC-ISP Ver 2.0B PCB 板完成下载 / 烧录：

关于硬件连接：

(1.) 根据单片机的工作电压选择单片机电源电压

A. 5V 单片机,短接 JP1 的 MCU-VCC, 5V 电源管脚

B. 3V 单片机,短接 JP1 的 MCU-VCC, 3.3V 电源管脚

(2.) 根据单片机的工作电压选择复位信号

A. 5V 单片机,短接 JP2 的 MCU-RST, 5V/MCU-RST 信号管脚

B. 3.3V 单片机,短接 JP2 的 MCU-RST, 3.3V/MCU-RST 信号管脚

(3.)连接线(宏晶提供)

A. 将一端有 9 芯连接座的插头插入 PC/ 电脑 RS-232 串行接口插座用于通信

B. 将同一端的 USB 插头插入 PC/ 电脑 USB 接口用于取电

C. 将只有一个 USB 插头的一端插入宏晶的 STC-ISP Ver 2.0B PCB 板 USB1 插座用于 RS-232 通信和供电,此时 +5V Power 灯亮(D10,USB 接口有电)

(4.)其他插座不需连接

(5.)SW1 开关处于非按下状态，此时 MCU-VCC Power 灯不亮(D9)，没有给单片机通电

(6.)SW3 开关

处于非按下状态，P1.0, P1.1 = 1, 1,不短接到地。

适用于：STC89C51RC / RD+, /RD 系列单片机

处于按下状态， P1.0, P1.1 = 0, 0,短接到地。

适用于：STC89LE516AD, STC89LE58AD, STC89LE54AD, STC89LE516X2

(7.)将单片机插进 U1-Socket 锁紧座，锁紧单片机

(8.)关于软件：选择“Download/ 下载”(必须在给单片机上电之前让 PC 先发一串合法下载命令)

(9.)按下 SW1 开关，给单片机上电复位，此时 MCU-VCC Power 灯亮(D9)

此时 STC 单片机进入 ISP 模式(STC89C51RC/RD+ 系列冷启动进入 ISP)

(10.)下载成功后，再按 SW1 开关，此时 SW1 开关处于非按下状态，MCU-VCC Power 灯不亮(D9)，给单片机断电，取下单片机。

利用 STC-ISP Ver 2.0B PCB 板进行 RS-232 转换
单片机在用户自己的板上完成下载 / 烧录：

1. U1-Socket 锁紧座不得插入单片机
2. 将用户系统上的电源(MCU-VCC,GND)及单片机的 P3.0/RXD,P3.1/TXD 接入转换板 CN2 插座这样用户系统上的单片机就具备了与 PC/ 电脑进行通信的能力
3. 将用户系统的单片机的 P1.0,P1.1 接入转换板 CN2 插座(仅 STC89LE516AD/X2 系列需要)
4. SW3 开关处于按下状态，P1.0, P1.1 = 0, 0,短接到地。仅 STC89LE516AD/X2 系列需要
5. 关于软件：选择“Download/ 下载”
6. 给单片机系统上电复位(注意是从用户系统自供电，不要从电脑 USB 取电,电脑 USB 座不插)
7. 下载程序时，如用户板有外部看门狗电路，不得启动，单片机必须有正确的复位,但不能在 ISP 下载程序时被外部看门狗复位,可将外部看门狗电路 WDI 端 / 或 WDO 端浮空
8. 如有 RS-485 晶片连到 P3.0/Rxd,P3.1/Txd,在下载时应将其断开。

STC89C51 RC / RD+ 系列 ISP / IAP 应用
STC89C51 RC / RD+ 系列 内部 EEPROM 的应用

-- 利用 IAP 技术可实现 EEPROM，内部 Flash 擦写次数为 100,000 次以上

RC/RD+ 系列 8051 单片机 ISP/IAP 特殊功能寄存器 ISP/IAP SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
ISP_DATA	E2h	ISP/IAP Flash Data Register									1111,1111
ISP_ADDRH	E3h	ISP/IAP Flash Address High									0000,0000
ISP_ADDRL	E4h	ISP/IAP Flash Address Low									0000,0000
ISP_CMD	E5h	ISP/IAP Flash Command Register	-	-	-	-	-	MS2	MS1	MS0	xxxx,x000
ISP_TRIG	E6h	ISP/IAP Flash Command Trigger									xxxx,xxxx
ISP_CONTR	E7h	ISP/IAP Control Register	ISPEN	SWBS	SWRST	-	-	WT2	WT1	WT0	000x,x000

ISP_DATA: ISP/IAP 操作时的数据寄存器。
ISP/IAP 从 Flash 读出的数据放在此处，向 Flash 写的数据也需放在此处

ISP_ADDRH: ISP/IAP 操作时的地址寄存器高八位。

ISP_ADDRL: ISP/IAP 操作时的地址寄存器低八位。

ISP_CMD: ISP/IAP 操作时的命令模式寄存器，须命令触发寄存器触发方可生效。

B7	B6	B5	B4	B3	B2	B1	B0	命令 / 操作 模式选择
保留					命令 选择			
-	-	-	-	-	0	0	0	Standby 待机模式，无ISP操作
-	-	-	-	-	0	0	1	AP-Flash / Data-Flash Read 对用户的应用程序Flash区及数据Flash区字节读
-	-	-	-	-	0	1	0	AP-Flash / Data-Flash Program 对用户的应用程序Flash区及数据Flash区字节编程
-	-	-	-	-	0	1	1	AP-Flash / Data-Flash Sector Erase 对用户的应用程序Flash区及数据Flash区扇区擦除

;5V 单片机，应在 3.6V 以下时让其复位，3V 单片机，应在 2.4V 以下时让其复位
;在此电压以下，此时再用 ISP/IAP 功能，编程 / 擦除 Flash，不能保证能达到要求
;复位电路可选 MAX810,STC810,STC6345,STC6344，813L,706P 等

程序在系统ISP程序区时可以对用户应用程序区/数据Flash区(EEPROM)进行字节读/字节编程/扇区擦除；程序在用户应用程序区时，仅可以对数据 Flash 区(EEPROM)进行字节读 / 字节编程 / 扇区擦除。已经固化有 ISP 引导码，并设置为上电复位进入 ISP 的 STC89C51RC/RD+ 系列单片机出厂时就已完全加密。

ISP_TRIG: ISP/IAP 操作时的命令触发寄存器。
在 ISPEN(ISP_CONTR.7) = 1 时，对 ISP_TRIG 先写入 46h，再写入 B9h，
ISP/IAP 命令才会生效。

ISP_CONTR: ISP/IAP 控制寄存器。

B7	B6	B5	B4	B3	B2	B1	B0	Reset Value
ISPEN	SWBS	SWRST	-	-	WT2	WT1	WT0	000x,x000

ISPEN: ISP/IAP 功能允许位。0：禁止 ISP/IAP 编程改变 Flash,1:允许编程改变 Flash
SWBS: 软件选择从用户主程序区启动（0），还是从 I S P 程序区启动（1）。
SWRST: 0： 不操作； 1： 产生软件系统复位，硬件自动清零。

设置等待时间			CPU 等待时间(机器周期)			
WT2	WT1	WT0	Read	Program	Sector Erase	Recommended System Clock
0	1	1	6	30	5471	5MHz
0	1	0	11	60	10942	10MHz
0	0	1	22	120	21885	20MHz
0	0	0	43	240	43769	40MHz

STC89C51RC/RD+ 系列内部可用 Data Flash(EEPROM)的地址(与程序空间是分开的)：
有网友来电说不能 IAP 写数据，后检查原来他把地址指向了程序区，被单片机忽略
程序在用户应用程序区(AP 区)时,仅可以对 Data Flash(EEPROM)进行 IAP/ISP 操作,不可以修改自身，这是系统可靠的基础。

STC89C51RC,STC89LE51RC 单片机内部可用 Data Flash(EEPROM)的地址：

第一扇区		第二扇区		第三扇区		第四扇区		每个扇区512字节 建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
1000h	11FFh	1200h	13FFh	1400h	15FFh	1600h	17FFh	
第五扇区		第六扇区		第七扇区		第八扇区		
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
1800h	19FFh	1A00h	1BFFh	1C00h	1DFFh	1E00h	1FFFh	

STC89C52RC,STC89LE52RC 单片机内部可用 Data Flash(EEPROM)的地址：

第一扇区		第二扇区		第三扇区		第四扇区		每个扇区512字节 建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
2000h	21FFh	2200h	23FFh	2400h	25FFh	2600h	27FFh	
第五扇区		第六扇区		第七扇区		第八扇区		
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
2800h	29FFh	2A00h	2BFFh	2C00h	2DFFh	2E00h	2FFFh	

