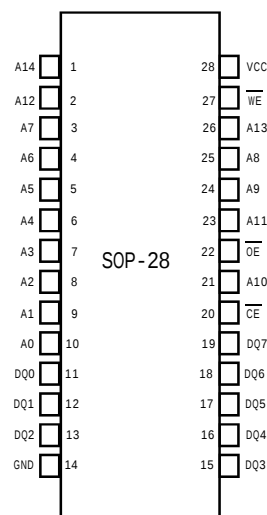


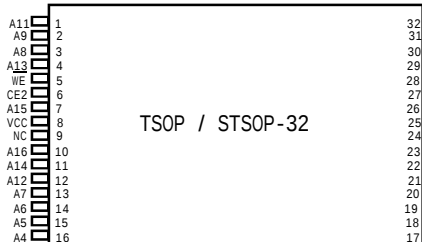
附录 H: STC 高性能 SRAM 选型一览表

型号	容量	工作电压	温度	速度	推荐封装	供货
STC62W256	32K x 8	2.4 - 5.5V	-40 ~ 85	70nS	SOP/TSOP/DIP	现货
STC62W1024	128K x 8	2.4 - 5.5V	-40 ~ 85	70nS	SOP/STSOP/TSOP	现货
STC62W2568	256K x 8	2.4 - 5.5V	-40 ~ 85	70nS	STSOP-32	现货
STC62W5128	512K x 8	2.4 - 5.5V	-40 ~ 85	70nS	STSOP/SOP-32	现货
STC62W1M8	1M x 8	2.4 - 5.5V	-40 ~ 85	70nS	TSOP2-44	订货
STC62W6416	64K x 8	2.4 - 5.5V	-40 ~ 85	70nS	TSOP2-44	现货
STC62W12816	128K x 16	2.4 - 5.5V	-40 ~ 85	70nS	TSOP2-44	现货
STC62LV12816	128K x 16	2.4 - 3.6V	-40 ~ 85	70nS	TSOP2-44	现货
STC62W25616	256K x 16	2.4 - 5.5V	-40 ~ 85	70nS	TSOP2-44	现货
STC62W51216	512K x 16	2.4 - 5.5V	-40 ~ 85	70nS	TSOP2-44	现货

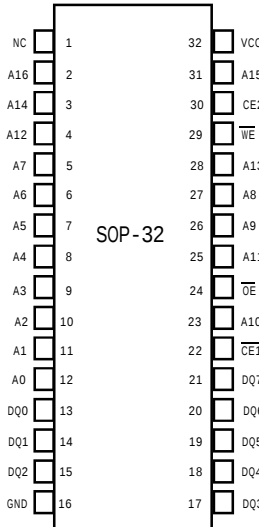
STC62W256



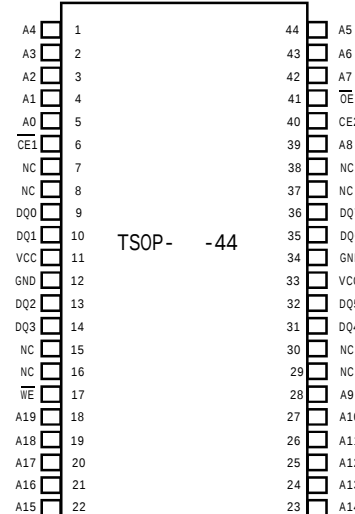
STC62WV1024



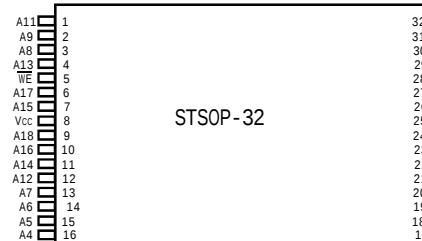
STC62WV1024



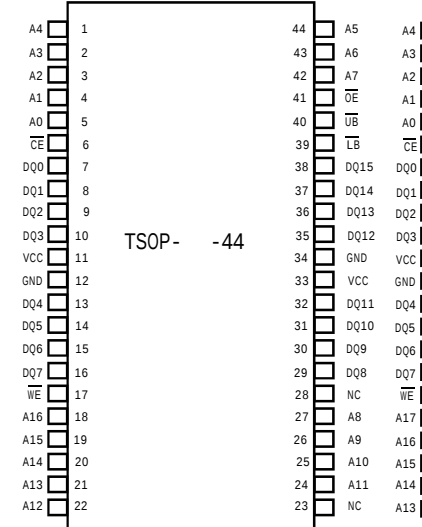
STC62WV1M8



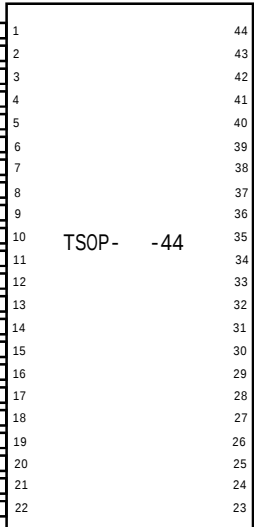
STC62WV5128



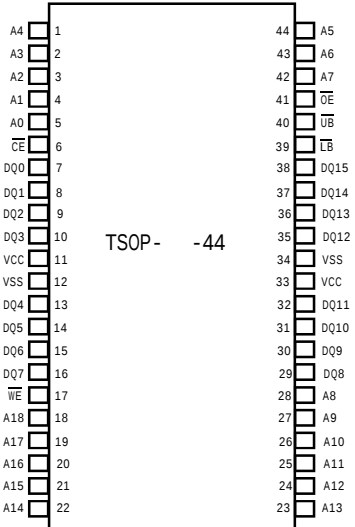
STC62WV12816



STC62WV25616



STC62WV51216



附录 I: STC 单片机配套工具价格

1. STC 单片机 ISP 下载编程工具, 人民币 50 元(可申请样品)
2. STC 单片机 ISP 到 ISP 脱机下载工具, 人民币 200 元(脱机就是量产时无须电脑, 并可限制次数)
3. STC89 系列单片机专用烧录器, 人民币 200 元(有些用户生产时希望擦除 ISP, 或要烧录自己开发的 ISP 程序, 用户自己开发的 ISP 程序可与用户应用程序一起烧录)
4. STC89 系列单片机仿真器, 人民币 1950 元

STC89C51RC/RD+ 系列 5V 单片机批量价格, 1K 以下零售加 0.3 元, 快递费 25 元

型 号	尾 缀	封 装	商 规 / 工 规	单 价	RAM 字 节	Flash 程 序	P4 口
STC89C51RC	40C-PDIP	DIP-40	商 规	4.7 元	512	4K	
STC89C51RC	40C-PLCC	PLCC-44	商 规	5.2 元	512	4K	有
STC89C51RC	40C-PQFP	PQFP-44	商 规	5.6 元	512	4K	有
STC89C51RC	40I-PDIP	DIP-40	工 规	5.5 元	512	4K	
STC89C51RC	40I-PLCC	PLCC-44	工 规	5.7 元	512	4K	有
STC89C51RC	40I-PQFP	PQFP-44	工 规	6.0 元	512	4K	有
STC89C52RC	40C-PDIP	DIP-40	商 规	5.4 元	512	8K	
STC89C52RC	40C-PLCC	PLCC-44	商 规	6.0 元	512	8K	有
STC89C52RC	40C-PQFP	PQFP-44	商 规	6.0 元	512	8K	有
STC89C52RC	40I-PDIP	DIP-40	工 规	6.7 元	512	8K	
STC89C52RC	40I-PLCC	PLCC-44	工 规	6.7 元	512	8K	有
STC89C52RC	40I-PQFP	PQFP-44	工 规	7.0 元	512	8K	有
STC89C53RC	40C-PQFP	DIP-40	商 规	8.0 元	512	15K	
STC89C53RC	40C-PLCC	PLCC-44	商 规	8.0 元	512	15K	有
STC89C53RC	40C-PQFP	PQFP-44	商 规	8.0 元	512	15K	有
STC89C53RC	40I-PQFP	DIP-40	工 规	9.0 元	512	15K	
STC89C53RC	40I-PLCC	PLCC-44	工 规	9.0 元	512	15K	有
STC89C53RC	40I-PQFP	PQFP-44	工 规	9.0 元	512	15K	有
STC89C54RD+	40C-PQFP	DIP-40	商 规	9.0 元	1280	16K	
STC89C54RD+	40C-PLCC	PLCC-44	商 规	9.0 元	1280	16K	有
STC89C54RD+	40C-PQFP	PQFP-44	商 规	9.0 元	1280	16K	有
STC89C54RD+	40I-PQFP	DIP-40	工 规	11 元	1280	16K	
STC89C54RD+	40I-PLCC	PLCC-44	工 规	11 元	1280	16K	有
STC89C54RD+	40I-PQFP	PQFP-44	工 规	11 元	1280	16K	有
STC89C58RD+	40C-PQFP	DIP-40	商 规	12 元	1280	32K	
STC89C58RD+	40C-PLCC	PLCC-44	商 规	12 元	1280	32K	有
STC89C58RD+	40C-PQFP	PQFP-44	商 规	12 元	1280	32K	有
STC89C58RD+	40I-PDIP	DIP-40	工 规	15 元	1280	32K	
STC89C58RD+	40I-PLCC	PLCC-44	工 规	15 元	1280	32K	有
STC89C58RD+	40I-PQFP	PQFP-44	工 规	15 元	1280	32K	有
STC89C516RD+	40C-PDIP	DIP-40	商 规	13.5 元	1280	63K	
STC89C516RD+	40C-PLCC	PLCC-44	商 规	13.5 元	1280	63K	有
STC89C516RD+	40C-PQFP	PQFP-44	商 规	13.5 元	1280	63K	有
STC89C516RD+	40I-PDIP	DIP-40	工 规	17 元	1280	63K	
STC89C516RD+	40I-PLCC	PLCC-44	工 规	17 元	1280	63K	有
STC89C516RD+	40I-PQFP	PQFP-44	工 规	17 元	1280	63K	有
STC89LE516AD				14 元			
STC89LE58AD				12.5 元			
STC89LE54AD				11 元			
STC89LE52AD				9 元			

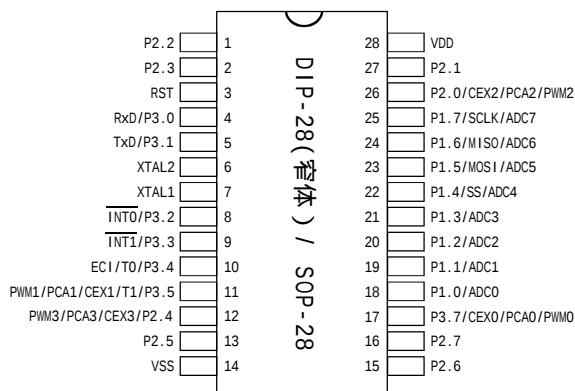
附录 J: STC12C5410AD 系列 1T 单片机简介

STC12C5410AD 系列单片机是宏晶科技推出的单时钟 / 机器周期(1T)的单片机,是高速 / 低功耗 / 超强抗干扰的新一代 8051 单片机,指令代码完全兼容传统 8051,但速度快 8-12 倍,内部集成 MAX810 专用复位电路。4 路 PWM,8 路高速 10 位 A/D 转换。

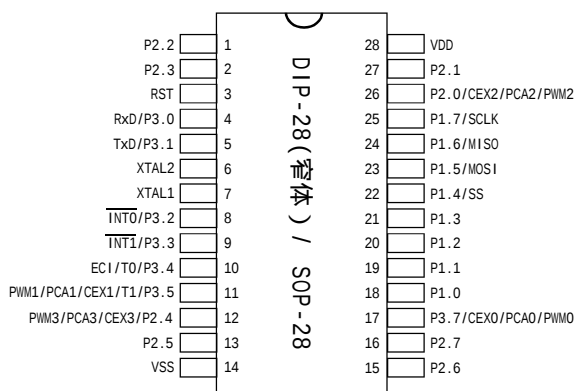
特 点 :

1. 增强型 8051 CPU, 1T, 单时钟 / 机器周期, 指令代码完全兼容传统 8051
2. 工作电压: 5.5V - 3.8V (5V 单片机) / 3.8V - 2.4V (3V 单片机)
3. 工作频率范围: 0 - 35 MHz, 相当于普通 8051 的 0 ~ 420MHz. 实际工作频率可达 48MHz.
4. 用户应用程序空间 12K / 10K / 8K / 6K / 4K / 2K / 1K 字节
5. 片上集成 512 字节 RAM
6. 通用 I/O 口 (27/23 个), 复位后为: 准双向口 / 弱上拉 (普通 8051 传统 I/O 口)
可设置成四种模式: 准双向口 / 弱上拉, 推挽 / 强上拉, 仅为输入 / 高阻, 开漏
每个 I/O 口驱动能力均可达到 20mA, 但整个芯片最大不得超过 55mA
7. ISP (在系统可编程) / IAP (在应用可编程), 无需专用编程器, 无需专用仿真器
可通过串口 (P3.0/P3.1) 直接下载用户程序, 数秒即可完成一片
8. EEPROM 功能
9. 看门狗
10. 内部集成 MAX810 专用复位电路 (外部晶体 20M 以下时, 可省外部复位电路)
11. 时钟源: 外部高精度晶体 / 时钟, 内部 R/C 振荡器
用户在下载用户程序时, 可选择是使用内部 R/C 振荡器还是外部晶体 / 时钟
常温下内部 R/C 振荡器频率为: 5.2MHz ~ 6.8MHz
精度要求不高时, 可选择使用内部时钟, 但因为有制造误差和温漂, 应认为是 4MHz ~ 8MHz
12. 共 2 个 16 位定时器 / 计数器, 但可用 PCA 模块再产生 4 个定时器
13. 外部中断 2 路, 下降沿中断或低电平触发中断, Power Down 模式可由外部中断低电平触发中断方式唤醒
14. PWM (4 路) / PCA (可编程计数器阵列)
--- 也可用来再实现 4 个定时器
--- 也可用来再实现 4 个外部中断 (上升沿中断 / 下降沿中断均可分别或同时支持)
15. ADC, 10 位精度 ADC, 共 8 路
16. 通用全双工异步串行口 (UART), 由于 STC12 系列是高速的 8051, 也可再用定时器软件实现多串口
17. SPI 同步通信口, 主模式 / 从模式
18. 工作温度范围: 0 - 75 / -40 - +85
19. 封装: PDIP-28, SOP-28, PDIP-20, SOP-20, PLCC-32, TSSOP-20 (超小封装 6.4mm × 6.4mm, 定货)

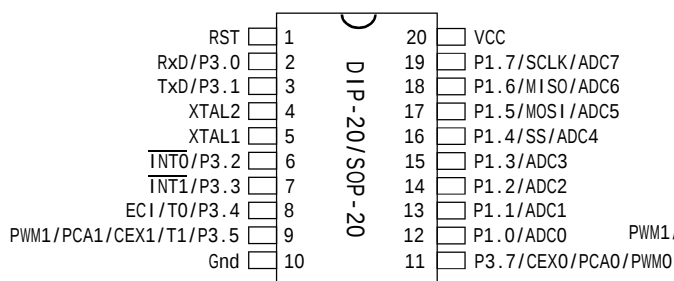
STC12C5410AD 系列 1T 单片机管脚图



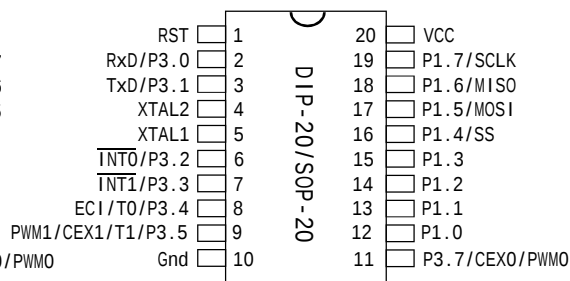
STC12C5410AD (有 A/D 转换)



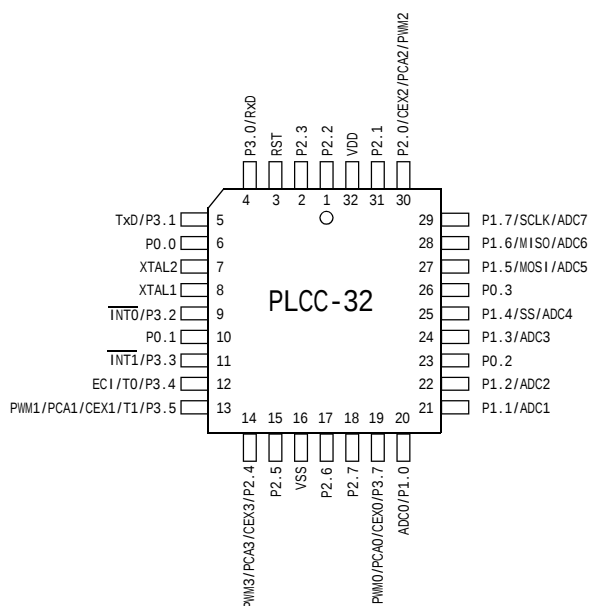
STC12C5410 (无 A/D 转换)



STC12C5410AD (有 A/D 转换)

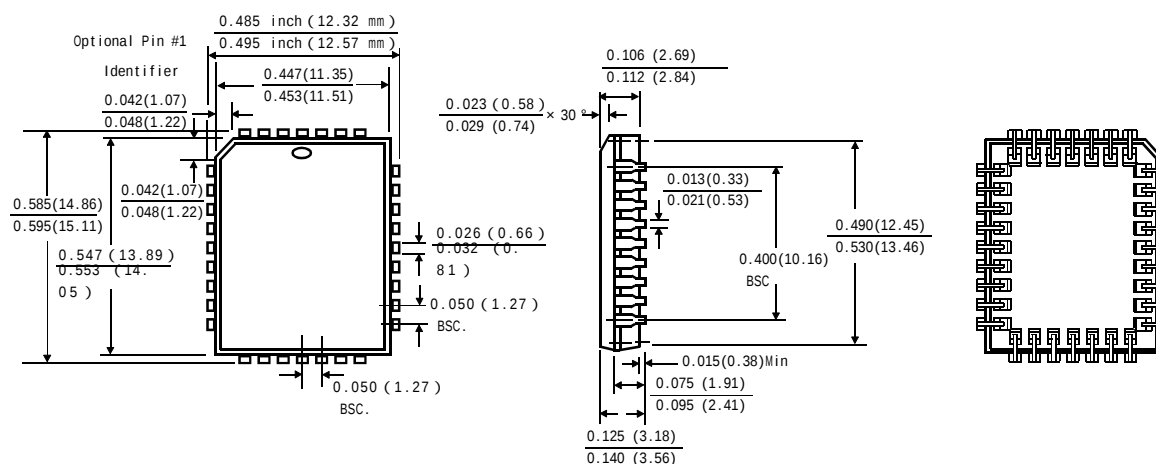


STC12C5410 (无 A/D 转换)

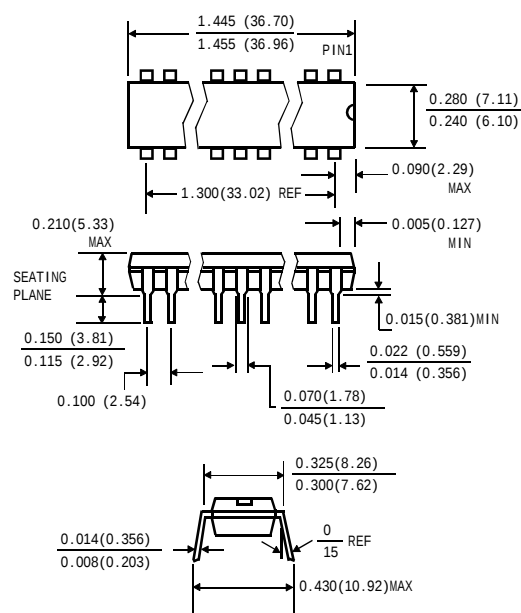


STC12C5410AD 系列 1T 单片机封装尺寸图

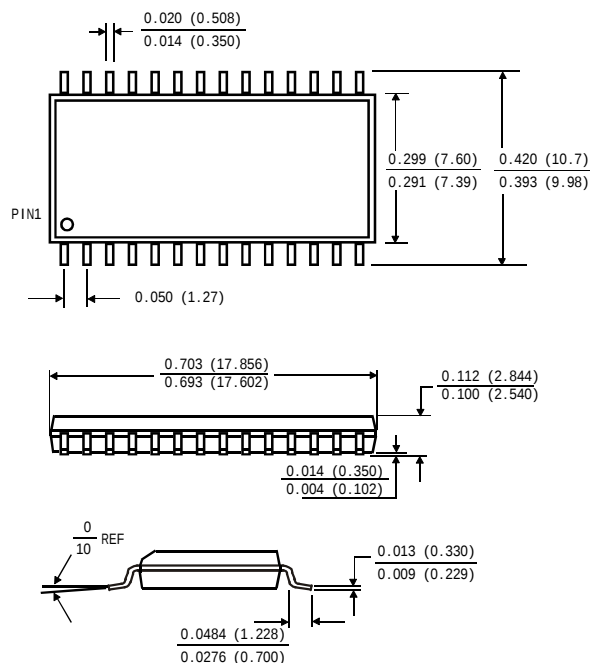
32-PIN PLASTIC LEAD CHIP CARRIER (PLCC)



28-PIN PLASTIC DUAL-IN-LINE PACKAGE (PDIP)



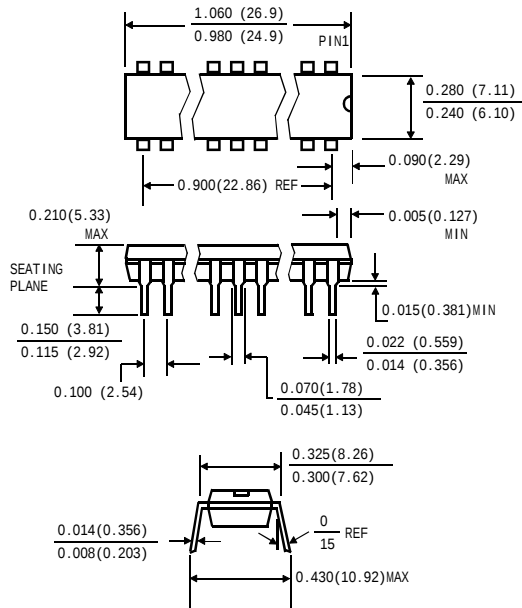
28-PIN SMALL OUTLINE PACKAGE (SOP)



20P3, 20-lead, 0.300" Wide, Plastic Dual Inline Package (PDIP-20)

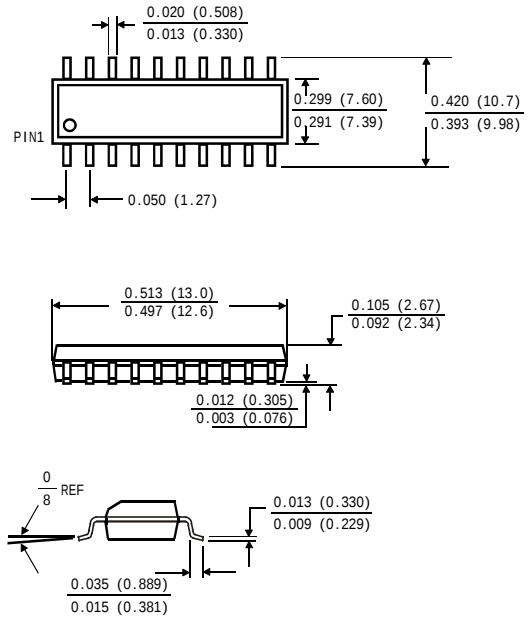
Dimensions in Inches and (Millimeters)

JEDEC STANDARD MS-001 AD



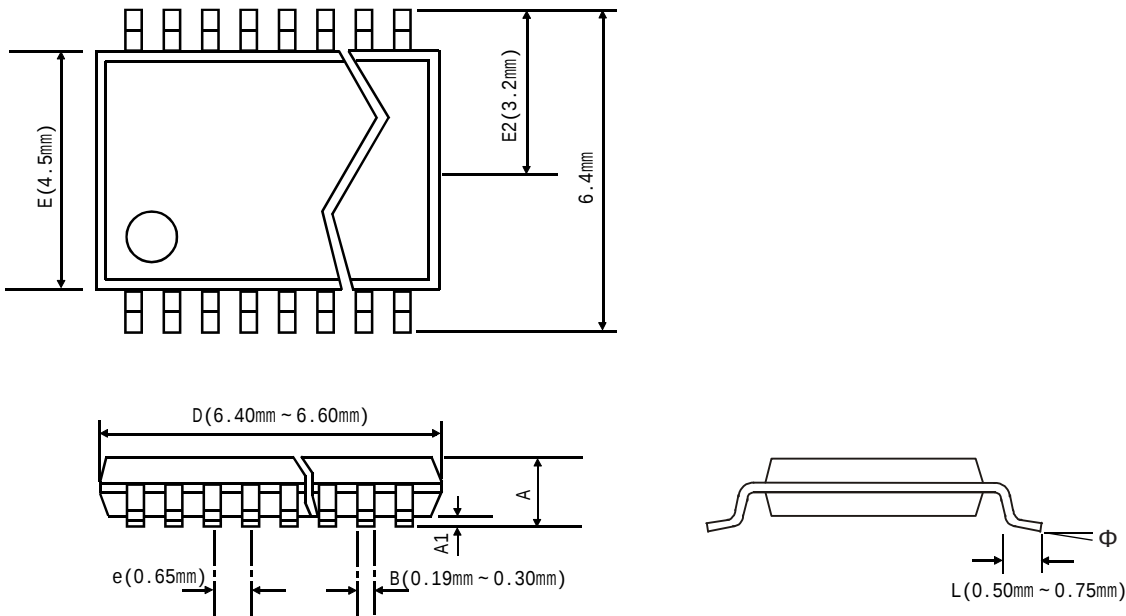
20S, 20-lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC-20 / SOP-20)

Dimensions in Inches and (Millimeters)



STC12C5410AD 系列的 TSSOP-20 封装需订货

PACKAGE : PLASTIC SHRINK SMALL OUTLINE (TSSOP-20 , 6.4mm × 6.4mm)