STC89C54RD+,STC89LE54RD+ 单片机内部可用 Data Flash(EEPROM)的地址：

第一扇区		第二扇区		第三扇区		第四扇区		每个扇区512字节
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
4000h	41FFh	4200h	43FFh	4400h	45FFh	4600h	47FFh	
第五扇区		第六扇区		第七扇区		第八扇区		建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
4800h	49FFh	4A00h	4BFFh	4C00h	4DFFh	4E00h	4FFFh	
第九扇区		第十扇区		第十一扇区		第十二扇区		每个扇区512字节
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
5000h	51FFh	5200h	53FFh	5400h	55FFh	5600h	57FFh	
第十三扇区		第十四扇区		第十五扇区		第十六扇区		建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
5800h	59FFh	5A00h	5BFFh	5C00h	5DFFh	5E00h	5FFFh	
第十七扇区		第十八扇区		第十九扇区		第二十扇区		每个扇区512字节
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
6000h	61FFh	6200h	63FFh	6400h	65FFh	6600h	67FFh	
第二十一扇区		第二十二扇区		第二十三扇区		第二十四扇区		建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
6800h	69FFh	6A00h	6BFFh	6C00h	6DFFh	6E00h	6FFFH	
第二十五扇区		第二十六扇区		第二十七扇区		第二十八扇区		每个扇区512字节
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
7000h	71FFh	7200h	73FFh	7400h	75FFh	7600h	77FFH	
第二十九扇区		第三十扇区		第三十一扇区		第三十二扇区		建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
7800h	79FFh	7A00h	7BFFh	7C00h	7DFFh	7E00h	7FFFH	
第三十三扇区		第三十四扇区		第三十五扇区		第三十六扇区		每个扇区512字节
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
8000h	81FFh	8200h	83FFh	8400h	85FFh	8600h	87FFH	
第三十七扇区		第三十八扇区		第三十九扇区		第四十扇区		建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
8800h	89FFh	8A00h	8BFFh	8C00h	8DFFh	8E00h	8FFFH	
第四十一扇区		第四十二扇区		第四十三扇区		第四十四扇区		每个扇区512字节
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
9000h	91FFh	9200h	93FFh	9400h	95FFh	9600h	97FFH	
第四十五扇区		第四十六扇区		第四十七扇区		第四十八扇区		建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
9800h	99FFh	9A00h	9BFFh	9C00h	9DFFh	9E00h	9FFFH	
第四十九扇区		第五十扇区		第五十一扇区		第五十二扇区		每个扇区512字节
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
A000h	A1FFh	A200h	A3FFh	A400h	A5FFh	A600h	A7FFH	
第五十三扇区		第五十四扇区		第五十五扇区		第五十六扇区		建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
A800h	A9FFh	AA00h	ABFFh	AC00h	ADFFh	AE00h	AFFFH	
第五十七扇区		第五十八扇区		第五十九扇区		第六十扇区		每个扇区512字节
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
B000h	B1FFh	B200h	B3FFh	B400h	B5FFh	B600h	B7FFH	
第六十一扇区		第六十二扇区		第六十三扇区		第六十四扇区		建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
B800h	B9FFh	BA00h	BBFFh	BC00h	BDFFh	BE00h	BFFFH	

STC89C58RD+,STC89LE58RD+ 单片机内部可用 Data Flash(EEPROM)的地址：

第一扇区		第二扇区		第三扇区		第四扇区		每个扇区512字节 建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
8000h	81FFh	8200h	83FFh	8400h	85FFh	8600h	87FFh	
第五扇区		第六扇区		第七扇区		第八扇区		
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
8800h	89FFh	8A00h	8BFFh	8C00h	8DFFh	8E00h	8FFFh	
第九扇区		第十扇区		第十一扇区		第十二扇区		每个扇区512字节 建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
9000h	91FFh	9200h	93FFh	9400h	95FFh	9600h	97FFh	
第十三扇区		第十四扇区		第十五扇区		第十六扇区		
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
9800h	99FFh	9A00h	9BFFh	9C00h	9DFFh	9E00h	9FFFh	
第十七扇区		第十八扇区		第十九扇区		第二十扇区		每个扇区512字节 建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
A000h	A1FFh	A200h	A3FFh	A400h	A5FFh	A600h	A7FFh	
第二十一扇区		第二十二扇区		第二十三扇区		第二十四扇区		
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
A800h	A9FFh	AA00h	ABFFh	AC00h	ADFFh	AE00h	AFFFH	
第二十五扇区		第二十六扇区		第二十七扇区		第二十八扇区		每个扇区512字节 建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
B000h	B1FFh	B200h	B3FFh	B400h	B5FFh	B600h	B7FFH	
第二十九扇区		第三十扇区		第三十一扇区		第三十二扇区		
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
B800h	B9FFh	BA00h	BBFFh	BC00h	BDFFh	BE00h	BFFFH	
第三十三扇区		第三十四扇区		第三十五扇区		第三十六扇区		每个扇区512字节 建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
C000h	C1FFh	C200h	C3FFh	C400h	C5FFh	C600h	C7FFH	
第三十七扇区		第三十八扇区		第三十九扇区		第四十扇区		
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
C800h	C9FFh	CA00h	CBFFh	CC00h	CDFFh	CE00h	CFFFH	
第四十一扇区		第四十二扇区		第四十三扇区		第四十四扇区		每个扇区512字节 建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
D000h	D1FFh	D200h	D3FFh	D400h	D5FFh	D600h	D7FFH	
第四十五扇区		第四十六扇区		第四十七扇区		第四十八扇区		
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
D800h	D9FFh	DA00h	DBFFh	DC00h	DDFFh	DE00h	DFFFH	
第四十九扇区		第五十扇区		第五十一扇区		第五十二扇区		每个扇区512字节 建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
E000h	E1FFh	E200h	E3FFh	E400h	E5FFh	E600h	E7FFH	
第五十三扇区		第五十四扇区		第五十五扇区		第五十六扇区		
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
E800h	E9FFh	EA00h	EBFFh	EC00h	EDFFh	EE00h	EFFFH	
第五十七扇区		第五十八扇区		第五十九扇区		第六十扇区		每个扇区512字节 建议只用128字节以下
起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	起始地址	结束地址	
F000h	F1FFh	F200h	F3FFh	F400h	F5FFh	F600h	F7FFH	

STC89C51 RC / RD+ 系列 IAP 应用汇编简介

STC89C51 RC / RD+ 系列 内部 EEPROM 的应用

有网友要求提供汇编示例，请尽量参考已调通的 C 原程序

```
;用 DATA 还是 EQU 声明新增特殊功能寄存器地址要看你用的汇编器 / 编译器
    ISP_DATA        DATA    0E2h;   或  ISP_DATA        EQU     0E2h
    ISP_ADDRH       DATA    0E3h;   或  ISP_ADDRH       EQU     0E3h
    ISP_ADDRL       DATA    0E4h;   或  ISP_ADDRL       EQU     0E4h
    ISP_CMD         DATA    0E5h;   或  ISP_CMD         EQU     0E5h
    ISP_TRIG        DATA    0E6h;   或  ISP_TRIG        EQU     0E6h
    ISP_CONTR       DATA    0E7h;   或  ISP_CONTR       EQU     0E7h
;定义 ISP/IAP 命令及等待时间
    ISP_IAP_BYTE_READ       EQU  1      ;字节读
    ISP_IAP_BYTE_PROGRAM    EQU  2      ;字节编程,前提是该字节是空，0FFh
    ISP_IAP_SECTOR_ERASE    EQU  3      ;扇区擦除,要某字节为空，要擦一扇区
    WAIT_TIME       EQU  0              ;设置等待时间,40MHz 以下 0,20M 以下 1,
                                        ;               10MHz 以下 2,5M 以下 3
;字节读
    MOV  ISP_ADDRH,     #BYTE_ADDR_HIGH     ;   送地址高字节
    MOV  ISP_ADDRL,     #BYTE_ADDR_LOW      ;   送地址低字节
    CLR  EA         ;   关中断,此时各中断请求，会被挂起，一开中断，立即响应
;加入软件陷阱判断，如为非法状态，则让单片机进入掉电模式或软复位
    MOV  ISP_CONTR,     #WAIT_TIME          ;   设置等待时间
    ORL  ISP_CONTR,     #10000000B          ;   允许 ISP/IAP 操作
    MOV  ISP_CMD,       #ISP_IAP_BYTE_READ  ;   送字节读命令
;加入软件陷阱判断，如为非法状态，则让单片机进入掉电模式或软复位
    MOV  ISP_TRIG,      #46h        ;先送 46h,再送 B9h 到 ISP/IAP 触发寄存器
;加入软件陷阱判断，如为非法状态，则让单片机进入掉电模式或软复位
    MOV  ISP_TRIG,      #0B9h       ;送完 B9h 后，ISP/IAP 命令立即被触发起动
;CPU 等待 IAP 动作完成后，才会继续执行程序，要先关中断（EA），
;再送 46h,B9h 到 ISP/IAP 触发寄存器,起动 ISP/IAP 命令,关中断在触发之前即可
    NOP                             ;数据读出到 ISP_DATA 寄存器后，CPU 继续执行程序
    MOV  ISP_CONTR,     #00000000B  ;禁止 ISP/IAP 操作
    MOV  ISP_CMD,       #00000000B  ;去除 ISP/IAP 命令
    MOV  ISP_TRIG,      #00000000B  ;防止 ISP/IAP 命令误触发
    MOV  ISP_ADDRH,     #0FF        ;送地址高字节单元为 FF,防止误操作
    MOV  ISP_ADDRL,     #0FF        ;送地址低字节单元为 FF,防止误操作
    SETB EA     ;   开中断，CPU 处理完 ISP/IAP 动作即可开中断
    MOV  A,             ISP_DATA    ;将读出的数据送往 Acc
```

```
;字节编程，该字节为FFh/空时，可对其编程，否则不行,要先执行扇区擦除
    MOV  ISP_DATA,     #ONE_DATA          ;    送字节编程数据到ISP_DATA
    MOV  ISP_ADDRH,    #BYTE_ADDR_HIGH    ;    送地址高字节
    MOV  ISP_ADDRL,    #BYTE_ADDR_LOW     ;    送地址低字节
    CLR  EA        ;    关中断,此时各中断请求，会被挂起，一开中断，立即响应
;加入软件陷阱判断，如为非法状态，则让单片机进入掉电模式或软复位
    MOV  ISP_CONTR,    #WAIT_TIME         ;    设置等待时间
    ORL  ISP_CONTR,    #10000000B         ;    允许ISP/IAP操作
    MOV  ISP_CMD,      #ISP_IAP_BYTE_PROGRAM   ;送字节编程命令
;加入软件陷阱判断，如为非法状态，则让单片机进入掉电模式或软复位
    MOV  ISP_TRIG,     #46h      ;先送46h,再送B9h到ISP/IAP触发寄存器
;加入软件陷阱判断，如为非法状态，则让单片机进入掉电模式或软复位
    MOV  ISP_TRIG,     #0B9h     ;送完B9h后，ISP/IAP命令立即被触发起动
;CPU等待IAP动作完成后，才会继续执行程序，要先关中断（EA），
;再送46h,B9h到ISP/IAP触发寄存器,起动ISP/IAP命令,关中断在触发之前即可
    NOP                          ;字节编程成功后，CPU继续执行程序
    MOV  ISP_CONTR,    #00000000B ;禁止ISP/IAP操作
    MOV  ISP_CMD,      #00000000B ;去除ISP/IAP命令
    MOV  ISP_TRIG,     #00000000B ;防止ISP/IAP命令误触发
    MOV  ISP_ADDRH,    #0FF       ;送地址高字节单元为FF,防止误操作
    MOV  ISP_ADDRL,    #0FF       ;送地址低字节单元为FF,防止误操作
    SETB EA    ;    开中断，CPU处理完ISP/IAP动作即可开中断
```

```
;扇区擦除，没有字节擦除，只有扇区擦除，512 字节 / 扇区,每个扇区建议只用
;1 字节 /2 字节 /3 字节 /4 字节 ---128 字节 /256 字节 /512 字节,用得越少越方便
;如果要对某个扇区进行擦除，而其中有些字节的内容需要保护，则需将其先读到单片机
;内部的 RAM 中保存，再将该扇区擦除，然后再将保存的数据写回该扇区，所以每个扇区
;中用的字节数越少越好，操作起来越灵活越快(每个扇区只用 1-128 字节以内较方便)

    MOV  ISP_ADDRH,    #SECTOR_FIRST_BYTE_ADDR_HIGH     ;送扇区起始地址高字节
    MOV  ISP_ADDRL,    #SECTOR_FIRST_BYTE_ADDR_LOW      ;送扇区起始地址低字节
    CLR  EA        ;    关中断,此时各中断请求，会被挂起，一开中断，立即响应
;加入软件陷阱判断，如为非法状态，则让单片机进入掉电模式或软复位
    MOV  ISP_CONTR,    #WAIT_TIME      ;设置等待时间
    ORL  ISP_CONTR,    #10000000B      ;允许 ISP/IAP
    MOV  ISP_CMD,      #ISP_IAP_SECTOR_ERASE            ;送扇区擦除命令
;加入软件陷阱判断，如为非法状态，则让单片机进入掉电模式或软复位
    MOV  ISP_TRIG,     #46h       ;先送 46h,再送 B9h 到 ISP/IAP 触发寄存器
;加入软件陷阱判断，如为非法状态，则让单片机进入掉电模式或软复位
    MOV  ISP_TRIG,     #0B9h      ;送完 B9h 后，ISP/IAP 命令立即被触发起动
;CPU 等待 IAP 动作完成后，才会继续执行程序，要先关中断（EA），
;再送 46h,B9h 到 ISP/IAP 触发寄存器,起动 ISP/IAP 命令,关中断在触发之前即可
    NOP                           ;扇区擦除成功后，CPU 继续执行程序
    MOV  ISP_CONTR,    #00000000B      ;禁止 ISP/IAP 操作
    MOV  ISP_CMD,      #00000000B      ;去除 ISP/IAP 命令
    MOV  ISP_TRIG,     #00000000B      ;防止 ISP/IAP 命令误触发
    MOV  ISP_ADDRH,    #0FF            ;送地址高字节单元为 FF,防止误操作
    MOV  ISP_ADDRL,    #0FF            ;送地址低字节单元为 FF,防止误操作
```

```
;从用户应用程序区(AP 区)软件复位并切换到 ISP 程序区开始执行程序
    MOV  ISP_CONTR,    #01100000B      ;SWBS = 1(选择 ISP 区), SWRST = 1(软复位)

;从 ISP 程序区软件复位并切换到用户应用程序区(AP 区)开始执行程序
    MOV  ISP_CONTR,    #00100000B      ;SWBS = 0(选择 AP 区), SWRST = 1(软复位)

;使用 ISP/IAP 功能的朋友尽量给 13922805190（姚工）一个电话交流一下
;建议在打开 ISP 之前及在 ISP 触发送 46H,B9H 之前三个地方，各加些软件陷阱
;5V 单片机，应在 3.6V 以下时让其复位，3V 单片机，应在 2.4V 以下时让其复位
;在此电压以下，此时再用 ISP/IAP 功能，编程 / 擦除 Flash，不能保证能达到要求
;复位电路可选 MAX810,STC810,STC6345,STC6344，813L,706P 等
```

附录 A:

为什么少数用户的普通 8051 程序烧录后，不能运行

1．增强型的 STC89C58RD+，STC89C52RC 系列单片机内部扩展了 AUX-RAM.

型号	内部扩展RAM	占外部64K数据空间	MOVX @DPTR / xdata	MOVX @Ri / pdata
STC89C51RC	256 字节	0000H - 00FFH	地址>=100H,才访问外部	只访问内部扩展RAM
STC89LE51RC	256 字节	0000H - 00FFH	地址>=100H,才访问外部	只访问内部扩展RAM
STC89C52RC	256 字节	0000H - 00FFH	地址>=100H,才访问外部	只访问内部扩展RAM
STC89LE52RC	256 字节	0000H - 00FFH	地址>=100H,才访问外部	只访问内部扩展RAM
STC89C53RC	256 字节	0000H - 00FFH	地址>=100H,才访问外部	只访问内部扩展RAM
STC89LE53RC	256 字节	0000H - 00FFH	地址>=100H,才访问外部	只访问内部扩展RAM
STC89C54RD+	1024 字节	0000H - 03FFH	地址>=400H,才访问外部	只访问内部扩展RAM
STC89LE54RD+	1024 字节	0000H - 03FFH	地址>=400H,才访问外部	只访问内部扩展RAM
STC89C58RD+	1024 字节	0000H - 03FFH	地址>=400H,才访问外部	只访问内部扩展RAM
STC89LE58RD+	1024 字节	0000H - 03FFH	地址>=400H,才访问外部	只访问内部扩展RAM
STC89C516RD+	1024 字节	0000H - 03FFH	地址>=400H,才访问外部	只访问内部扩展RAM
STC89LE516RD+	1024 字节	0000H - 03FFH	地址>=400H,才访问外部	只访问内部扩展RAM
STC89LE516AD	256 字节		0000-FFFFH,均访问外部	只访问内部扩展RAM
STC89LE516X2	256 字节		0000-FFFFH,均访问外部	只访问内部扩展RAM

STC89C52RC / STC89C58RD+ 系列

如果用户的单片机 P0 / P2 口是作为普通 I/O 口(输入 / 输出口)用的，无冲突

如果用户的单片机 P0 / P2 口是作为总线扩展用的，外部扩展器件的地址在内部扩展 RAM 之上，无冲突

如果用户的单片机 P0 / P2 口是作为总线扩展用的，访问外部扩展器件的地址在内部扩展 RAM 的范围内，则访问的是内部扩展的 AUX-RAM,所以有些系统用户要禁止内部扩展 RAM

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset value
AUXR	8Eh	Auxiliary Register 0	-	-	-	-	-	-	EXTRAM	ALEOFF	xxxx,xx00

将 AUXR 寄存器的 EXTRAM 设为 1，即可禁止内部扩展的 AUX-RAM，使之与标准 8052 一样。

STC89LE516AD / STC89LE516X2 系列 AUXR 寄存器无 EXTRAM 控制位，通过寻址方式区分，“MOVX @Ri”固定访问内部扩展 RAM,“MOVX, @DPTR”固定访问外部扩展 64K 数据空间，详见相应部分说明。

2.晶振 / 时钟电路

STC89C52RC / STC89C58RD+ 现系列要求将 XTAL1 管脚接的电容去掉，XTAL2 管脚接的电容为 47pF,33pF,30pF，详见相应部分说明。

STC89LE516AD / STC89LE516X2 系列 XTAL1 / XTAL2 管脚的电容接 10pF

3.复位电路

电阻 / 电容复位的值为 电阻 = 10k，电容 = 10uF。如为用外部专用芯片复位，RESET 管脚(复位脚)不要加任何上拉 / 下拉电阻。详见相应部分说明

附录 B:

STC89LE516AD 系列单片机指南

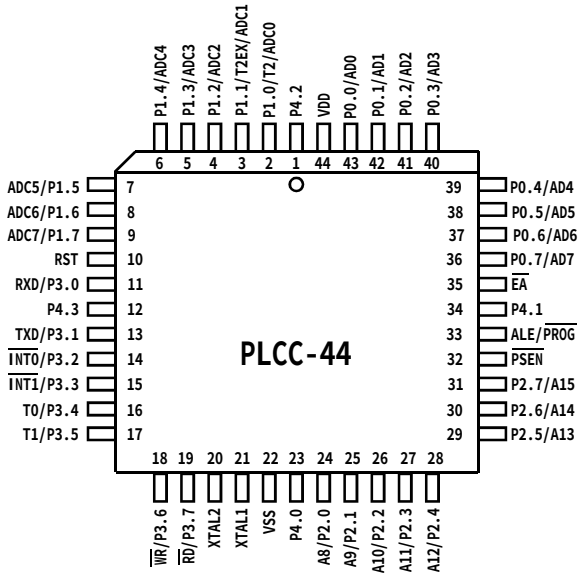
特殊功能寄存器映像 SFR Mapping

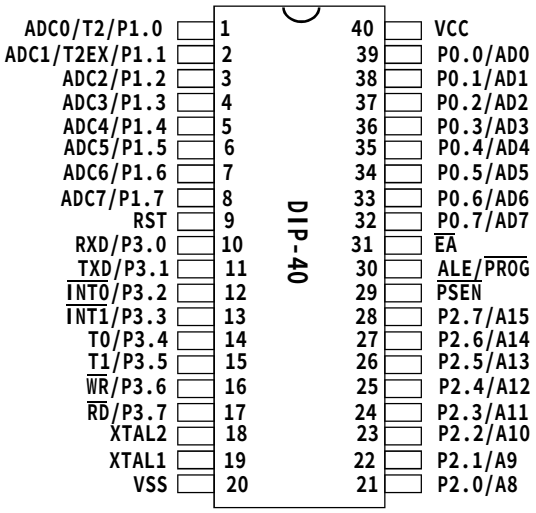
STC89LE516AD,STC89LE58AD,STC89LE54AD,STC89LE52AD 为单倍速

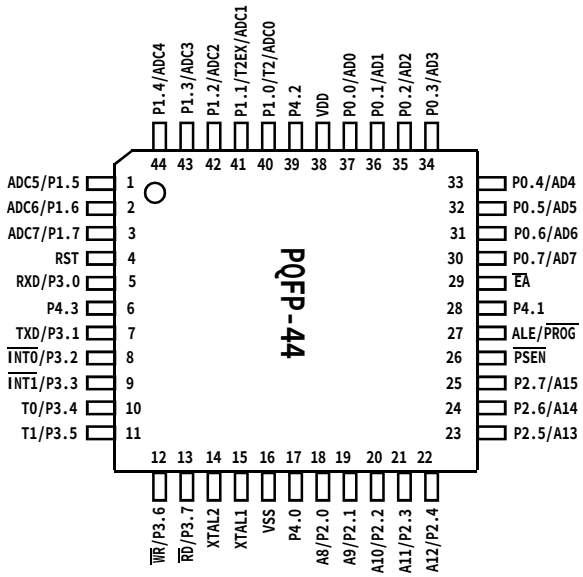
因 AD 系列用户 ISP 设不了双倍速，故有一 STC89LE516AD 的双倍速版本(6T)STC89LE516X2 是同一芯片，只是出厂时就设为双倍速了。