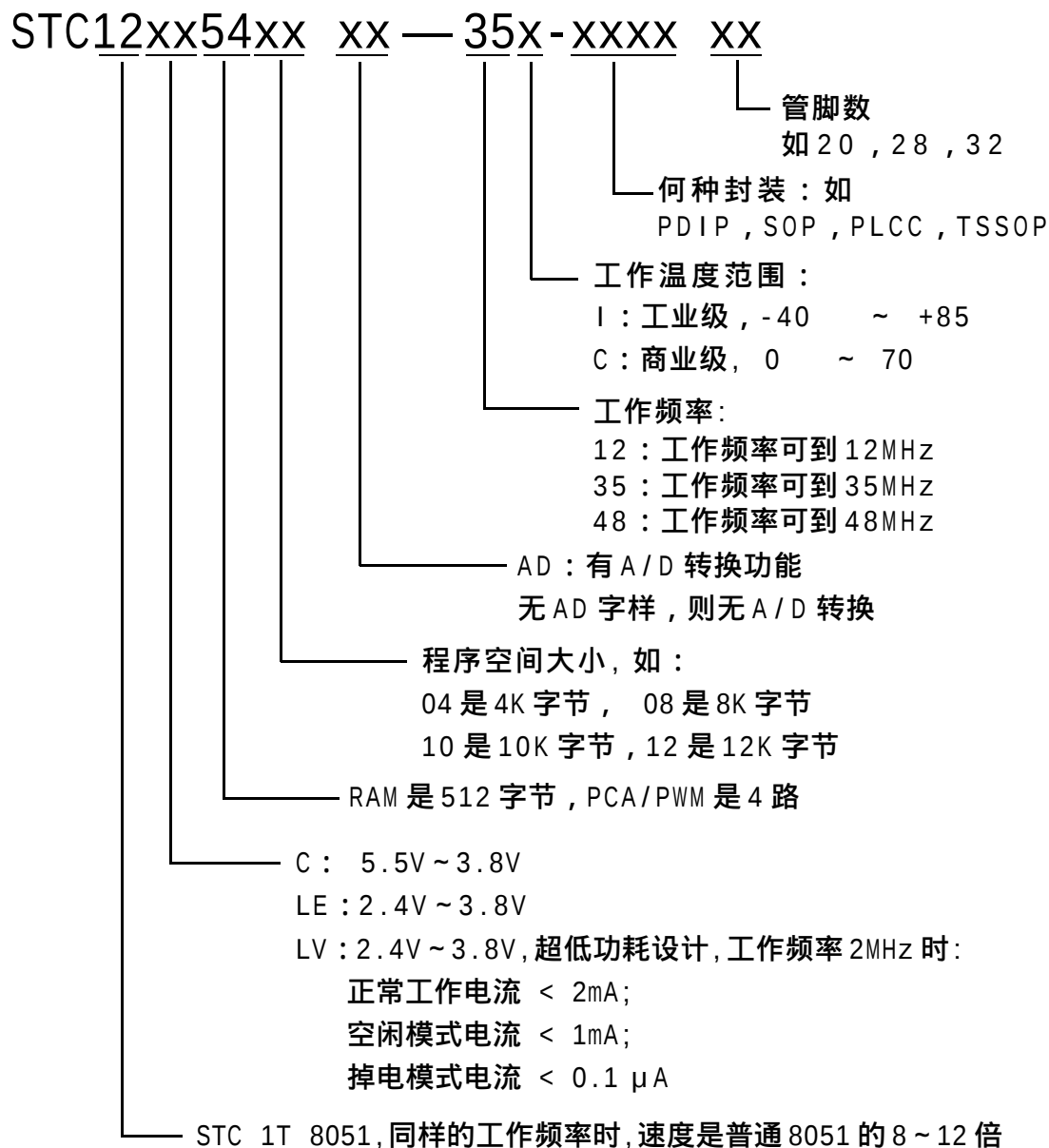


DIMENSIONS in inches (mm) Minimum/Maximum		20-PIN
A	-	0.043 (1.10)
A1	0.002/0.006 (0.05/0.15mm)	
B	0.007/0.012 (0.19/0.30mm)	
D	0.252/0.260 (6.40/6.60mm)	
E	0.169/0.177 (4.30/4.50mm)	
e	0.026 BSC (0.65mm BSC)	
E2	0.126 BSC (3.20mm BSC)	
L	0.020/0.030 (0.50/0.75mm)	
Φ	0°/8°	

STC12C5410AD 系列单片机选型一览表

	工作 电压(V)	Flash 程序 存储器 字节	SRAM 字节	定 时 器	UART	PCA PWM	A/D	I/O	看 门 狗	内 置 复 位	EEP ROM	S P I	封装 20-Pin	封装 28-Pin	封装 32-Pin
STC12C5401	5.5 - 3.8	1K	512	2	有	4路		27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12C5401AD	5.5 - 3.8	1K	512	2	有	4路	有	27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12C5402	5.5 - 3.8	2K	512	2	有	4路		27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12C5402AD	5.5 - 3.8	2K	512	2	有	4路	有	27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12C5404	5.5 - 3.8	4K	512	2	有	4路		27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12C5404AD	5.5 - 3.8	4K	512	2	有	4路	有	27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12C5406	5.5 - 3.8	6K	512	2	有	4路		27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12C5406AD	5.5 - 3.8	6K	512	2	有	4路	有	27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12C5408	5.5 - 3.8	8K	512	2	有	4路		27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12C5408AD	5.5 - 3.8	8K	512	2	有	4路	有	27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12C5410	5.5 - 3.8	10K	512	2	有	4路		27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12C5410AD	5.5 - 3.8	10K	512	2	有	4路	有	27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12C5412	5.5 - 3.8	12K	512	2	有	4路	有	27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12C5412AD	5.5 - 3.8	12K	512	2	有	4路	有	27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12LE5401	2.4 - 3.8	1K	512	2	有	4路		27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12LE5401AD	2.4 - 3.8	1K	512	2	有	4路	有	27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12LE5402	2.4 - 3.8	2K	512	2	有	4路		27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12LE5402AD	2.4 - 3.8	2K	512	2	有	4路	有	27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12LE5404	2.4 - 3.8	4K	512	2	有	4路		27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12LE5404AD	2.4 - 3.8	4K	512	2	有	4路	有	27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12LE5406	2.4 - 3.8	6K	512	2	有	4路		27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12LE5406AD	2.4 - 3.8	6K	512	2	有	4路	有	27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12LE5408	2.4 - 3.8	8K	512	2	有	4路		27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12LE5408AD	2.4 - 3.8	8K	512	2	有	4路	有	27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12LE5410	2.4 - 3.8	10K	512	2	有	4路		27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12LE5410AD	2.4 - 3.8	10K	512	2	有	4路	有	27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12LE5412	2.4 - 3.8	12K	512	2	有	4路		27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32
STC12LE5412AD	2.4 - 3.8	12K	512	2	有	4路	有	27/23	有	有	有	有	DIP/SOP/TSSOP	DIP/SOP	PLCC-32

STC12C5410AD 系列 单片机 命名规则



STC12C5410AD 系列 单片机 典型应用电路(28Pin)

指令系统分类总结及与普通 8051 指令执行时间对比

如果按功能分类，STC89/12 系列单片机指令系统可分为：

- 1. 数据传送类指令；
- 2. 算术操作类指令；
- 3. 逻辑操作类指令；
- 4. 控制转移类指令；
- 5. 布尔变量操作类指令。

按功能分类的指令系统表如下表所示。

数据传送类指令					
助记符	功能说明	字节数	12时钟/机器周期 所需时钟	1时钟/机器周期 所需时钟	效率 提升
MOV A, Rn	寄存器内容送入累加器	1	12	1	12倍
MOV A, direct	直接地址单元中的数据送入累加器	2	12	2	6倍
MOV A, @Ri	间接RAM中的数据送入累加器	1	12	2	6倍
MOV A, #data	立即送入累加器	2	12	2	6倍
MOV Rn, A	累加器内容送入寄存器	1	12	2	6倍
MOV Rn, direct	直接地址单元中的数据送入寄存器	2	24	4	6倍
MOV Rn, #data	立即数送入寄存器	2	12	2	6倍
MOV direct, A	累加器内容送入直接地址单元	2	12	3	4倍
MOV direct, Rn	寄存器内容送入直接地址单元	2	24	3	8倍
MOV direct, direct	直接地址单元中的数据送入另一个直接地址单元	3	24	4	6倍
MOV direct, @Ri	间接RAM中的数据送入直接地址单元	2	24	4	6倍
MOV direct, #data	立即数送入直接地址单元	3	24	3	8倍
MOV @Ri, A	累加器内容送入间接RAM单元	1	12	3	4倍
MOV @Ri, direct	直接地址单元数据送入间接RAM单元	2	24	3	8倍
MOV @Ri, #data	立即数送入间接RAM单元	2	12	3	4倍
MOV DPTR, #data16	16位立即数送入地址寄存器	3	24	3	8倍
MOVC A, @A+DPTR	以DPTR为基地址变址寻址单元中的数据送入累加器	1	24	4	6倍
MOVC A, @A+PC	以PC为基地址变址寻址单元中的数据送入累加器	1	24	4	6倍
MOVB A, @Ri	外部RAM（8位地址）送入累加器	1	24	3	8倍
MOVB A, @DPTR	外部RAM（16位地址）送入累加器	1	24	3	8倍
MOVB @Ri, A	累加器送外部RAM（8位地址）	1	24	3	8倍
MOVB @DPTR, A	累加器送外部RAM（16位地址）	1	24	3	8倍
PUSH direct	直接地址单元中的数据压入堆栈	2	24	4	6倍
POP direct	出栈送直接地址单元	2	24	3	8倍
XCH A, Rn	寄存器与累加器交换	1	12	3	4倍
XCH A, direct	直接地址单元与累加器交换	2	12	4	3倍
XCH A, @Ri	间接RAM与累加器交换	1	12	4	3倍
XCHD A, @Ri	间接RAM的低半字节与累加器交换	1	12	4	3倍

传统 12T 的 8051

STC12C5410AD 系列

算术操作类指令

助记符	功能说明	字节数	12时钟/周期 所需时钟	1时钟/周期 所需时钟	提升 效率
ADD A, Rn	寄存器内容加到累加器	1	12	2	6倍
ADD A, direct	直接地址单元中的数据加到累加器	2	12	3	4倍
ADD A, @Ri	间接RAM中的数据加到累加器	1	12	3	4倍
ADD A, #data	立即加到累加器	2	12	2	6倍
ADDC A, Rn	寄存器内容带进位加到累加器	1	12	2	6倍
ADDC A, direct	直接地址单元的内容带进位加到累加器	2	12	3	4倍
ADDC A, @Ri	间接RAM内容带进位加到累加器	1	12	3	4倍
ADDC A, #data	立即数带进位加到累加器	2	12	2	6倍
SUBB A, Rn	累加器带借位减寄存器内容	1	12	2	6倍
SUBB A, direct	累加器带借位减直接地址单元的内容	2	12	3	4倍
SUBB A, @Ri	累加器带借位减间接RAM中的内容	1	12	3	4倍
SUBB A, #data	累加器带借位减立即数	2	12	2	6倍
INC A	累加器加1	1	12	2	6倍
INC Rn	寄存器加1	1	12	3	4倍
INC direct	直接地址单元加1	2	12	4	3倍
INC @Ri	间接RAM单元加1	1	12	4	3倍
DEC A	累加器减1	1	12	2	6倍
DEC Rn	寄存器减1	1	12	3	4倍
DEC direct	直接地址单元减1	2	12	4	3倍
DEC @Ri	间接RAM单元减1	1	12	4	3倍
INC DPTR	地址寄存器DPTR加1	1	24	1	24倍
MUL AB	A乘以B	1	48	4	12倍
DIV AB	A除以B	1	48	5	9.6倍
DA A	累加器十进制调整	1	12	4	3倍

逻辑操作类指令

助记符	功能说明	字节数	12时钟/周期 所需时钟	1时钟/周期 所需时钟	提升 效率
ANL A, Rn	累加器与寄存器相“与”	1	12	2	6倍
ANL A, direct	累加器与直接地址单元相“与”	2	12	3	4倍
ANL A, @Ri	累加器与间接RAM单元相“与”	1	12	3	4倍
ANL A, #data	累加器与立即数相“与”	2	12	2	6倍
ANL direct, A	直接地址单元与累加器相“与”	2	12	4	3倍
ANL direct, #data	直接地址单元与立即数相“与”	3	24	4	6倍
ORL A, Rn	累加器与寄存器相“或”	1	12	2	6倍
ORL A, direct	累加器与直接地址单元相“或”	2	12	3	4倍
ORL A, @Ri	累加器与间接RAM单元相“或”	1	12	3	4倍
ORL A, #data	累加器与立即数相“或”	2	12	2	6倍
ORL direct, A	直接地址单元与累加器相“或”	2	12	4	3倍
ORL direct, #data	直接地址单元与立即数相“或”	3	24	4	6倍
XRL A, Rn	累加器与寄存器相“异或”	1	12	2	6倍
XRL A, direct	累加器与直接地址单元相“异或”	2	12	3	4倍
XRL A, @Ri	累加器与间接RAM单元相“异或”	1	12	3	4倍
XRL A, #data	累加器与立即数相“异或”	2	12	2	6倍
XRL direct, A	直接地址单元与累加器相“异或”	2	12	4	3倍
XRL direct, #data	直接地址单元与立即数相“异或”	3	24	4	6倍
CLR A	累加器清“0”	1	12	1	12倍
CPL A	累加器求反	1	12	2	6倍
RL A	累加器循环左移	1	12	1	12倍
RLC A	累加器带进位位循环左移	1	12	1	12倍
RR A	累加器循环右移	1	12	1	12倍
RRC A	累加器带进位位循环右移	1	12	1	12倍
SWAP A	累加器半字节交换	1	12	1	12倍

控制转移类指令

助记符	功能说明	字节数	12时钟/周期 所需时钟	1时钟/周期 所需时钟	提升 效率
ACALL addr11	绝对(短)调用子程序	2	24	6	4倍
LCALL addr16	长调用子程序	3	24	6	4倍
RET	子程序返回	1	24	4	6倍
RETI	中断返回	1	24	4	6倍
AJMP addr11	绝对(短)转移	2	24	3	8倍
LJMP addr16	长转移	3	24	4	6倍
SJMP re1	相对转移	2	24	3	8倍
JMP @A+DPTR	相对于DPTR的间接转移	1	24	3	8倍
JZ re1	累加器为零转移	2	24	3	8倍
JNZ re1	累加器非零转移	2	24	3	8倍
CJNE A, direct, re1	累加器与直接地址单元比较, 不相等则转移	3	24	5	4.8倍
CJNE A, #data, re1	累加器与立即数比较, 不相等则转移	3	24	4	6倍
CJNE Rn, #data, re1	寄存器与立即数比较, 不相等则转移	3	24	4	6倍
CJNE @Ri, #data, re1	间接RAM单元与立即数比较, 不相等则转移	3	24	5	4.8倍
DJNZ Rn, re1	寄存器减1, 非零转移	3	24	4	6倍
DJNZ direct, re1	直接地址单元减1, 非零转移	3	24	5	4.8倍
NOP	空操作	1	12	1	12倍

布尔变量操作类指令

助记符	功能说明	字节数	12时钟/周期 所需时钟	1时钟/周期 所需时钟	提升 效率
CLR C	清0进位位	1	12	1	12倍
CLR bit	清0直接地址位	2	12	4	3倍
SETB C	置1进位位	1	12	1	12倍
SETB bit	置1直接地址位	2	12	4	3倍
CPL C	进位位求反	1	12	1	12倍
CPL bit	直接地址位求反	2	12	4	3倍
ANL C, bit	进位位和直接地址位相“与”	2	24	3	8倍
ANL C, $\overline{\text{bit}}$	进位位和直接地址位的反码相“与”	2	24	3	8倍
ORL C, bit	进位位和直接地址位相“或”	2	24	3	8倍
ORL C, $\overline{\text{bit}}$	进位位和直接地址位的反码相“或”	2	24	3	8倍
MOV C, bit	直接地址位送入进位位	2	12	3	4倍
MOV bit, C	进位位送入直接地址位	2	24	3	8倍
JC re1	进位位为1则转移	2	24	3	8倍
JNC re1	进位位为0则转移	2	24	3	8倍
JB bit, re1	直接地址位为1则转移	3	24	4	6倍
JNB bit, re1	直接地址位为0则转移	3	24	4	6倍
JBC bit, re1	直接地址位为1则转移, 该位清0	3	24	5	4.8倍

指令执行速度效率提升总结：

指令系统共包括 111 条指令，其中：

执行速度快 24 倍的	共 1 条
执行速度快 12 倍的	共 12 条
执行速度快 9.6 倍的	共 1 条
执行速度快 8 倍的	共 21 条
执行速度快 6 倍的	共 37 条
执行速度快 4.8 倍的	共 4 条
执行速度快 4 倍的	共 21 条
执行速度快 3 倍的	共 14 条

根据对指令的使用频率分析统计，STC12 系列 1T 的 8051 单片机比普通的 8051 单片机在同样的工作频率下运行速度提升了 8 ~ 12 倍。

指令执行时钟数统计（供参考）：

指令系统共包括 111 条指令，其中：

1 个时钟就可执行完成的指令	共 12 条
2 个时钟就可执行完成的指令	共 20 条
3 个时钟就可执行完成的指令	共 40 条
4 个时钟就可执行完成的指令	共 32 条
5 个时钟就可执行完成的指令	共 5 条
6 个时钟就可执行完成的指令	共 2 条

特殊功能寄存器映像 SFR Mapping

	Bit Addressable	Non Bit Addressable							
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8h		CH 0000,0000	CCAP0H 0000,0000	CCAP1H 0000,0000	CCAP2H 0000,0000	CCAP3H 0000,0000			FFh
F0h	B 0000,0000		PCA_PWM0 xxxx,xx00	PCA_PWM1 xxxx,xx00	PCA_PWM2 xxxx,xx00	PCA_PWM3 xxxx,xx00			F7h
E8h		CL 0000,0000	CCAP0L 0000,0000	CCAP1L 0000,0000	CCAP2L 0000,0000	CCAP3L 0000,0000			EFh
E0h	ACC 0000,0000	WDT_CONTR 0x00,0000	ISP_DATA 1111,1111	ISP_ADDRH 0000,0000	ISP_ADDRL 0000,0000	ISP_CMD xxxx,xx00	ISP_TRIG xxxx,xxxx	ISP_CONTR 0000,1000	E7h
D8h	CCON 00xx,0000	CMOD 0xxx,x000	CCAPM0 x000,0000	CCAPM1 x000,0000	CCAPM2 x000,0000	CCAPM3 x000,0000			DFh
D0h	PSW 0000,0000								D7h
C8h									CFh
C0h						ADC_CONTR 0000,0000	ADC_DATA 0000,0000	CLK_DIV xxxx,x000	C7h
B8h	IP x000,0000	SADEN don't use					ADC_LOW2 0000,0000		BFh
B0h	P3 1x11,1111	P3M0 0000,0000	P3M1 0000,0000					IPH x000,0000	B7h
A8h	IE 0000,0000	SADDR don't use							AFh
A0h	P2 1111,1111							TEST_WDT don't use	A7h
98h	SCON 0000,0000	SBUF xxxx,xxxx							9Fh
90h	P1 1111,1111	P1M0 0000,0000	P1M1 0000,0000	P0M0 0000,0000	P0M1 0000,0000	P2M0 0000,0000	P2M1 0000,0000		97h
88h	TCON 0000,0000	TMOD 0000,0000	TL0 0000,0000	TL1 0000,0000	TH0 0000,0000	TH1 0000,0000	AUXR 0000,00xx		8Fh
80h	P0 xxxx,1111	SP 0000,0111	DPL 0000,0000	DPH 0000,0000	SPSTAT 00xx,xxxx	SPCTL 0000,0100	SPDAT 0000,0000	PCON 0011,0000	87h
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

特别标出部分为在 Intel 8052 基础上新增加的特殊功能寄存器，一般用户可不管

STC12C5410AD 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	F1	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

STC12C5410AD 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	SMOD0	LVDF	POF	GF1	GF0	PD	IDL	0011,0000
AUXR	8Eh	Auxiliary Register	T0x12	T1x12	UART_M0x6	EADCI	ESPI	ELVDI	-	-	0000,00xx
CLK_DIV	C7h	Clock Divder	-	-	-	-	-	CLKS2	CLKS1	CLKS0	xxxx,x000

STC12C5410AD 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	-	-	-	-	P0.3	P0.2	P0.1	P0.0	xxxx,1111
P0M0	93h										0000,0000
P0M1	94h										0000,0000
P1	90h	8-bit Port 1	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	1111,1111
P1M0	91h										0000,0000
P1M1	92h										0000,0000
P2	A0h	8-bit Port 2	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	1111,1111
P2M0	95h										0000,0000
P2M1	96h										0000,0000
P3	B0h	8-bit Port 3	P3.7	-	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	1x11,1111
P3M0	B1h										0000,0000
P3M1	B2h										0000,0000

STC12C5410AD 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
AUXR	8Eh	Auxiliary Register	T0x12	T1x12	UART_M0x6	EADCI	ESPI	ELVDI	-	-	0000,00xx

STC12C5410AD 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
AUXR	8Eh	Auxiliary Register	T0x12	T1x12	UART_M0x6	EADCI	ESPI	ELVDI	-	-	0000,00xx

STC12C5410AD 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	WDT_FLAG	-	EN_WDT	CLR_WDT	IDLE_WDT	PS2	PS1	PS0	xx00,0000

STC12C5410AD 系列 1T 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	EPCA_LVD	EADC_SPI	ES	ET1	EX1	ET0	EX0	0000,0000
IP	B8h	Interrupt Priority Low	-	PPCA_LVD	PADC_SPI	PS	PT1	PX1	PT0	PX0	xx00,0000
IPH	B7h	Interrupt Priority High	-	PPCA_LVDH	PADC_SPIH	PSH	PT1H	PX1H	PT0H	PX0H	0000,0000
TCON	88h	Timer / Counter 0 and 1 Control	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000,0000
SCON	98h	Serial Control	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	0000,0000
PCON	87h	Power Control	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL	0011,0000
AUXR	8Eh	Auxiliary Register	T0x12	T1x12	UART_M0x6	EADC1	ESPI	ELVD1	-	-	0000,00xx
ADC_CONTR	C5h	A/D 转换控制寄存器	ADC_POWER	SPEED1	SPEED0	ADC_FLAG	ADC_START	CHS2	CHS1	CHS0	0xx0,0000
SPSTAT	84h	SPI Status Register	SPIF	WCOL	-	-	-	-	-	-	00xx,xxxx
CCON	D8h	PCA Control Register	CF	CR	-	-	CCF3	CCF2	CCF1	CCF0	00xx,0000
CMOD	D9h	PCA Mode Register	CIDL	-	-	-	-	CPS1	CPS0	ECF	0xxx,x000
CCAPM0	DAh	PCA Module 0 Mode Register	-	ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECCF0	x000,0000
CCAPM1	DBh	PCA Module 1 Mode Register	-	ECOM1	CAPP1	CAPN1	MAT1	TOG1	PWM1	ECCF1	x000,0000
CCAPM2	DCh	PCA Module 2 Mode Register	-	ECOM2	CAPP2	CAPN2	MAT2	TOG2	PWM2	ECCF2	x000,0000
CCAPM3	DDh	PCA Module 3 Mode Register	-	ECOM3	CAPP3	CAPN3	MAT3	TOG3	PWM3	ECCF3	x000,0000

STC12C5410AD 系列 8051 单片机 PCA/PWM 特殊功能寄存器 PCA/PWM SFRs

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset value
CCON	D8h	PCA Control Register	CF	CR	-	-	CCF3	CCF2	CCF1	CCF0	00xx,0000
CMOD	D9h	PCA Mode Register	CIDL	-	-	-	-	CPS1	CPS0	ECF	0xxx,x000
CCAPM0	DAh	PCA Module 0 Mode Register	-	ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECCF0	x000,0000
CCAPM1	DBh	PCA Module 1 Mode Register	-	ECOM1	CAPP1	CAPN1	MAT1	TOG1	PWM1	ECCF1	x000,0000
CCAPM2	DCh	PCA Module 2 Mode Register	-	ECOM2	CAPP2	CAPN2	MAT2	TOG2	PWM2	ECCF2	x000,0000
CCAPM3	DDh	PCA Module 3 Mode Register	-	ECOM3	CAPP3	CAPN3	MAT3	TOG3	PWM3	ECCF3	x000,0000
CL	E9h	PCA Base Timer Low									0000,0000
CH	F9h	PCA Base Timer High									0000,0000
CCAP0L	EAh	PCA Module-0 Capture Register Low									0000,0000
CCAP0H	FAh	PCA Module-0 Capture Register High									0000,0000
CCAP1L	EBh	PCA Module-1 Capture Register Low									0000,0000
CCAP1H	FBh	PCA Module-1 Capture Register High									0000,0000
CCAP2L	ECh	PCA Module-2 Capture Register Low									0000,0000
CCAP2H	FCh	PCA Module-2 Capture Register High									0000,0000
CCAP3L	EDh	PCA Module-3 Capture Register Low									0000,0000
CCAP3H	FDh	PCA Module-3 Capture Register High									0000,0000
PCA_PWM0	F2h	PCA PWM Mode Auxiliary Register 0	-	-	-	-	-	-	EPC0H	EPC0L	xxxx,xx00
PCA_PWM1	F3h	PCA PWM Mode Auxiliary Register 1	-	-	-	-	-	-	EPC1H	EPC1L	xxxx,xx00
PCA_PWM2	F4h	PCA PWM Mode Auxiliary Register 2	-	-	-	-	-	-	EPC2H	EPC2L	xxxx,xx00
PCA_PWM3	F5h	PCA PWM Mode Auxiliary Register 3	-	-	-	-	-	-	EPC3H	EPC3L	xxxx,xx00

STC12C5410AD 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	-	-	-	-	-	-	MS1	MS0	xxxx,x000
ISP_TRIG	E6h	ISP/IAP Flash Command Trigger									xxxx,xxxx
ISP_CONTR	E7h	ISP/IAP Control Register	ISPEN	SWBS	SWRST	CMD_FAIL	1	WT2	WT1	WT0	0000,1000

中断

STC12C5410AD 系列 1T 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	EPCA_LVD	EADC_SPI	ES	ET1	EX1	ET0	EX0	0000,0000
IP	B8h	Interrupt Priority Low	-	PPCA_LVD	PADC_SPI	PS	PT1	PX1	PT0	PX0	xx00,0000
IPH	B7h	Interrupt Priority High	-	PPCA_LVDH	PADC_SPIH	PSH	PT1H	PX1H	PT0H	PX0H	0000,0000
TCON	88h	Timer / Counter 0 and 1 Control	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0000,0000
SCON	98h	Serial Control	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	0000,0000
PCON	87h	Power Control	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL	0011,0000
AUXR	8Eh	Auxiliary Register	T0x12	T1x12	UART_M0x6	EADC1	ESPI	ELVD1	-	-	0000,00xx
ADC_CONTR	C5h	A/D 转换控制寄存器	ADC_POWER	SPEED1	SPEED0	ADC_FLAG	ADC_START	CHS2	CHS1	CHS0	0xx0,0000
SPSTAT	84h	SPI Status Register	SPIF	WCOL	-	-	-	-	-	-	00xx,xxxx
CCON	D8h	PCA Control Register	CF	CR	-	-	CCF3	CCF2	CCF1	CCF0	00xx,0000
CMOD	D9h	PCA Mode Register	CIDL	-	-	-	-	CPS1	CPS0	ECF	0xxx,x000
CCAPM0	DAh	PCA Module 0 Mode Register	-	ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECCF0	x000,0000
CCAPM1	DBh	PCA Module 1 Mode Register	-	ECOM1	CAPP1	CAPN1	MAT1	TOG1	PWM1	ECCF1	x000,0000
CCAPM2	DCh	PCA Module 2 Mode Register	-	ECOM2	CAPP2	CAPN2	MAT2	TOG2	PWM2	ECCF2	x000,0000
CCAPM3	DDh	PCA Module 3 Mode Register	-	ECOM3	CAPP3	CAPN3	MAT3	TOG3	PWM3	ECCF3	x000,0000