	Bit Addressable	Non Bit Addressable							
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8h									FFh
F0h	B 0000,0000								F7h
E8h									EFh
E0h	ACC 0000,0000								E7h
D8h									DFh
D0h	PSW 0000,0000								D7h
C8h	T2CON 0000,0000	T2MOD xxxx,xx00	RCAP2L 0000,0000	RCAP2H 0000,0000	TL2 0000,0000	TH2 0000,0000			CFh
C0h	P4 xxxx,1111					ADC_CONTR xxx0,0000	ADC_DATA xxxx,xxxx		C7h
B8h	IP x000,0000	SADEN 0000,0000							BFh
B0h	P3 1111,1111							IPH x000,0000	B7h
A8h	IE 0000,0000	SADDR 0000,0000							AFh
A0h	P2 1111,1111		AUXR1 xxxx,xxx0						A7h
98h	SCON 0000,0000	SBUF xxxx,xxxx							9Fh
90h	P1 1111,1111							P1_ADC_EN 0000,0000	97h
88h	TCON 0000,0000	TMOD 0000,0000	TL0 0000,0000	TL1 0000,0000	TH0 0000,0000	TH1 0000,0000	AUXR xxxx,xxx0		8Fh
80h	P0 1111,1111	SP 0000,0111	DPL 0000,0000	DPH 0000,0000				PCON 0xx1,0000	87h
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

STC89LE516AD / X2 系列单片机 管脚图

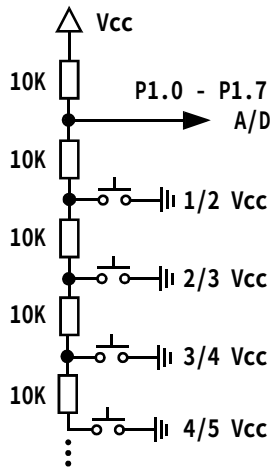






A/D 转换典型应用线路：

按键扫描



工作电压：3.6V - 2.4V， 可低至 2.4V - 1.9V

I/O 口： P0,P2,P3,P4 口可以承受 5V 输入。

P1 口不要接到 5V 系统，只能接 3.3V 系统.

P4 口： 地址在 C0H，P4.3 - P4.0

A/D 转换在 P1 口，P1.0 - P1.7 八路。

1.STC89LE516AD 系列单片机扩展 AUX-RAM 的使用

STC89LE516AD/X2 系列单片机内部的 RAM 为 512 字节(256+256),即扩展了 256 字节的 AUX-RAM,其访问方式为:

汇编语言: (访问内部扩展的 256 字节 AUX-RAM)

```
MOVX  @Ri,  A   ;  将累加器 A 的值送至 @Ri 指向的单元, i = 0, 1
MOVX  A,    @Ri ;  将 @Ri 指向的单元的值读到累加器 A, i = 0, 1
```

STC89LE516AD/X2 系列单片机用“MOVX A, @Ri”,“MOVX A, @Ri”指令访问不到外部 64K 数据空间。

汇编语言: (访问外部 64K 数据空间)

```
MOVX  @DPTR,  A     ;  将累加器 A 的值送至 @DPTR 指向的单元
MOVX  A,      @DPTR ;  将 @DPTR 指向的单元的值读到累加器 A
```

STC89LE516AD/X2 系列单片机用“MOVX A, @DPTR”,“MOVX A, @DPTR”指令访问不到内部扩展的 256 字节 AUX-RAM。

C 语言:

用 pdata 声明的变量访问单片机内部扩展的 256 字节 AUX-RAM
用 xdata 声明的变量访问单片机外部 64K 数据空间

2.双数据指针 及 AUXR1 寄存器

AUXR1	A2h	Auxiliary Register 1	-	-	-	-	-	-	-	DPS	xxxx,xxx0

DPS = 0 时选择 DPTR0, DPS = 1 时选择 DPTR1
可以用 “INC AUXR1” 快速切换 DPTR0 / DPTR1

3.禁止 ALE 输出 及 AUXR1 寄存器

AUXR	8Eh	Auxiliary Register 0	-	-	-	-	-	-	-	ALEOFF	xxxx,xxx0

ALEOFF = 0, ALE 信号正常输出。
ALEOFF = 1, 禁止 ALE 信号输出。但在访问外部数据空间及外部程序空间时有信号输出。

4.中断 及 中断优先级控制寄存器 IP / IPH

中断与普通 8052 完全兼容, 优先级可设为 4 级,通过增加的 IPH 寄存器

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
IE	A8h	Interrupt Enable	EA	-	ET2	ES	ET1	EX1	ET0	EX0	0000,0000
IP	B8h	Interrupt Priority Low	-	-	PT2	PS	PT1	PX1	PT0	PX0	xx00,0000
IPH	B7h	Interrupt Priority High	-	-	PT2H	PSH	PT1H	PX1H	PT0H	PX0H	xx00,0000

Interrupt Source 中断源	Vector Address 中断向量地址	Polling Sequence 中断查询次序	中断优先级设置	优先级0 最低	优先级1	优先级2	优先级3 最高	Interrupt Request 中断请求
/INT0	0003H	1	PX0H,PX0	0,0	0,1	1,0	1,1	IE0
Timer 0	000BH	2	PT0H,PT0	0,0	0,1	1,0	1,1	TF0
/INT1	0013H	3	PX1H,PX1	0,0	0,1	1,0	1,1	IE1
Timer 1	001BH	4	PT1H,PT1	0,0	0,1	1,0	1,1	IF1
UART	0023H	5	PSH, PS	0,0	0,1	1,0	1,1	RI + TI
Timer 2	002BH	6	PT2H,PT2	0,0	0,1	1,0	1,1	TF2 + EXF2

4.A/D 及 A/D转换寄存器 ADC_CONTR/ADC_DATA/P1_ADC_EN

STC89LE516AD/X2 在 P1 口，有 8 位精度的高速 A/D 转换器，P1.7 - P1.0 共 8 路电压输入型，可做按键扫描，电池电压检测，频谱检测等。17 个机器周期可完成一次转换，时钟在 40MHz 以下时。

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
P1_ADC_EN	97h	允许P1.x成为A/D口	ADC_P17	ADC_P16	ADC_P15	ADC_P14	ADC_P13	ADC_P12	ADC_P11	ADC_P10	0000,0000
ADC_CONTR	C5h	A/D 转换控制寄存器	-	-	-	ADC_FLAG	ADC_START	CHS2	CHS1	CHS0	xxx0,0000
ADC_DATA	C6h	A/D 转换结果寄存器	-	-	-	-	-	-	-	-	0000,0000

P1_ADC_EN 特殊功能寄存器： P1.x 作为 A/D 转换输入通道来用允许特殊功能寄存器

允许P1.x成为A/D口	ADC_P17	ADC_P16	ADC_P15	ADC_P14	ADC_P13	ADC_P12	ADC_P11	ADC_P10	0000,0000

相应位为“1”时，对应的 P1.x 口作为 A/D 转换使用，内部上拉电阻自动断开

ADC_CONTR 特殊功能寄存器： A/D 转换控制特殊功能寄存器

A/D 转换控制寄存器	-	-	-	ADC_FLAG	ADC_START	CHS2	CHS1	CHS0	xxx0,0000

CHS2 / CHS1 / CHS0：模拟输入通道选择，CHS2 / CHS1 / CHS0

CHS2	CHS1	CHS0	Analog Input Channel Select 模拟输入通道选择
0	0	0	选择 P1.0 作为 A/D 输入来用
0	0	1	选择 P1.1 作为 A/D 输入来用
0	1	0	选择 P1.2 作为 A/D 输入来用
0	1	1	选择 P1.3 作为 A/D 输入来用
1	0	0	选择 P1.4 作为 A/D 输入来用
1	0	1	选择 P1.5 作为 A/D 输入来用
1	1	0	选择 P1.6 作为 A/D 输入来用
1	1	1	选择 P1.7 作为 A/D 输入来用

ADC_START： 模拟 / 数字转换(ADC)启动控制位，设置为“1”时，开始转换
ADC_FLAG： 模拟 / 数字转换结束标志位，当 A/D 转换完成后，ADC_FLAG = 1。