STC12C5410AD 系列 中断与普通 8051 完全兼容，优先级可设为 4 级。

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	TF1
UART	0023H	4	PSH, PS	0,0	0,1	1,0	1,1	RI + TI
ADC/SPI	002BH	5	PADC_SPIH, PADC_SPI	0,0	0,1	1,0	1,1	ADC_FLAG + SPIF
PCA/LVD	0033H	6	PPCA_LVDH, PPCA_LVD	0,0	0,1	1,0	1,1	CF + CCF0 + CCF1 + CCF2 + CCF3 + LVDF

PCA/PWM 特殊功能寄存器，其中部分位与 PCA 中断有关

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset value
CCON	D8h	PCA Control Register	CF	CR	-	-	CCF3	CCF2	CCF1	CCF0	00xx,0000
CMOD	D9h	PCA Mode Register	CIDL	-	-	-	-	CPS1	CPS0	ECF	0xxx,x000
CCAPM0	DAh	PCA Module 0 Mode Register	-	ECOM0	CAPP0	CAPN0	MAT0	TOG0	PWM0	ECCF0	x000,0000
CCAPM1	DBh	PCA Module 1 Mode Register	-	ECOM1	CAPP1	CAPN1	MAT1	TOG1	PWM1	ECCF1	x000,0000
CCAPM2	DCh	PCA Module 2 Mode Register	-	ECOM2	CAPP2	CAPN2	MAT2	TOG2	PWM2	ECCF2	x000,0000
CCAPM3	DDh	PCA Module 3 Mode Register	-	ECOM3	CAPP3	CAPN3	MAT3	TOG3	PWM3	ECCF3	x000,0000
CL	E9h	PCA Base Timer Low									0000,0000
CH	F9h	PCA Base Timer High									0000,0000
CCAP0L	EAh	PCA Module-0 Capture Register Low									0000,0000
CCAP0H	FAh	PCA Module-0 Capture Register High									0000,0000
CCAP1L	EBh	PCA Module-1 Capture Register Low									0000,0000
CCAP1H	FBh	PCA Module-1 Capture Register High									0000,0000
CCAP2L	ECh	PCA Module-2 Capture Register Low									0000,0000
CCAP2H	FCh	PCA Module-2 Capture Register High									0000,0000
CCAP3L	EDh	PCA Module-3 Capture Register Low									0000,0000
CCAP3H	FDh	PCA Module-3 Capture Register High									0000,0000
PCA_PWM0	F2h	PCA PWM Mode Auxiliary Register 0	-	-	-	-	-	-	EPC0H	EPC0L	xxxx,xx00
PCA_PWM1	F3h	PCA PWM Mode Auxiliary Register 1	-	-	-	-	-	-	EPC1H	EPC1L	xxxx,xx00
PCA_PWM2	F4h	PCA PWM Mode Auxiliary Register 2	-	-	-	-	-	-	EPC2H	EPC2L	xxxx,xx00
PCA_PWM3	F5h	PCA PWM Mode Auxiliary Register 3	-	-	-	-	-	-	EPC3H	EPC3L	xxxx,xx00

STC12C5410AD 系列 8051 单片机 SPI 功能模块特殊功能寄存器 其中 SPIF 位与中断有关

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset value
SPCTL	85h	SPI Control Register	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	0000,0000
SPSTAT	84h	SPI Status Register	SPIF	WCOL	-	-	-	-	-	-	00xx,xxxx
SPDAT	86h	SPI Data Register									0000,0000

定时器 0/ 定时器 1 , UART 串口的速度

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
AUXR	8Eh	Auxiliary Register	T0x12	T1x12	UART_M0x6	EADCI	ESPI	ELVDI	-	-	0000,00xx

定时器 0 和定时器 1:

STC12C5410AD 系列是 1T 的 8051 单片机, 为了兼容传统 8051, 定时器 0 和定时器 1 复位后是传统 8051 的速度, 即 12 分频, 这是为了兼容传统 8051。但也可不进行 12 分频, 实现真正的 1T。

T0x12: 0, 定时器 0 是传统 8051 速度, 12 分频; 1, 定时器 0 的速度是传统 8051 的 12 倍, 不分频

T1x12: 0, 定时器 1 是传统 8051 速度, 12 分频; 1, 定时器 1 的速度是传统 8051 的 12 倍, 不分频

UART 串口的模式 0:

STC12C2052AD 系列是 1T 的 8051 单片机, 为了兼容传统 8051, UART 串口复位后是兼容传统 8051 的。

UART_M0x6: 0, UART 串口的模式 0 是传统 12T 的 8051 速度, 12 分频;

 1, UART 串口的模式 0 的速度是传统 12T 的 8051 的 6 倍, 2 分频

EADCI: 0, 禁止 A/D 中断; 1, 允许 A/D 中断

ESPI: 0, 禁止 SPI 中断; 1, 允许 SPI 中断

ELVDI: 0, 禁止低压中断; 1, 允许低压中断。

5V 单片机, 3.7V 以下为低压, 3V 单片机, 2.4V 以下为低压,
如 ELVDI=1 (允许低压中断), 则会产生低压中断

系统工作时钟

STC12C5410AD 系列是 1T 的 8051 单片机，系统时钟兼容传统 8051。

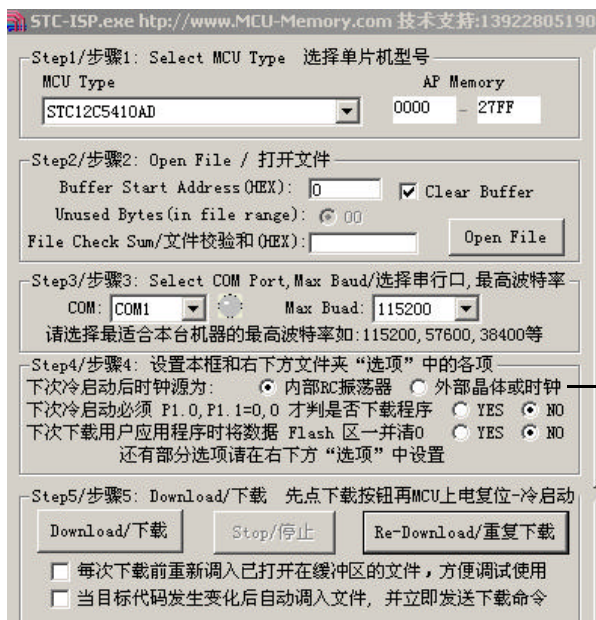
现出厂标准配置是使用芯片内部的 R/C 振荡器，5V 单片机常温下频率是 5.65MHz - 5.95MHz，因为随着温度的变化，内部 R/C 振荡器的频率会有一些温飘，故内部 R/C 振荡器只适用于对时钟频率要求不敏感的场所。

在对 STC12C5410AD 系列单片机进行 ISP 下载用户程序时，可以在选项中选择：

“下次冷启动后时钟源为外部晶体或时钟”

这样下载完用户程序后，停电，再冷启动后单片机的工作时钟使用的就不是内部 R/C 振荡器，而是外部晶体振荡后产生的高精度时钟了（接在 XTAL1/XTAL2 管脚上），也可以直接从 XTAL1 脚输入外部时钟，XTAL2 脚浮空。

如果还要设置成使用内部 R/C 振荡器，在对 STC12C5410AD 系列单片机进行 ISP 下载用户程序时，可以在选项中选择：“下次冷启动后时钟源为内部 R/C 振荡器”



系统时钟分频寄存器

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset value
CLK_DIV	C7h	Clock Divider	-	-	-	-	-	CLKS2	CLKS1	CLKS0	xxxx, x000

如用户系统希望大幅降低功耗，还可对系统时钟进行分频。

CLKS2	CLKS1	CLKS0	CPU的实际工作时钟
0	0	0	系统时钟(外部时钟或内部R/C振荡时钟)
0	0	1	系统时钟/2
0	1	0	系统时钟/4
0	1	1	系统时钟/8
1	0	0	系统时钟/16
1	0	1	系统时钟/32
1	1	0	系统时钟/64
1	1	1	系统时钟/128

I/O 口结构

I/O 口配置

STC12C5410AD 系列单片机其所有 I/O 口均可由软件配置成 4 种工作类型之一，如下表所示。4 种类型分别为：准双向口（标准 8051 输出模式）、推挽输出、仅为输入（高阻）或开漏输出功能。每个口由 2 个控制寄存器中的相应位控制每个引脚工作类型。STC12C5410AD 系列单片机上电复位后为准双向口（标准 8051 输出模式）模式。2V 以上时为高电平，0.8V 以下时为低电平。

I/O 口工作类型设定

P3 口设定 <P3.7,P3.6,P3.5,P3.4,P3.3,P3.2,P3.1,P3.0>

P3M0【7：0】	P3M1【7：0】	I/O 口模式
0	0	准双向口(传统 8051 I/O 口模式)，灌电流可达 20mA，拉电流为 230μA，由于制造误差，实际为 250uA ~ 160uA
0	1	推挽输出（强上拉输出，可达 20mA，尽量少用）
1	0	仅为输入（高阻）
1	1	开漏(Open Drain)，内部上拉电阻断开，要外加

P2 口设定 <P2.7,P2.6,P2.5,P2.4,P2.3,P2.2,P2.1,P2.0>

P2M0【7：0】	P2M1【7：0】	I/O 口模式
0	0	准双向口(传统 8051 I/O 口模式)，灌电流可达 20mA，拉电流为 230μA，由于制造误差，实际为 250uA ~ 160uA
0	1	推挽输出（强上拉输出，可达 20mA，尽量少用）
1	0	仅为输入（高阻）
1	1	开漏(Open Drain)，内部上拉电阻断开，要外加

P1 口设定 <P1.7,P1.6,P1.5,P1.4,P1.3,P1.2,P1.1,P1.0>

P1M0【7：0】	P1M1【7：0】	I/O 口模式（P1.x 如做 A/D 使用，需先将其设置成开漏或高阻输入）
0	0	准双向口（传统 8051 I/O 口模式），灌电流可达 20mA，拉电流为 230μA，由于制造误差，实际为 250uA ~ 160uA
0	1	推挽输出（强上拉输出，可达 20mA，尽量少用）
1	0	仅为输入（高阻），如果该 I/O 口需作为 A/D 使用，可选此模式
1	1	开漏(Open Drain)，如果该 I/O 口需作为 A/D 使用，可选此模式

P0 口设定 <P0.7,P0.6,P0.5,P0.4,P0.3,P0.2,P0.1,P0.0>

P0M0【7：0】	P0M1【7：0】	I/O 口模式
0	0	准双向口(传统 8051 I/O 口模式)，灌电流可达 20mA，拉电流为 230μA，由于制造误差，实际为 250uA ~ 160uA
0	1	推挽输出（强上拉输出，可达 20mA，尽量少用）
1	0	仅为输入（高阻）
1	1	开漏(Open Drain)，内部上拉电阻断开，要外加

举例：

MOV P1M0,#11000000B

MOV P1M1,#10100000B

；P1.7 为开漏，P1.6 为高阻输入，P1.5 为强推挽输出，P1.4/P1.3/P1.2/P1.1/P1.0 为弱上拉

注意：

虽然每个 I/O 口在弱上拉时都能承受 20mA 的灌电流（还是要加限流电阻，如 1K，560 Ω 等），在强推挽输出时都能输出 20mA 的拉电流，但是整个芯片的工作电流推荐不要超过 55mA。

1. 准双向口输出配置

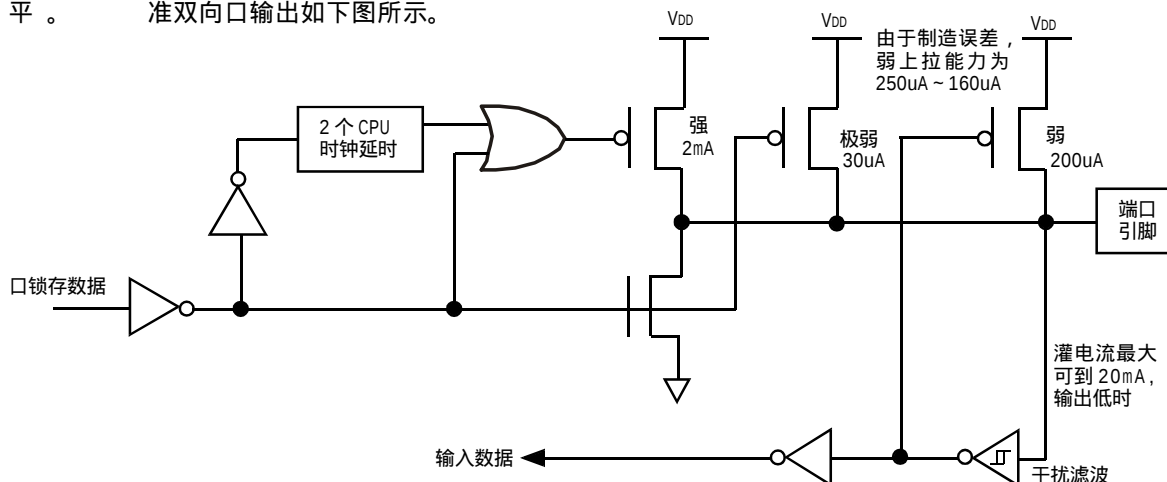
准双向口输出类型可用作输出和输入功能而不需重新配置口线输出状态。这是因为当口线输出为 1 时驱动能力很弱，允许外部装置将其拉低。当引脚输出为低时，它的驱动能力很强，可吸收相当大的电流。准双向口有 3 个上拉晶体管适应不同的需要。

在 3 个上拉晶体管中，有 1 个上拉晶体管称为“弱上拉”，当口线寄存器为 1 且引脚本身也为 1 时打开。此上拉提供基本驱动电流使准双向口输出为 1。如果一个引脚输出为 1 而由外部装置下拉到低时，弱上拉关闭而“极弱上拉”维持开状态，为了把这个引脚强拉为低，外部装置必须有足够的灌电流能力使引脚上的电压降到门槛电压以下。

第 2 个上拉晶体管，称为“极弱上拉”，当口线锁存为 1 时打开。当引脚悬空时，这个极弱的上拉源产生很弱的上拉电流将引脚上拉为高电平。

第 3 个上拉晶体管称为“强上拉”。当口线锁存器由 0 到 1 跳变时，这个上拉用来加快准双向口由逻辑 0 到逻辑 1 转换。当发生这种情况时，强上拉打开约 2 个机器周期以使引脚能够迅速地上拉到高电平。

准双向口输出如下图所示。



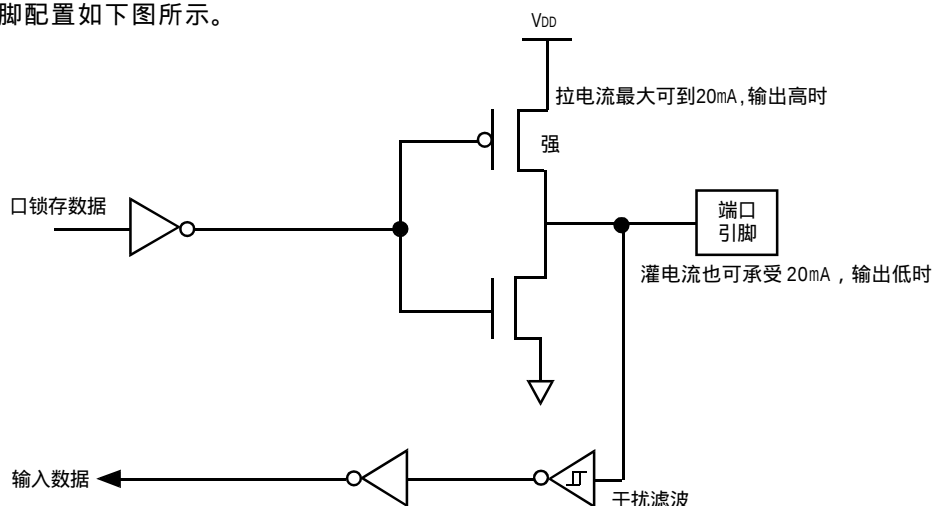
STC12LE5410 系列单片机为 3V 器件，如果用户在引脚加上 5V 电压，将会有电流从引脚流向 VDD，这样导致额外的功率消耗。因此，建议不要在准双向口模式中向 3V 单片机引脚施加 5V 电压，如使用的话，要加限流电阻，或用二极管做输入隔离，或用三极管做输出隔离。

准双向口带有一个施密特触发输入以及一个干扰抑制电路。

2. 推挽输出配置

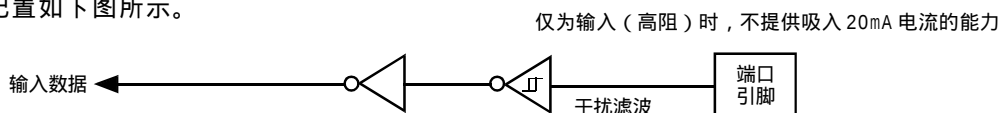
推挽输出配置的下拉结构与开漏输出以及准双向口的下拉结构相同，但当锁存器为 1 时提供持续的强上拉。推挽模式一般用于需要更大驱动电流的情况。

推挽引脚配置如下图所示。



3. 仅为输入（高阻）配置

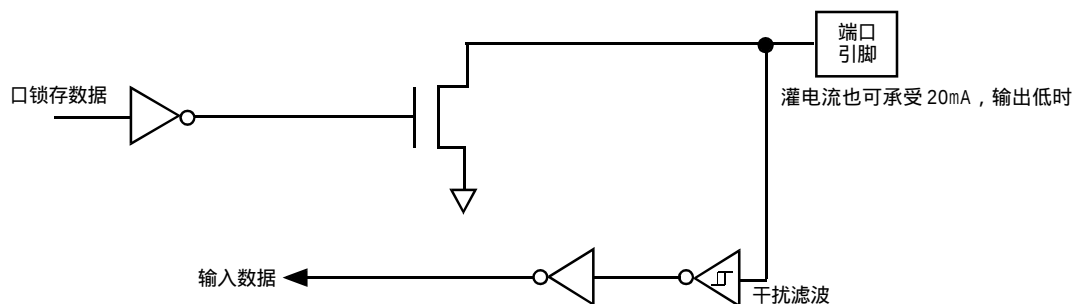
输入口配置如下图所示。



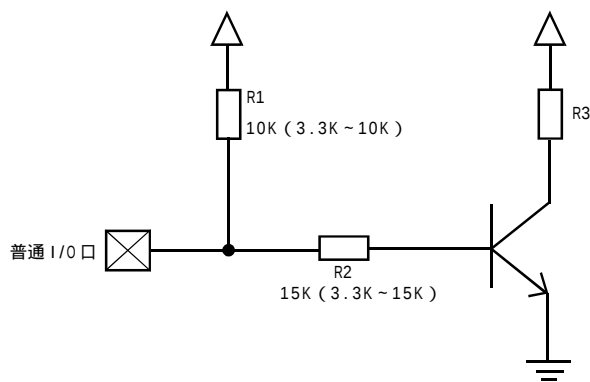
输入口带有一个施密特触发输入以及一个干扰抑制电路。

4. 开漏输出配置

当口线锁存器为 0 时，开漏输出关闭所有上拉晶体管。当作为一个逻辑输出时，这种配置方式必须有外部上拉，一般通过电阻外接到 VDD。这种方式的下拉与准双向口相同。输出口线配置如下图所示。开漏端口带有一个施密特触发输入以及一个干扰抑制电路。



一种典型三极管控制电路



如果用弱上拉控制，建议加上拉电阻 R1（3.3K ~ 10K），如果不加上拉电阻 R1（3.3K ~ 10K），建议 R2 的值在 15K 以上，或用强推挽输出

A/D 及 A/D 转换控制寄存器 ADC_CONTR/ADC_DATA

STC12C5410AD 系列带 A/D 转换的单片机在 P1 口，有 8 路 10 位高速 A/D 转换器，速度可达 100KHz。P1.7 - P1.0 共 8 路电压输入型 A/D，可做温度检测、电池电压检测、按键扫描、频谱检测等。上电复位后 P1 口为弱上拉型 I/O 口，用户可以通过软件设置将 8 路中的任何一路设置为 A/D 转换，不需作为 A/D 使用的口可继续作为 I/O 口使用。需作为 A/D 使用的口需先将其设置为高阻输入或开漏模式。在 P1M0、P1M1 中对相应的位进行设置。

P1M0【7:0】 地址：91h	P1M1【7:0】 地址：92h	I/O 口模式（P1.x 如做 A/D 使用，需先将其设置成开漏或高阻输入）
0	0	准双向口（传统 8051 I/O 口模式）， 灌电流可达 20mA，拉电流为 230μA
0	1	推挽输出（强上拉输出，可达 20mA，尽量少用）
1	0	仅为输入（高阻），如果该 I/O 口需作为 A/D 使用，可选此模式
1	1	开漏（Open Drain），如果该 I/O 口需作为 A/D 使用，可选此模式

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset Value
ADC_CONTR	C5h	A/D 转换控制寄存器	ADC_POWER	SPEED1	SPEED0	ADC_FLAG	ADC_START	CHS2	CHS1	CHS0	0xx0,0000
ADC_DATA	C6h	A/D 转换结果寄存器,高8位	-	-	-	-	-	-	-	-	XXXX,XXXX
ADC_LOW2	BEh	A/D 转换结果寄存器,低2位	-	-	-	-	-	-	-	-	XXXX,XXXX

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_POWER	SPEED1	SPEED0	ADC_FLAG	ADC_START	CHS2	CHS1	CHS0	0xx0,0000
-------------	-----------	--------	--------	----------	-----------	------	------	------	-----------

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：模数转换器(ADC)转换启动控制位，设置为“1”时，开始转换，转换结束后为 0。

ADC_FLAG：模数转换器转换结束标志位，当 A/D 转换完成后，ADC_FLAG = 1，要由软件清 0。

不管是 A/D 转换完成后由该位申请产生中断，还是由软件查询该标志位 A/D 转换是否结束，当 A/D 转换完成后，ADC_FLAG = 1，一定要软件清 0。

SPEED1，SPEED0：模数转换器转换速度控制位

SPEED1	SPEED0	A/D 转换所需时间
1	1	210 个时钟周期转换一次， CPU 工作频率 20MHz 时， A/D 转换速度约 100KHz
1	0	420 个时钟周期转换一次
0	1	630 个时钟周期转换一次
0	0	840 个时钟周期转换一次

ADC_POWER：ADC 电源控制位。

0：关闭 ADC 电源；1：打开 A/D 转换器电源

启动 AD 转换前一定要确认 AD 电源已打开，AD 转换结束后关闭 AD 电源可降低功耗，也可不关闭。

初次打开内部 A/D 转换模拟电源，需适当延时，等内部模拟电源稳定后，再启动 A/D 转换

建议启动 A/D 转换后，在 A/D 转换结束之前，不改变任何 I/O 口的状态，有利于高精度 A/D 转换

ADC_DATA / ADC_LOW2 特殊功能寄存器： A/D 转换结果特殊功能寄存器

ADC_DATA	C6h	A/D 转换结果寄存器,全部8位有效，为10位A/D转换结果的高8位	-	-	-	-	-	-	-	-	XXXX,XXXX
ADC_LOW2	BEh	A/D 转换结果寄存器,只有低2位有效，为10位A/D转换结果的低2位	x	x	x	x	x	x	-	-	XXXX,XXXX

模拟 / 数字转换结果计算公式如下： $结果(ADC_DATA[7:0],ADC_LOW2[1:0]) = 1024 \times V_{in} / V_{cc}$

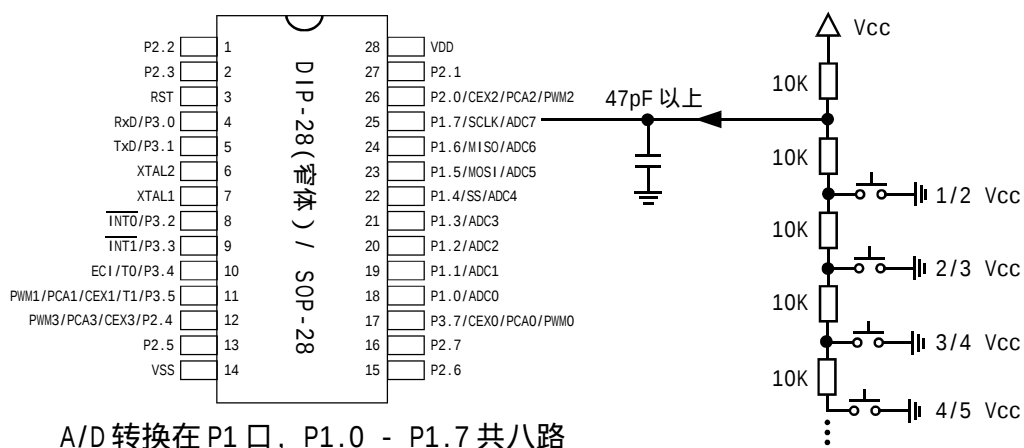
Vin 为模拟输入通道输入电压，Vcc 为单片机实际工作电压，用单片机工作电压作为模拟参考电压。

取 ADC_DATA 的 8 位为 ADC 转换的高 8 位，取 ADC_LOW2 的低 2 位为 ADC 转换的低 2 位，则为 10 位精度。

如果舍弃 ADC_LOW2 的低 2 位，只用 ADC_DATA 寄存器的 8 位，则 A/D 转换结果为 8 位精度。

结果 $ADC_DATA[7:0] = 256 \times V_{in} / V_{cc}$

A/D 转换典型应用线路，按键扫描



A/D 转换功能汇编程序示例

```

; /* --- STC International Limited ----- */
; /* --- 宏晶科技 姚永平 设计 2006/1/6 V1.0 ----- */
; /* --- 演示 STC12C5410AD 系列 MCU 的 A/D 转换功能 - */
; /* --- 演示 STC12C2052AD 系列 MCU 的 A/D 转换功能 - */
; /* --- Mobile: 13922805190 ----- */
; /* --- Fax: 0755-82944243 ----- */
; /* --- Tel: 0755-82948409 ----- */
; /* --- Web: www.mcu-memory.com ----- */

```

;ADC DEMO_5410_ASM.ASM 汇编程序演示 STC12C5410AD 系列 MCU 的 A/D 转换功能。
;本演示程序在 STC-ISP Ver 3.0A.PCB 的下载编程工具上测试通过,相关的 A/D 转换结果在 P1 口上显示
;转换结果也以 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。
;时钟 18.432MHz, 波特率 = 9600。
;转换结果也在 P1 口利用 LED 显示出来, 方便观察。

```

LED_MCU_START EQU P3.7
ADC_CONTR EQU 0C5H ;A/D 转换寄存器
ADC_DATA EQU 0C6H ;A/D 转换结果寄存器, 为 10 位 A/D 转换结果的高 8 位
;ADC_LOW2 EQU 0BEH ;A/D 转换结果寄存器, 低 2 位有效, 为 10 位 A/D 转换结果的低 2 位
;          ;如果不用 ADC_LOW2 的低 2 位,只用 ADC_DATA 的 8 位,则为 8 位 A/D 转换
P1M0 EQU 91H ;P1 口模式选择寄存器 0
P1M1 EQU 92H ;P1 口模式选择寄存器 1