ADC_DATA 特殊功能寄存器： A/D 转换结果特殊功能寄存器

A/D 转换结果寄存器	-	-	-	-	-	-	-	-	0000,0000

模拟 / 数字转换结果计算公式如下： 结果 = 256 x Vin / Vcc

Vin 为模拟输入通道输入电压，Vcc 为单片机实际工作电压，用单片机工作电压作为模拟参考电压。

```
/* ------------ 宏晶科技    2005/1/5       A/D 转换 C 语言示例 ------------ */
/* ---Mobile: 13922805190, Tel: 0755 - 82908285,  Fax: 0755 - 82944243  */
/* ----Website: www.mcu-memory.com   Email: support@dsp-memory.com  --- */

//   ADC DEMO 程序演示 STC89LE516AD/X2 系列 MCU 的 A/D 转换功能。 时钟   11.0592MHz
//  转换结果以 16 进制形式输出到串行口，可以用串行口调试程序观察输出结果。

#include <reg52.H>
#include <intrins.H>

//定义与 ADC 有关的特殊功能寄存器
sfr   P1_ADC_EN       =    0x97;              // A/D 转换功能允许寄存器
sfr   ADC_CONTR       =    0xC5;              // A/D 转换控制寄存器
sfr   ADC_DATA        =    0xC6;              // A/D 转换结果寄存器

typedef    unsigned char   INT8U;
typedef    unsigned int    INT16U;

void delay(INT8U delay_time)            // 延时函数
{
    INT8U       n;
    INT16U      m;
    for (n=0; n<delay_time; n++)
    {
        for (m=0; m<10000; m++);
    }
}

void initiate_RS232 (void)          // 串口初始化
{
    ES = 0;                         // 禁止串口中断
    SCON = 0x50;                    // 0101,0000  8 位数据位， 无奇偶校验
    T2CON = 0x34;                    // 0011，0100， 由 T2 作为波特率发生器

    RCAP2H = 0xFF;                  // 时钟 11.0592MHz， 9600 波特率
    RCAP2L = 0xDB;

    ES = 1;                         // 允许串口中断
}

void Send_Byte(INT8U one_byte)      // 发送一个字节
{
    TI = 0;                         // 清零串口发送中断标志
    SBUF = one_byte;
    while (TI == 0);
    TI = 0;                         // 清零串口发送中断标志
}
```

```
INT8U get_AD_result(INT8U channel)
{
    INT8U AD_finished      =    0;            // 存储 A/D 转换标志
    ADC_DATA    =     0;
    ADC_CONTR  = channel;                  // 选择 A/D 当前通道
    delay(1);                              //使输入电压达到稳定
    ADC_CONTR |= 0x08;                     //0000,1000 令 ADC_START = 1, 启动A/D转换
    AD_finished = 0;
    while ( AD_finished == 0 )             // 等待A/D转换结束
    {
        AD_finished = (ADC_CONTR & 0x10);  //0001,0000, ADC_FLAG ==1 测试A/D转换结束否
    }
    ADC_CONTR &= 0xF7;                    //1111,0111 令 ADC_START = 0, 关闭A/D转换,
    return (ADC_DATA);                    // 返回 A/D 转换结果
}
void main()
{
    initiate_RS232();
    P1     =    P1    |    0x63;   // 0110,0011,要设置为 A/D 转换的P1.x口,先设为高
    P1_ADC_EN = 0x63;      //0110,0011, P1 的P1.0,P1.1,P1.5,P1.6设置为 A/D 转换输入脚
                           // 断开P1.0,P1.1,P1.5,P1.6内部上拉电阻
    while(1)
    {
        Send_Byte(get_AD_result(0));  //P1.0 为 A/D 当前通道, 测量并发送结果
        delay(0x200);

        Send_Byte(get_AD_result(1));  //P1.1 为 A/D 当前通道, 测量并发送结果
        delay(0x200);

        Send_Byte(get_AD_result(5));  //P1.5 为 A/D 当前通道, 测量并发送结果
        delay(0x200);

        Send_Byte(get_AD_result(6));  //P1.6 为 A/D 当前通道, 测量并发送结果
        delay(0x200);

        Send_Byte(0);                 // 连续发送 4 个 00H, 便于观察输出显示
        Send_Byte(0);
        Send_Byte(0);
        Send_Byte(0);

        delay(0x200);                 // 延时
        delay(0x200);
        delay(0x200);
        delay(0x200);
        delay(0x200);
        delay(0x200);
    }
}
```

附录 C:

STC89C51 RC / RD+ 系列 单片机 ISP (DIY)

自己动手写自己的 ISP,写自己的远程升级程序还需了解的信息,只提供给策略性伙伴

用户程序区空间和 ISP 程序区空间:

型号	用户应用程序区	ISP 引导区空间(ISP Space),从ISP区启动MCU才有IAP功能			
	AP Space 地址	0 K	1K(保留IAP)	2 K(保留IAP)	4K(保留IAP)
STC89C51RC	0000H - 0FFFH	目前版本禁止ISP/IAP操作	3800H - 3BFFH	3400H - 3BFFH	2C00H - 3BFFH
STC89LE51RC	0000H - 0FFFH		3800H - 3BFFH	3400H - 3BFFH	2C00H - 3BFFH
STC89C52RC	0000H - 1FFFH		3800H - 3BFFH	3400H - 3BFFH	2C00H - 3BFFH
STC89LE52RC	0000H - 1FFFH		3800H - 3BFFH	3400H - 3BFFH	2C00H - 3BFFH

型号	用户应用程序区	ISP 引导区空间(ISP Space),从ISP区启动MCU才有IAP功能			
	AP Space 地址	0 K	1K(保留IAP)	2 K(保留IAP)	4K(保留IAP)
STC89C53RC	0000H - 3BFFH	目前版本禁止ISP/IAP操作			
STC89LE53RC	0000H - 3BFFH				
STC89C53RC	0000H - 37FFH		3800H - 3BFFH		
STC89LE53RC	0000H - 37FFH		3800H - 3BFFH		
STC89C53RC	0000H - 33FFH			3400H - 3BFFH	
STC89LE53RC	0000H - 33FFH			3400H - 3BFFH	
STC89C53RC	0000H - 2BFFH				2C00H - 3BFFH
STC89LE53RC	0000H - 2BFFH				2C00H - 3BFFH

型号	用户应用程序区	ISP 引导区空间(ISP Space),从ISP区启动MCU才有IAP功能			
	AP Space 地址	0 K	1K(保留IAP)	2 K(保留IAP)	4K(保留IAP)
STC89C54RD+	0000H - 3FFFH	目前版本禁止ISP/IAP操作	F800H - FBFFH	F400H - FBFFH	EC00H - FBFFH
STC89LE54RD+	0000H - 3FFFH		F800H - FBFFH	F400H - FBFFH	EC00H - FBFFH
STC89C58RD+	0000H - 7FFFH		F800H - FBFFH	F400H - FBFFH	EC00H - FBFFH
STC89LE58RD+	0000H - 7FFFH		F800H - FBFFH	F400H - FBFFH	EC00H - FBFFH

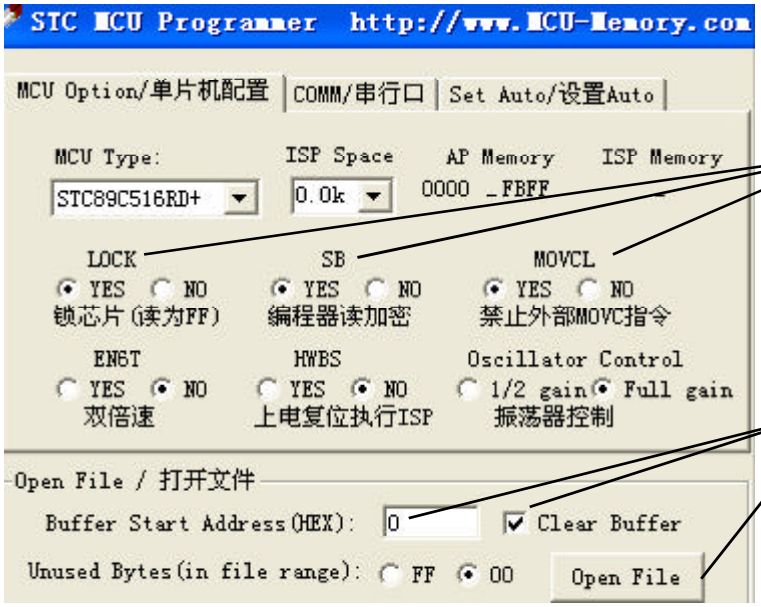
型号	用户应用程序区	ISP 引导区空间(ISP Space),从ISP区启动MCU才有IAP功能			
	AP Space 地址	0 K	1K(保留IAP)	2 K(保留IAP)	4K(保留IAP)
STC89C516RD+	0000H - FBFFH	目前版本禁止ISP/IAP操作			
STC89LE516RD+	0000H - FBFFH				
STC89C516RD+	0000H - F7FFH		F800H - FBFFH		
STC89LE516RD+	0000H - F7FFH		F800H - FBFFH		
STC89C516RD+	0000H - F3FFH			F400H - FBFFH	
STC89LE516RD+	0000H - F3FFH			F400H - FBFFH	
STC89C516RD+	0000H - EBFFH				EC00H - FBFFH
STC89LE516RD+	0000H - EBFFH				EC00H - FBFFH

动手写自己的 ISP 还必需要的工具，只提供给策略性伙伴

-----STC89C51RC / RD+ 系列单片机专用编程工具（200 元）

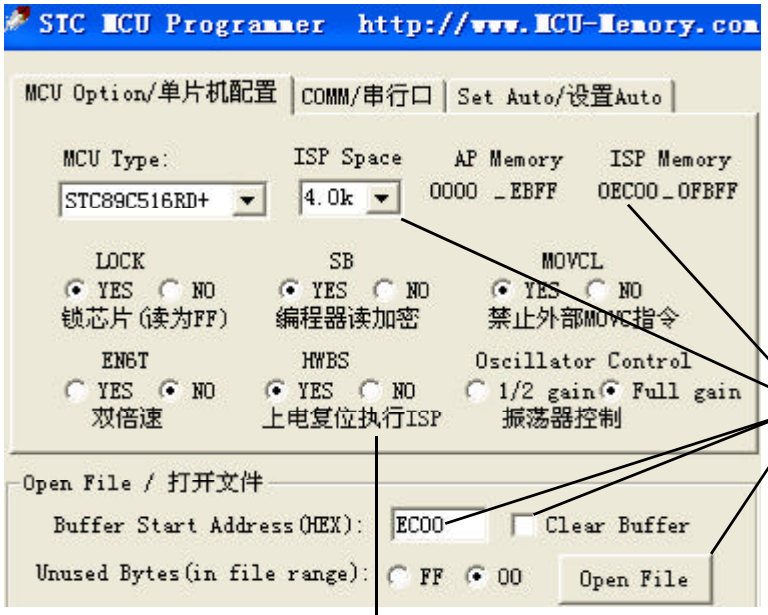
STC89C51RC / RD+ 系列单片机出厂时一般都固化有 ISP 引导码程序，如只烧录普通的用户程序，则只需将P3.0/P3.1经过RS-232转换器连到电脑的RS-232串口，通过STC-ISP下载软件打开用户程序下载就可以了。如果不用 STC 的 ISP 程序而要编写自己的 ISP 程序，则需要 STC89C51RC / RD+ 系列单片机专用编程工具，才可将用户的 ISP 程序烧录进单片机内部，软件使用 STC MCU Programmer 编程工具。