```

```

ADC_Power_On_Speed_Channel_0 EQU 11100000B ;P1.0 作为 A/D 输入
ADC_Power_On_Speed_Channel_1 EQU 11100001B ;P1.1 作为 A/D 输入
ADC_Power_On_Speed_Channel_2 EQU 11100010B ;P1.2 作为 A/D 输入
ADC_Power_On_Speed_Channel_3 EQU 11100011B ;P1.3 作为 A/D 输入
ADC_Power_On_Speed_Channel_4 EQU 11100100B ;P1.4 作为 A/D 输入
ADC_Power_On_Speed_Channel_5 EQU 11100101B ;P1.5 作为 A/D 输入
ADC_Power_On_Speed_Channel_6 EQU 11100110B ;P1.6 作为 A/D 输入
ADC_Power_On_Speed_Channel_7 EQU 11100111B ;P1.7 作为 A/D 输入

```

```

;-----

```

;定义变量

```

ADC_Channel_0_Result EQU 30H ;0 通道 A/D 转换结果
ADC_Channel_1_Result EQU 31H ;1 通道 A/D 转换结果
ADC_Channel_2_Result EQU 32H ;2 通道 A/D 转换结果
ADC_Channel_3_Result EQU 33H ;3 通道 A/D 转换结果
ADC_Channel_4_Result EQU 34H ;4 通道 A/D 转换结果
ADC_Channel_5_Result EQU 35H ;5 通道 A/D 转换结果
ADC_Channel_6_Result EQU 36H ;6 通道 A/D 转换结果
ADC_Channel_7_Result EQU 37H ;7 通道 A/D 转换结果

```

```

;-----
    ORG    0000H
    LJMP   MAIN

    ORG    0050H
MAIN:
    CLR    LED_MCU_START        ;MCU 工作指示灯 LED_MCU_START  EQU  P3.7
    MOV     SP, #7FH            ;设置堆栈

    ACALL  Initiate_RS232        ;初始化串口

    ACALL  ADC_Power_On          ;开 ADC 电源， 第一次使用时要打开内部模拟电源
                                ;开 ADC 电源， 可适当加延时， 1mS 以内就足够了

    ACALL  Set_P12_Open_Drain    ;设置 P1.2 为开漏
    ACALL  Set_ADC_Channel_2     ;设置 P1.2 作为 A/D 转换通道

    ACALL  Get_AD_Result         ;测量电压并且取 A/D 转换结果
    ACALL  Send_AD_Result        ;发送转换结果到 PC 机

    ACALL  Set_P12_Normal_IO     ;设置 P1.2 为普通 IO
    MOV     A, ADC_Channel_2_Result ;用 P1 口显示 A/D 转换结果
    CPL     A
    MOV     P1, A

Wait_Loop:
    SJMP   Wait_Loop            ;停机

;-----
;-----
;-----
;-----
Initiate_RS232:                  ;串口初始化
    CLR     ES                  ;禁止串口中断
    MOV     TMOD, #20H          ;设置 T1 为波特率发生器
    MOV     SCON, #50H          ;0101,0000 8 位数据位， 无奇偶校验
    MOV     TH1, #0FBH          ;18.432MHz 晶振， 波特率 = 9600
    MOV     TL1, #0FBH

    SETB    TR1                 ;启动 T1
    RET

;-----
Send_Byte:
    CLR     TI
    MOV     SBUF, A
Send_Byte_Wait_Finish:
    JNB     TI, Send_Byte_Wait_Finish
    CLR     TI
    RET

```

;-----

ADC_Power_On:

```
PUSH ACC
ORL ADC_CONTR, #80H      ;开 A/D 转换电源
MOV A, #20H
ACALL Delay              ;开 A/D 转换电源后要加延时, 1mS 以内就足够了
POP ACC
RET
```

;-----

;设置 P1.2, 设置 A/D 通道所在的 I/O 为开漏模式

Set_P12_Open_Drain:

```
PUSH ACC
MOV A, #00000100B
ORL P1M0, A
ORL P1M1, A
POP ACC
RET
```

;-----

;设置 P1.2 为普通 I/O

Set_P12_Normal_IO:

```
PUSH ACC
MOV A, #11111011B
ANL P1M0, A
ANL P1M1, A
POP ACC
RET
```

;-----

Set_ADC_Channel_2:

```
MOV ADC_CONTR, #ADC_Power_On_Speed_Channel_2
                                ;选择 P1.2 作为 A/D 转换通道
MOV A, #05H                   ;更换 A/D 转换通道后要适当延时, 使输入电压稳定
                                ;以后如果不更换 A/D 转换通道的话, 不需要加延时
ACALL Delay                   ;切换 A/D 转换通道, 加延时 20uS ~ 200uS 就可以了, 与输入电压源的内阻有关
RET
```

;-----

Send_AD_Result:

```
PUSH ACC
MOV A, ADC_Channel_2_Result   ;取 AD 转换结果
ACALL Send_Byte               ;发送转换结果到 PC 机
POP ACC
RET
```

```

;-----
Get_AD_Result:
    PUSH    ACC                      ;入栈保护
    MOV     ADC_DATA, #0
    ORL     ADC_CONTR, #00001000B   ;启动 AD 转换

Wait_AD_Finishe:
    MOV     A, #00010000B           ;判断 AD 转换是否完成
    ANL     A, ADC_CONTR
    JZ      Wait_AD_Finishe         ;AD 转换尚未完成, 继续等待

    ANL     ADC_CONTR, #11100111B   ;清 0 ADC_FLAG, ADC_START 位, 停止 A/D 转换

    MOV     A, ADC_DATA
    MOV     ADC_Channel_2_Result, A ;保存 AD 转换结果
    POP     ACC
    RET

;-----
Delay:
    PUSH    02                      ;将寄存器组 0 的 R2 入栈
    PUSH    03                      ;将寄存器组 0 的 R3 入栈
    PUSH    04                      ;将寄存器组 0 的 R4 入栈
    MOV     R4, A

Delay_Loop0:
    MOV     R3, #200                ;2 CLOCK -----+
Delay_Loop1:
    MOV     R2, #249                ;2 CLOCK -----+ |
Delay_Loop:
    DJNZ    R2, Delay_Loop           ;4 CLOCK      | |
    DJNZ    R3, Delay_Loop1          ;4 CLOCK -----+ |
    DJNZ    R4, Delay_Loop0          ;4 CLOCK -----+ |

    POP     04
    POP     03
    POP     02
    RET

;-----
END

```

看门狗应用

适用型号： STC12C5410AD 系列

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

Symbol 符号 Function 功能

WDT_FLAG When WDT overflows, this bit is set. It can be cleared by software.

看门狗溢出标志位,当溢出时,该位由硬件置1,可用软件将其清0。

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
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 = (12 x Pre-scale x 32768) / Oscillator frequency

设时钟为 12MHz :

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
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 :

看门狗溢出时间 = $(12 \times \text{Pre-scale} \times 32768) / 11059200 = \text{Pre-scale} \times 393216 / 11059200$

PS2	PS1	PS0	Pre-scale 预分频	WDT Period @11.0592MHz
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, #00111100B; 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 = 1, PS2 = 1, PS1 = 0, PS0 = 0
...
Main_Loop:
LCALL Display_Loop
LCALL Keyboard_Loop
...
MOV WDT_CONTR, #00111100B ; 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()
{
    ...
    WDT_CONTR = 0x3c;
    /* 0011,1100 EN_WDT = 1,CLR_WDT = 1,IDLE_WDT = 1,PS2 = 1,PS1 = 0,PS0 = 0 */
    while(1){
        display();
        keyboard();
        ...
        WDT_CONTR = 0x3c; /* 喂狗, 不要用 WDT_CONTR = WDT_CONTR | 0x10; */
    }
}
```

; 本程序验证 STC12C5410AD 系列及 STC12C2052AD 系列单片机的看门狗及其溢出时间计算公式

; 本演示程序在 STC-ISP Ver 3.0A.PCB 的下载编程工具上测试通过, 相关的工作状态在 P1 口上显示

; 看门狗及其溢出时间 = (12 * Pre_scale * 32768) / Oscillator frequency

WDT_CONTR EQU 0E1H ; 看门狗地址

WDT_TIME_LED EQU P1.5 ; 用 P1.5 控制看门狗溢出时间指示灯,

; 看门狗溢出时间可由该指示灯亮的时间长度或熄灭的时间长度表示

WDT_FLAG_LED EQU P1.7 ; 用 P1.7 控制看门狗溢出复位指示灯, 如点亮表示为看门狗溢出复位

Last_WDT_Time_LED_Status EQU 00H ; 位变量, 存储看门狗溢出时间指示灯的上一次状态位

; WDT 复位时间(所用的 Oscillator frequency = 18.432MHz):

; Pre_scale_Word EQU 00111100B ; 清 0, 启动看门狗, 预分频数=32, 0.68S

Pre_scale_Word EQU 00111101B ; 清 0, 启动看门狗, 预分频数=64, 1.36S

; Pre_scale_Word EQU 00111110B ; 清 0, 启动看门狗, 预分频数=128, 2.72S

; Pre_scale_Word EQU 00111111B ; 清 0, 启动看门狗, 预分频数=256, 5.44S

ORG 0000H

AJMP MAIN

ORG 0100H

MAIN:

MOV A, WDT_CONTR ; 检测是否为看门狗复位

ANL A, #10000000B

JNZ WDT_Reset ; WDT_CONTR.7 = 1, 看门狗复位, 跳转到看门狗复位程序

; WDT_CONTR.7 = 0, 上电复位, 冷启动, RAM 单元内容为随机值

SETB Last_WDT_Time_LED_Status ; 上电复位,

; 初始化看门狗溢出时间指示灯的状态位 = 1

CLR WDT_TIME_LED

; 上电复位, 点亮看门狗溢出时间指示灯

MOV WDT_CONTR, #Pre_scale_Word ; 启动看门狗

WAIT1:

SJMP WAIT1 ; 循环执行本语句(停机), 等待看门狗溢出复位

; WDT_CONTR.7 = 1, 看门狗复位, 热启动, RAM 单元内容不变, 为复位前的值

WDT_Reset: ; 看门狗复位, 热启动

CLR WDT_FLAG_LED ; 是看门狗复位, 点亮看门狗溢出复位指示灯

JB Last_WDT_Time_LED_Status, Power_Off_WDT_TIME_LED; 为 1 熄灭相应的灯, 为 0 亮相应灯

; 根据看门狗溢出时间指示灯的上一次状态位设置 WDT_TIME_LED 灯,

; 若上次亮本次就熄灭, 若上次熄灭本次就亮

CLR WDT_TIME_LED ; 上次熄灭本次点亮看门狗溢出时间指示灯

CPL Last_WDT_Time_LED_Status ; 将看门狗溢出时间指示灯的上一次状态位取反

WAIT2:

SJMP WAIT2 ; 循环执行本语句(停机), 等待看门狗溢出复位

Power_Off_WDT_TIME_LED:

SETB WDT_TIME_LED ; 上次亮本次就熄灭看门狗溢出时间指示灯

CPL Last_WDT_Time_LED_Status ; 将看门狗溢出时间指示灯的上一次状态位取反

WAIT3:

SJMP WAIT3 ; 循环执行本语句(停机), 等待看门狗溢出复位

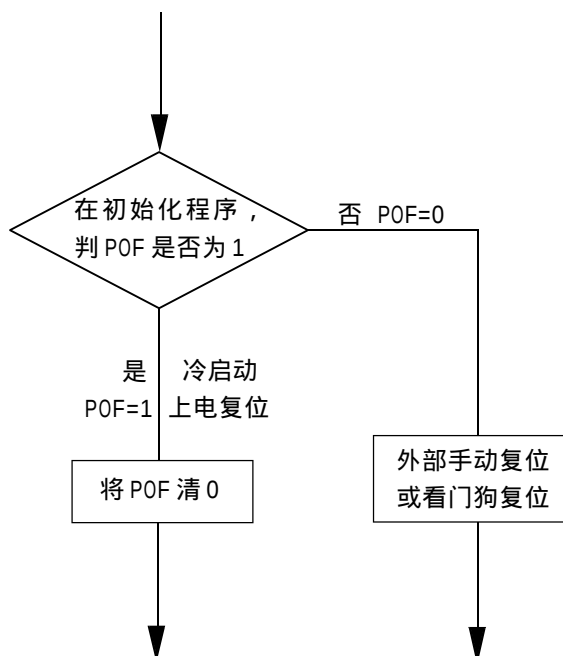
END

PCON 寄存器的高级应用，上电复位标志，进入掉电模式

Mnemonic	Add	Name	7	6	5	4	3	2	1	0	Reset value
PCON	87h	Power Control	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL	0011,0000

POF：上电复位标志位，单片机停电后，上电复位标志位为 1，可由软件清 0。

实际应用：要判断是上电复位（冷启动），还是外部复位脚输入复位信号产生的复位，还是内部看门狗复位，可通过如下方法来判断：



PD：将其置 1 时，进入 Power Down 模式，可由外部中断低电平触发或下降沿触发中断模式唤醒。

进入掉电模式时，外部时钟停振，CPU、定时器、串行口全部停止工作，只有外部中断继续工作。

IDL：将其置 1 时，进入 IDLE 模式（空闲），除 CPU 不工作外，其余仍继续工作，可由任何一个中断唤醒。

现 C 版本开始大量供货，C 版本 IDLE 模式可正常使用（原 A 版本和 B 版本建议不要用 IDLE 模式）。

STC12C5410AD 系列 1T 单片机通过外部中断从掉电模式唤醒

```
*****
;
;Wake Up Idle and Wake Up Power Down
*****
ORG    0000H
AJMP  MAIN

ORG    0003H
int0_interrupt:
    CLR    P1.7           ;点亮 P1.7 LED 表示已响应 int0 中断
    ACALL  delay          ;延时是为了便于观察，实际应用不需延时
    CLR    EA             ;关闭中断，简化实验。实际应用不需关闭中断
    RETI