方式一：用户主应用程序和 ISP 程序一起烧录，上电复位后程序从 ISP 区开始运行



一．三个加密项全部加密

二．缓冲区从 0 开始，清缓冲区调入用户主应用程序



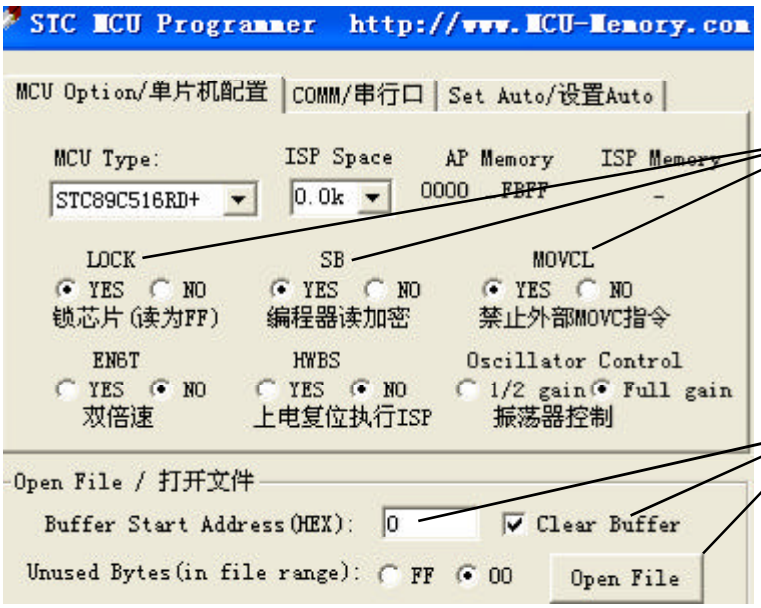
三．选定 ISP 空间从那里开始，不清缓冲区调入 ISP 程序，保留已调入的程序，用户 ISP空间可选0/1K/2K/4K

四．选择从 ISP 区启动

五．烧录程序

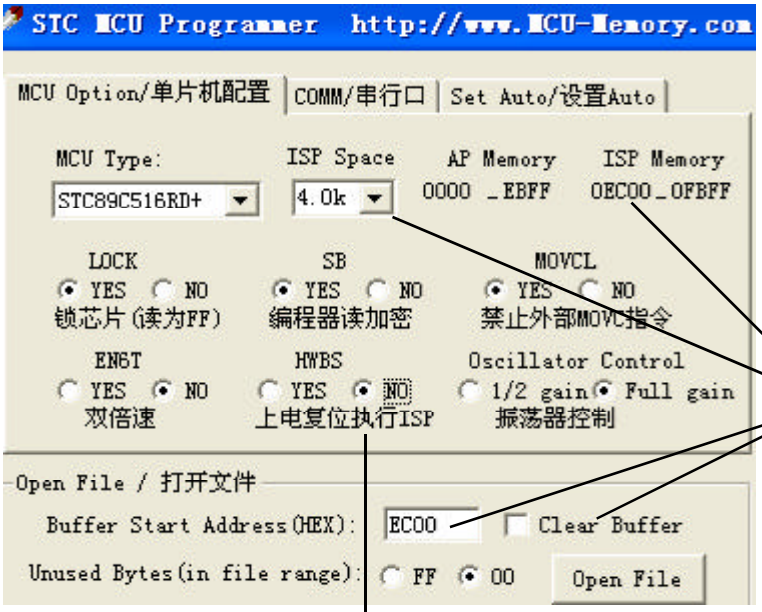
也可直接调入 ISP 程序至 ISP 区，选择从 ISP 程序区启动，再烧录。用户主应用程序由用户自己编的 ISP 程序下载。还可以把 ISP 区当成用户程序区，把余下的空间当成 Data Flash 数据空间。如 STC89C516RD+,4K ISP 区当用户程序使用，余下 59K 当 EEPROM 用。

方式二：用户主应用程序和 ISP 程序一起烧录，上电复位后程序从用户主应用程序区开始运行(但现版本单片机 ISP/IAP 就会无效，只能当成无 ISP/IAP 的单片机使用，故此法对于写自己的 ISP 程序来说无效)



一．三个加密项全部加密

二．缓冲区从 0 开始，清缓冲区调入用户主应用程序



三．选定 ISP 空间从那里开始，不清缓冲区调入用户自己的 ISP 程序，保留已调入的程序

四．选择从用户主应用程序区启动，现版本芯片会无 ISP/IAP 功能，故应设置成从 ISP 区启动。

五．烧录程序

已加过密的单片机，或已设置过 ISP 空间为 1k/2k/4k，上电复位从 ISP 空间启动的单片机，重新烧录程序时，要先将整个芯片擦除，再停电(单片机电放光后)，再上电，再重烧程序，才会写对。擦除整个芯片后，重新允许读出芯片内容的状态，是停电再上电后才能生效。

附录 D： ISP Demo(演示版)软件(*.hex)及通信协议

www.mcu-memory.com网站上也会提供用户可直接使用的单片机ISP-Demo软件(*.hex),可由用户直接用 STC 专用编程器烧入即可，在 STC-ISP.exe 软件中选择 “ ISP Demo ”功能项下载程序即可。将来的 STC 专用编程器软件将会自带此单片机 ISP 程序功能,格式是偶校验，8 位数据位，1 位停止位，用户要根据单片机工作时钟，确定用那种合理的波特率。
(说明：出厂时单片机一般都已烧录有标准的 ISP 文件，不过那个协议不公开)

STC89C51RC STC89LE51RC	115,200 bps	57,600 bps	38,400 bps	19,200 bps	9,600 bps	4,800 bps	2,400 bps	1,200 bps
40MHz	51_40M_ 115200.hex	51_40M_ 57600.hex	51_40M_ 38400.hex	51_40M_ 19200.hex	51_40M_ 9600.hex	51_40M_ 4800.hex	51_40M_ 2400.hex	51_40M_ 1200.hex
33MHz	51_33M_ 115200.hex	51_33M_ 57600.hex	51_33M_ 38400.hex	51_33M_ 19200.hex	51_33M_ 9600.hex	51_33M_ 4800.hex	51_33M_ 2400.hex	51_33M_ 1200.hex
27MHz	51_27M_ 115200.hex	51_27M_ 57600.hex	51_27M_ 38400.hex	51_27M_ 19200.hex	51_27M_ 9600.hex	51_27M_ 4800.hex	51_27M_ 2400.hex	51_27M_ 1200.hex
24MHz	51_24M_ 115200.hex	51_24M_ 57600.hex	51_24M_ 38400.hex	51_24M_ 19200.hex	51_24M_ 9600.hex	51_24M_ 4800.hex	51_24M_ 2400.hex	51_24M_ 1200.hex
22.1184MHz	51_22.1184M_ 115200.hex	51_22.1184M_ 57600.hex	51_22.1184M_ 38400.hex	51_22.1184M_ 19200.hex	51_22.1184M_ 9600.hex	51_22.1184M_ 4800.hex	51_22.1184M_ 2400.hex	51_22.1184M_ 1200.hex
20MHz	51_20M_ 115200.hex	51_20M_ 57600.hex	51_20M_ 38400.hex	51_20M_ 19200.hex	51_20M_ 9600.hex	51_20M_ 4800.hex	51_20M_ 2400.hex	51_20M_ 1200.hex
18.432MHz	51_18.432M_ 115200.hex	51_18.432M_ 57600.hex	51_18.432M_ 38400.hex	51_18.432M_ 19200.hex	51_18.432M_ 9600.hex	51_18.432M_ 4800.hex	51_18.432M_ 2400.hex	51_18.432M_ 1200.hex
12MHz	51_12M_ 115200.hex	51_12M_ 57600.hex	51_12M_ 38400.hex	51_12M_ 19200.hex	51_12M_ 9600.hex	51_12M_ 4800.hex	51_12M_ 2400.hex	51_12M_ 1200.hex
11.0592MHz	51_11.0592M_ 115200.hex	51_11.0592M_ 57600.hex	51_11.0592M_ 38400.hex	51_11.0592M_ 19200.hex	51_11.0592M_ 9600.hex	51_11.0592M_ 4800.hex	51_11.0592M_ 2400.hex	51_11.0592M_ 1200.hex

STC89C52RC STC89LE52RC	115,200 bps	57,600 bps	38,400 bps	19,200 bps	9,600 bps	4,800 bps	2,400 bps	1,200 bps
40MHz	52_40M_ 115200.hex	52_40M_ 57600.hex	52_40M_ 38400.hex	52_40M_ 19200.hex	52_40M_ 9600.hex	52_40M_ 4800.hex	52_40M_ 2400.hex	52_40M_ 1200.hex
33MHz	52_33M_ 115200.hex	52_33M_ 57600.hex	52_33M_ 38400.hex	52_33M_ 19200.hex	52_33M_ 9600.hex	52_33M_ 4800.hex	52_33M_ 2400.hex	52_33M_ 1200.hex
27MHz	52_27M_ 115200.hex	52_27M_ 57600.hex	52_27M_ 38400.hex	52_27M_ 19200.hex	52_27M_ 9600.hex	52_27M_ 4800.hex	52_27M_ 2400.hex	52_27M_ 1200.hex
24MHz	52_24M_ 115200.hex	52_24M_ 57600.hex	52_24M_ 38400.hex	52_24M_ 19200.hex	52_24M_ 9600.hex	52_24M_ 4800.hex	52_24M_ 2400.hex	52_24M_ 1200.hex
22.1184MHz	52_22.1184M_ 115200.hex	52_22.1184M_ 57600.hex	52_22.1184M_ 38400.hex	52_22.1184M_ 19200.hex	52_22.1184M_ 9600.hex	52_22.1184M_ 4800.hex	52_22.1184M_ 2400.hex	52_22.1184M_ 1200.hex
20MHz	52_20M_ 115200.hex	52_20M_ 57600.hex	52_20M_ 38400.hex	52_20M_ 19200.hex	52_20M_ 9600.hex	52_20M_ 4800.hex	52_20M_ 2400.hex	52_20M_ 1200.hex
18.432MHz	52_18.432M_ 115200.hex	52_18.432M_ 57600.hex	52_18.432M_ 38400.hex	52_18.432M_ 19200.hex	52_18.432M_ 9600.hex	52_18.432M_ 4800.hex	52_18.432M_ 2400.hex	52_18.432M_ 1200.hex
12MHz	52_12M_ 115200.hex	52_12M_ 57600.hex	52_12M_ 38400.hex	52_12M_ 19200.hex	52_12M_ 9600.hex	52_12M_ 4800.hex	52_12M_ 2400.hex	52_12M_ 1200.hex
11.0592MHz	52_11.0592M_ 115200.hex	52_11.0592M_ 57600.hex	52_11.0592M_ 38400.hex	52_11.0592M_ 19200.hex	52_11.0592M_ 9600.hex	52_11.0592M_ 4800.hex	52_11.0592M_ 2400.hex	52_11.0592M_ 1200.hex

```
;                    STC ISP DEMO 程序通讯协议

;STC89LE51RC，STC89LE52RC，STC89LE53RC，STC89LE54RD+，STC89LE58RD+，STC89LE516RD+
;STC89C51RC，STC89C52RC，STC89C53RC，STC89C54RD+，STC89C58RD+，STC89C516RD+
;
;1. 功能
;        STC ISP DEMO 程序以固定的波特率与上位机(PC 机)通讯，将上位机
;    传来的程序代码或数据烧录到 MCU 的用户应用程序区中(以下简称 AP 区)和
;     Data 区。
;
;1.1 应用范围
;     本协议仅适用于 STC89C/LE5xRC 和 STC89C/LE5xxRD+ 系列单片机。
;1.2 使用方法
;   1) 将 STC ISP DEMO 程序烧录到 MCU 的 ISP 区，并设置为上电复位后从
;        ISP 区开始运行。
;    2) 先运行上位机程序，不断地发送问讯帧，然后给 MCU 上电。
;    3) MCU 收到问讯帧后回发问讯应答帧，讯问应答反复 3 次双方连接成功。
;    4) 上位机发送一系列命令，完成下载工作。
;    5) 下载完毕，PC 发送切换命令，MCU 软复位切换到 AP 程序运行。
;
;        STC-ISP (2.9 Beta3 版本以上)下载软件提供了 STC ISP DEMO 程序的上位机
;    功能，该软件可到 www.MCU-Memory.COM 网站下载。
;        MCU 上电复位后检测串口是否有上位机发来的问讯帧，若没有就立刻软复位
;   切换到 AP 程序运行。也可以参照 STC 下载软件中"用户自定义下载" 功能，
;   从应用程序软复位切换到 ISP 进行下载，而不是每次都要上电复位(冷启动)
;    进入 ISP 程序。
;
;2.   帧格式。
;     所有的通讯均由 PC 机启动。
;
;2.1 PC 机命令帧格式。
;        5AH              0101,1010 起始字节 1 ----        ------
;        A5H              1010,0101 起始字节 2     |             |
;        len_H            帧长度高字节             |             |
;        len_L            帧长度低字节                           |
;        Command          命令字              帧校验和范围       |
;        data 0           数据区第 1 字节
;         ..                                   |        帧长度范围
;         ..                                   |
;         ..                                   |             |
;        data n-1         数据区第 n 字节     ----             |
;        check_sum_high   帧校验和高字节                       |
;        chenk_sum_low    帧校验和低字节                       |
;        16H              结束字节  ---------------------------
```

```
2.3  帧长度。
;     帧长度范围从起始字节1 至结束字节。
;
;2.4 校验和。
;     从起始字节1开始到校验和之前所有的字节相加。
;
;3.0 PC 命令及 MCU 应答
;     命令                          命令代码
;
;     问讯                           B0H     1011, 0000
;     软复位切换到用户程序          B1H     1011, 0001
;     擦除 AP、Data flash           B2H     1011, 0010
;     写数据到 AP、Data flash       B3H     1011, 0011
;     先擦除扇区再写数据            B4H     1011, 0100
;     读固件版本号                  B5H     1011, 0101
;
;3.1 问讯帧。
;
;3.1.1 PC 发送问讯帧。
;     序号  内容
;       0   5AH              起始字节 1
;       1   A5H              起始字节 2
;       2   00H              帧长度高字节
;       3   08H              帧长度低字节
;       4   B0H              问讯帧命令字
;       5   01H              校验和高字节 (5AH + A5H + 00H + 08H + B0H = 01B7H)
;       6   B7H              校验和低字节
;       7   16H              结束字节
;
;3.1.2 MCU 回发问讯应答帧。
;     序号  内容
;       0   5AH              起始字节 1
;       1   A5H              起始字节 2
;       2   00H              帧长度高字节
;       3   08H              帧长度低字节
;       4   B0H              问讯应答帧命令字
;       5   01H              校验和高字节 (5AH + A5H + 00H + 08H + B0H = 01B7H)
;       6   B7H              校验和低字节
;       7   16H              结束字节
;
;3.2 结束运行 ISP 程序，软复位切换到用户程序。
;      ISP 程序收到命令后启动软复位，切换到用户程序运行。
```

```
;3.2 结束运行 ISP 程序，软复位切换到用户程序。
;      ISP 程序收到命令后启动软复位，切换到用户程序运行。
;3.2.1 PC 命令
;    序号  内容
;      0   5AH              起始字节 1
;      1   A5H              起始字节 2
;      2   00H              帧长度高字节
;      3   08H              帧长度低字节
;      4   B1H              软复位切换到用户程序命令字
;      5   01H              校验和高字节 (5AH + A5H + 00H + 08H + B1H = 01B8H)
;      6   B8H              校验和低字节
;      7   16H              结束字节
;
;3.2.2 MCU 应答：
;    用"问讯应答帧"应答，见 3.1.2 MCU 问讯应答帧。
;
;3.3 擦除应用程序区和数据 Flash 区
;3.3.1 PC 命令：
;    序号  内容
;      0   5AH              起始字节 1
;      1   A5H              起始字节 2
;      2   00H              帧长度高字节
;      3   08H              帧长度低字节
;      4   B2H              擦除应用程序区命令字
;      5   01H              校验和高字节 (5AH + A5H + 00H + 08H + B2H = 01B9H)
;      6   B9H              校验和低字节
;      7   16H              结束字节
;
;3.3.2 MCU 应答：
;    用"问讯应答帧"应答，见 3.1.2 MCU 问讯应答帧。
```

```
;3.4 写数据到 flash AP 区
;3.4.1 PC 命令：
;    序号  内容
;      0   5AH            起始字节 1
;      1   A5H            起始字节 2
;      2   00H            帧长度高字节
;      3   xxH            帧长度低字节
;      4   B3H            写数据到 flash 命令字
;      5   00H            保留字节
;      6   00H            保留字节
;      7   address_H      目标首地址高字节
;      8   address_L      目标首地址低字节
;      9   00H            数据块长度高字节。
;     10   xxH            数据块长度低字节
;                        每次最多写 128 字节。写完一个 512 字节的扇区需分别
;                        发送 4 帧数据。某个扇区一旦写失败，须擦除该扇区后
;                        从头再写。
;     11   data 1         数据块第 1 字节
;
;     ..    ..
;
;     xx   data n        数据块第 n 字节
;     xx   xxH           校验和高字节
;     xx   xxH           校验和低字节
;     xx   16H           结束字
;
;3.4.2 MCU 应答：
;    序号  内容
;      0   5AH            起始字节 1
;      1   A5H            起始字节 2
;      2   00H            帧长度高字节
;      3   0AH            帧长度低字节
;      4   B3H            写数据到 flash 命令字
;      5   xxH            数据区校验和高字节，见 3.4.3 写数据到 flash AP 区后的校验
;      6   xxH            数据区校验和低字节，见 3.4.3 写数据到 flash AP 区后的校验
;      7   xxH            帧校验和高字节
;      8   xxH            帧校验和低字节
;      9   16H            结束字节
;
;3.4.3 写数据到 flash AP 区后的校验：
;        MCU 把本次接收到的应用程序和数据写入 Flash 后再将它们逐一读出来，计算它们的
;    累加和。累加和的校验区由 2.4.1 的目标首地址和数据块长度指明(PC 命令的 7,8,9,10 字节)
;    累加和为 2 个字节，发回 PC 供其判断写数据正确与否。
```

;3.5 擦除扇区后再写数据到 flash AP 区

; 除了命令字为 B4H 外，其它与写数据到 flash AP 区完全一致。

;

;3.6 读 MCU 代号、固件版本号命令。

;3.6.1 PC 命令：

;	序号	内容	
;	0	5AH	起始字节 1
;	1	A5H	起始字节 2
;	2	00H	帧长度高字节
;	3	08H	帧长度低字节
;	4	B5H	读 MCU 代号、固件版本号命令字
;	5	01H	帧校验和高字节
;	6	F9H	帧校验和低字节
;	7	16H	结束字节

;

;3.6.2 MCU 应答：

;	序号	内容	
;	0	5AH	起始字节 1
;	1	A5H	起始字节 2
;	2	00H	帧长度高字节
;	3	0CH	帧长度低字节
;	4	B5H	读 MCU 代号、固件版本号命令字
;	5	FIRMWARE_VERSION	固件版本号
;	6	MCU_TYPE_HIGH	MCU 代号高字节
;	7	MCU_TYPE_LOW	MCU 代号低字节
;	8	xxH	保留字节
;	9	xxH	帧校验和高字节
;	10	xxH	帧校验和低字节
;	11	16H	结束字节

附录E：如何实现运行中自定义下载,无仿真器时方便调试

自定义下载原理：STC-ISP.exe 软件

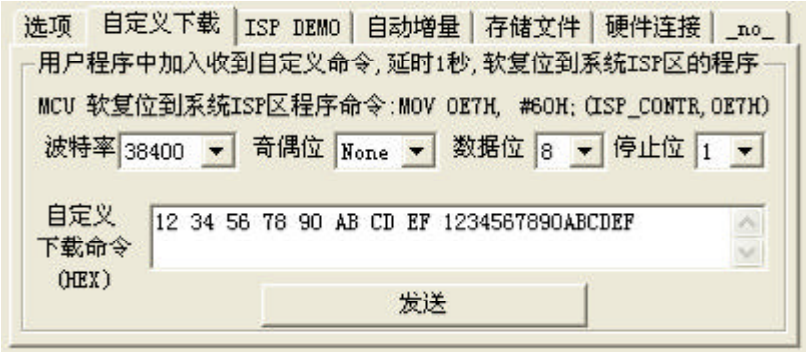
1.STC-ISP.exe 软件,可由用户设置，按 UART/RS-232 的格式向用户程序发送命令

波特率： 38400bps， 19200bps， 9600bps， 4800bps， 2400bps， 1200bps 等

奇偶校验位： 无， 偶校验， 奇校验

数据位几位： 8 位， 7 位，6 位，5 位， 等

停止位几位： 1 位， 1.5 位，2 位，等



2.需向用户程序发送的命令用户可在上图自定义下载命令输入窗口中输入(HEX)

命令之间建议加一个空格，也可不加，STC-ISP.exe 会处理，上例为发送自定义命令 12H,34H,56H,78H,0ABH,0CDH,0EFH， 12H,34H,56H,78H,0ABH,0CDH,0EFH

3.用户程序中应加入收到自定义下载命令，延时一秒，软复位到系统 ISP 程序区的程序

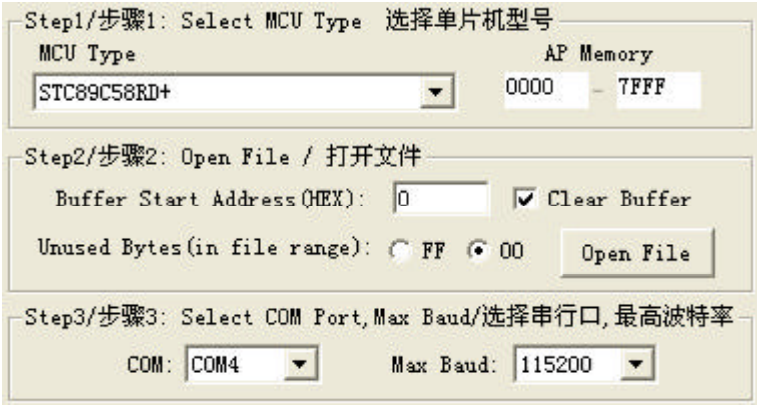
4.将以上含有接收自定义下载命令的用户程序先用老方法下载进 STC 单片机内部：

STC89C51RC,STC89C52RC,STC89C53RC,STC89C54RD+,STC89C58RD+,STC89C516RD+

STC89LE52RC,STC89LE53RC,STC89LE54RD+,STC89LE58RD+,STC89LE516RD+

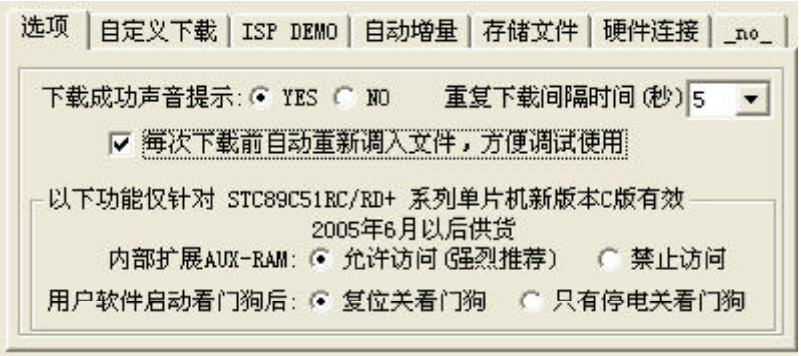
这样以上型号的 STC 单片机就具有了“不停电 / 运行中 / 自定义下载功能”

6.以下选好型号，打开文件...,在自定义下载中设置相关选项，选择“发送”即可，



STC-ISP.exe 在“发送”完用户自定义下载命令后,就会转去调用老的那一套下载命令,而不管用户单片机程序收到命令没有。不过这个转换有些电脑有时需要将近1S的时间,所以用户应用程序要延时1S，否则系统 ISP 程序收不到下载命令,又会回到用户应用程序。

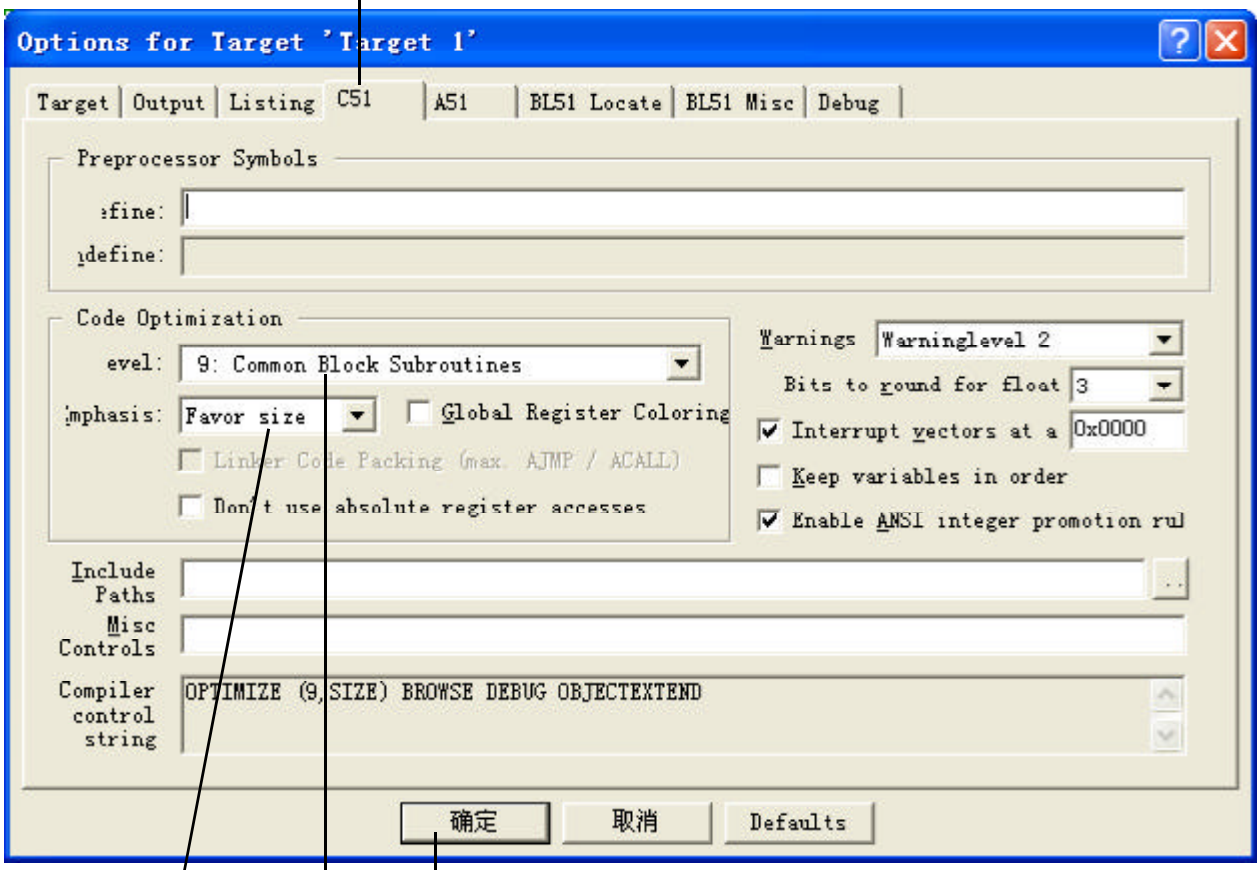
7.调试程序时，还可以在选项中选择“每次下载前自动重新调入文件”，这样你每次修改原文件并从新编译/汇编后生成的*.hex,*.bin文件就不要再次手工调入了，生产时不要用。



附录 F: Keil C51 高级语言编程的软件如何减少代码长度

1.在“Project”菜单中选择“Options for Target”

2.在“Options for Target”中选择“C51”



3.选择按空间大小， 9 级优化程序

4.重新编译程序即可。

附录G:(写给用STC89C51RC/RD+系列单片机做仿真器的朋友)

STC89C51 RC / RD+ 系列 单片机 做仿真器须知

1.对Flash的IAP字节读/字节编程/扇区擦除等待机器周期

设置等待时间			CPU 等待时间(机器周期)			
WT2	WT1	WT0	Read	Program	Sector Erase	Recommended System Clock
0	1	1	6	30	5471	5MHz
0	1	0	11	60	10942	10MHz
0	0	1	22	120	21885	20MHz
0	0	0	43	240	43769	40MHz

512Byte/Sector

2.程序在ISP区可对用户AP区进行IAP读/编程/扇区擦除

3.程序在用户AP区不可对ISP区进行IAP操作。

4.程序在ISP/AP区可对Data Flash区进行IAP操作。

5.可从ISP区软复位到AP区，也可从AP区软复位到ISP区。

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
ISP_DATA	E2h	ISP/IAP Flash Data Register									1111,1111
ISP_ADDRH	E3h	ISP/IAP Flash Address High									0000,0000
ISP_ADDRL	E4h	ISP/IAP Flash Address Low									0000,0000
ISP_CMD	E5h	ISP/IAP Flash Command Register	-	-	-	-	-	MS2	MS1	MS0	xxxx,x000
ISP_TRIG	E6h	ISP/IAP Flash Command Trigger									xxxx,xxxx
ISP_CONTR	E7h	ISP/IAP Control Register	ISPEN	SWBS	SWRST	-	-	WT2	WT1	WT0	000x,x000

```
;从用户应用程序区(AP区)软件复位并切换到ISP程序区开始执行程序
    MOV   ISP_CONTR,    #01100000B    ;SWBS = 1(选择ISP区), SWRST = 1(软复位)

;从ISP程序区软件复位并切换到用户应用程序区(AP区)开始执行程序
    MOV   ISP_CONTR,    #00100000B    ;SWBS = 0(选择AP区), SWRST = 1(软复位)
```

软复位后所有的寄存器复位，可在软复位前将需保存的SFR的值，保存在RAM或DATA Flash中，建议采用STC89C58RD+设计。

STC89C58RD+50-C-PI: 独立32k字节用户程序区，独立可选0/1/2/4k字节系统ISP区。
独立27k字节数据Flash区

只有选择1/2/4k字节ISP区,并选择上电复位从ISP区启动，晶片IAP功能才起作用

工作电压：5.5v - 3.4v，不要过低，否则工作不了。5V串2个1N4001是3.6V