ORG    0013H
int1_interrupt:
    CLR    P1.6           ;点亮 P1.6 LED 表示已响应 int1 中断
    ACALL  delay          ;延时是为了便于观察，实际应用不需延时
    CLR    EA             ;关闭中断，简化实验。实际应用不需关闭中断
    RETI

ORG    0100H
delay:
    CLR    A
    MOV    R0, A
    MOV    R1, A
    MOV    R2, #02
delay_loop:
    DJNZ   R0, delay_loop
    DJNZ   R1, delay_loop
    DJNZ   R2, delay_loop
    RET

main:
    MOV    R3, #0         ;P1 LED 递增方式变化，表示程序开始运行
main_loop:
    MOV    A, R3
    CPL    A
    MOV    P1, A
    ACALL  delay
```

```
    INC    R3
    MOV    A, R3
    SUBB   A, #18H
    JC     main_loop

    MOV    P1, #0FFH      ;熄灭全部灯表示进入 Power Down 状态

    CLR    IT0             ;设置低电平激活外部中断
;   SETB   IT0

    SETB   EX0            ;允许外部中断0

    CLR    IT1            ;设置低电平激活外部中断
;   SETB   IT1
    SETB   EX1            ;允许外部中断1
;   SETB   ET0 ;如果是 STC12C2052AD 系列 A 版本,
                        ;要由外部中断1唤醒,“ET0=1”是必须的,硬件就这样做的,C版本就不需要
                        ;外部中断0就无此必要,建议 Powerdown 用外部中断0唤醒

    SETB   EA             ;开中断,若不开中断就不能唤醒 Power Down
```

;下条语句将使 MCU 进入 idle 状态或 Power Down 状态
;低电平激活外部中断可以将 MCU 从 Power Down 状态中唤醒
;其方法为:将外部中断脚拉低

```
    MOV    A, PCON        ;令 PD=1, 进入 Power Down 状态, PD = PCON.2
    ORL    A, #02H
    MOV    PCON, A

    MOV    PCON, #01H     ;删除本语句前的";", 同时将前3条语句前加上注释符号";",
                        ;令 IDL=1, 可进入 idle 状态, IDL = PCON.1

    MOV    P1, #0DFH      ;请注意:
                        ; 1.外部中断使MCU退出 Power Down 状态,执行本条指令后
                        ;响应中断, 表现为 P1.5 与 P1.7 的 LED 同时亮(INT0 唤醒)
                        ; 2.外部中断使MCU退出 idle 状态,先响应中断然后再执行本
                        ;条指令, 表现为 P1.7 的 LED 先亮(INT0 唤醒)P1.5 的 LED 后亮

WAIT1:
    SJMP   WAIT1          ;跳转到本语句, 停机
    END

;A 版本和 B 版本建议不要用 IDLE 模式,现 C 版本可以正常使用
```

STC12C5410AD 系列单片机进入掉电模式前的 I/O 设置特别注意事项 --- 仅针对 A 版本，现供货的是最新版本 C 版本

STC12C5410AD 系列单片机最新的 C 版本已开始大量供货，已解决此问题，B 版本也无问题，只有 A 版本需要注意

针对 A 版本如下应用注意事项：

STC12Cxx 要 Power Down < 0.1uA，进入 Power Down 之前

1.

如果您系统外部所接的口是输入，我们 STC12Cxx 是输出，

请将我们的该 I/O 设为 Push-Pull 输出

外部是低，你就要为低；外部是高，你就要为高，否则就 > 0.1uA

如果是后续的版本，不一定要设置成 Push-Pull 输出，用弱上拉就可以了。

2.

如果您系统外部所接的口是输出，我们 STC12Cxx 是输入，

请将我们的该 I/O 设为 Only-Input，高阻仅为输入

如果是后续的版本，不一定要设置成 Only-Input 输出，用弱上拉就可以了。

3.

如果您系统外部所接的口是高阻，我们 STC12Cxx ，

请将我们的该 I/O 设为 Push-Pull 输出

请不要将我们的该 I/O 设为 Only-Input/ 高阻

如果是后续的版本，不一定要设置成 Push-Pull 输出，用弱上拉就可以了。

现在 C 版本已开始供货，定货时指定 C 版本就可以了

STC12C5410AD 系列 1T 8051 单片机 IAP 应用

STC12C5410AD 系列 1T 8051 单片机内部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

STC12C5410AD 系列 1T 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	-	-	-	-	-	-	MS1	MS0	xxxx,xx00
ISP_TRIG	E6h	ISP/IAP Flash Command Trigger									xxxx,xxxx
ISP_CONTR	E7h	ISP/IAP Control Register	ISPEN	SWBS	SWRST	CMD_FAIL	1	WT2	WT1	WT0	0000,1000

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		
-	-	-	-	-	-	0	0	Standby 待机模式，无ISP操作
-	-	-	-	-	-	0	1	从用户的应用程序区对 "Data Flash/EEPROM区" 进行字节读
-	-	-	-	-	-	1	0	从用户的应用程序区对 "Data Flash/EEPROM区" 进行字节编程
-	-	-	-	-	-	1	1	从用户的应用程序区对 "Data Flash/EEPROM区" 进行扇区擦除

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，STC12C5412AD/STC12LE5412AD 除外。STC12C5410AD 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	CMD_FAIL	1	WT2	WT1	WT0	0000,1000

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ISP 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。

CMD_FAIL: 如果送了ISP/IAP 命令，并对ISP_TRIG 送 46h/B9h 触发失败，则为1，需由软件清零。

设置等待时间			CPU 等待时间 (CPU 的工作时钟)			
WT2	WT1	WT0	Read	Program	Sector Erase	Recommended System Clock
1	1	1	2	55	21012	1MHz
1	1	0	2	110	42024	2MHz
1	0	1	2	165	63036	3MHz
1	0	0	2	330	126072	6MHz
0	1	1	2	660	252144	12MHz
0	1	0	2	1100	420240	20MHz
0	0	1	2	1320	504288	24MHz
0	0	0	2	1760	672384	30MHz

STC12C5410AD 系列单片机内部可用 Data Flash(EEPROM)的地址(与程序空间是分开的):

如果对应用程序区进行 IAP 写数据,则该语句会被单片机忽略,继续执行下一句。

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。

但 STC12C5412AD/STC12LE5412AD 在应用程序区可以修改应用程序区(灵活)。

STC12C5410AD,STC12C5410,STC12LE5410AD,STC12LE5410

STC12C5408AD,STC12C5408,STC12LE5408AD,STC12LE5408

STC12C5406AD,STC12C5406,STC12LE5406AD,STC12LE5406

STC12C5404AD,STC12C5404,STC12LE5404AD,STC12LE5404

STC12C5402AD,STC12C5402,STC12LE5402AD,STC12LE5402

STC12C5401AD,STC12C5401,STC12LE5401AD,STC12LE5401

系列单片机内部可用 Data Flash(EEPROM)的地址:

第一扇区		第二扇区		每个扇区 512 字节 共 4 个扇区 建议同一次修改的数据放在同一个扇区,不必用满,当然可全用,用满则为 2K 字节 EEPROM。由于擦除是按扇区擦除,所以每个扇区用的越少越方便,256 个字节以内较合理。
起始地址	结束地址	起始地址	结束地址	
2800h	29FFh	2A00h	2BFFh	
第三扇区		第四扇区		
起始地址	结束地址	起始地址	结束地址	
2C00h	2DFFh	2E00h	2FFFh	

STC12C5412,STC12C5412AD,STC12LE5412,STC12LE5412AD 单片机可对自身内部应用程序区进行 IAP/ISP 操作,故所有部分均可当 Data Flash(EEPROM)使用,其地址如下:

第一扇区		第二扇区		第三扇区		第四扇区	
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
0000h	01FFh	0200h	03FFh	0400h	05FFh	0600h	07FFh
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
0800h	09FFh	0A00h	0BFFh	0C00h	0DFFh	0E00h	0FFFh
第九扇区		第十扇区		第十一扇区		第十二扇区	
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
1000h	11FFh	1200h	13FFh	1400h	15FFh	1600h	17FFh
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
1800h	19FFh	1A00h	1BFFh	1C00h	1DFFh	1E00h	1FFFh
第十七扇区		第十八扇区		第十九扇区		第二十扇区	
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
2000h	21FFh	2200h	23FFh	2400h	25FFh	2600h	27FFh
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
2800h	29FFh	2A00h	2BFFh	2C00h	2DFFh	2E00h	2FFFh

每个扇区
512字节

建议同一次
修改的数据
放在一个扇
区，不当然
用全

STC12C5410AD 系列 IAP 应用汇编简介

STC12C5410AD 系列 内部 EEPROM 的应用

;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,30MHz 以下 0,24M 以下 1,
;20MHz 以下 2,12M 以下 3,6M 以下 4,3M 以下 5,2M 以下 6,1M 以下 7,
    
```

;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
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,命令不需改变时,不需重新送命令
MOV ISP_TRIG, #46h ;先送 46h,再送 B9h 到 ISP / IAP 触发寄存器,每次都需如此
MOV ISP_TRIG, #0B9h ;送完 B9h 后,ISP / IAP 命令立即被触发启动
    
```

} 地址需要改变时才需重新送地址

} 此两句可以合成一句,并且只送一次就够了

;CPU 等待 IAP 动作完成后,才会继续执行程序。

```

NOP ;数据读出到 ISP_DATA 寄存器后,CPU 继续执行程序
MOV A, ISP_DATA ;将读出的数据送往 Acc
    
```

;以下语句可不用,只是出于安全考虑而已

```

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, #0 ;送地址高字节单元为 00,指向非 EEPROM 区
;MOV ISP_ADDRL, #0 ;送地址低字节单元为 00,防止误操作
    
```

; 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, 只有数据改变时才需重新送
MOV  ISP_ADDRH, #BYTE_ADDR_HIGH ;送地址高字节
MOV  ISP_ADDRL, #BYTE_ADDR_LOW  ;送地址低字节
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
MOV  ISP_TRIG, #46h          ;先送 46h, 再送 B9h 到 ISP/IAP 触发寄存器, 每次都需如此
MOV  ISP_TRIG, #0B9h         ;送完 B9h 后, ISP/IAP 命令立即被触发起动
```

地址需要改变时才需重新送地址

此两句可合成一句, 并且只送一次就够了

; CPU 等待 IAP 动作完成后, 才会继续执行程序.

```
NOP ; 字节编程成功后, CPU 继续执行程序
```

; 以下语句可不用, 只是出于安全考虑而已

```
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, #0 ;送地址高字节单元为 00, 指向非 EEPROM 区, 防止误操作
;MOV ISP_ADDRL, #0 ;送地址低字节单元为 00, 指向非 EEPROM 区, 防止误操作
```

小常识: (STC 单片机的 Data Flash 当 EEPROM 功能使用)

3 个基本命令 - - - - 字节读, 字节编程, 扇区擦除

字节编程: 如果该字节是“11111, 11111B”, 则可将其中的“1”编程为“0”, 如果该字节中有位为“0”, 则须先将整个扇区擦除, 因为只有“扇区擦除”才可以将“0”变为“1”。

扇区擦除: 只有“扇区擦除”才可能将“0”擦除为“1”。

大建议:

1. 同一次修改的数据放在同一扇区中, 不是同一次修改的数据放在另外的扇区, 就不须读出保护。
2. 如果一个扇区只用一个字节, 那就是真正的 EEPROM, STC 单片机的 Data Flash 比外部 EEPROM 要快很多, 读一个字节 / 编程一个字节 / 擦除一个扇区大概是 10uS/60uS/10mS。
3. 如果在一个扇区中存放了大量的数据, 某次只需要修改其中的一个字节或一部分字节时, 则另外的不需要修改的数据须先读出放在 STC 单片机的 RAM 中, 然后擦除整个扇区, 再将需要保留的数据和需修改的数据一并写回该扇区中。这时每个扇区使用的字节数是使用的越少越方便(不需读出一大堆需保留数据)。

; 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留, 则需将其先读到单片机

; 内部的 RAM 中保存, 再将该扇区擦除, 然后将须保留的数据写回该扇区, 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.

```
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
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,命令不需改变时,不需重新送命令
MOV  ISP_TRIG,   #46h ;先送 46h,再送 B9h 到 ISP/IAP 触发寄存器,每次都需如此
MOV  ISP_TRIG,   #0B9h;送完 B9h 后,ISP/IAP 命令立即被触发起动
```

地址需要改变时
才需重新送地址

此两句可以合成一句,并且只送一次就够了

; CPU 等待 IAP 动作完成后, 才会继续执行程序.

```
NOP ;扇区擦除成功后, CPU 继续执行程序
```

; 以下语句可不用, 只是出于安全考虑而已

```
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, #0 ;送地址高字节单元为 00, 指向非 EEPROM 区
;MOV ISP_ADDRL, #0 ;送地址低字节单元为 00, 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

STC12C5410AD 系列和 STC12C2052AD 系列单片机 EEPROM / IAP 功能测试程序演示

```
;/* --- STC International Limited ----- */
;/* --- 宏晶科技 姚永平 设计 2006/1/6 V1.0 ----- */
;/* --- 演示 STC12C5410AD 系列 MCU EEPROM/IAP 功能 - */
;/* --- 演示 STC12C2052AD 系列 MCU EEPROM/IAP 功能 - */
;/* --- Mobile: 13922805190 ----- */
;/* --- Fax: 0755-82944243 ----- */
;/* --- Tel: 0755-82948409 ----- */
;/* --- Web: www.mcu-memory.com ----- */
;本演示程序在 STC-ISP Ver 3.0A.PCB 的下载编程工具上测试通过,EEPROM 的数据在 P1 口上显示
;-----
;声明与 IAP/ISP/EEPROM 有关的特殊功能寄存器的地址
ISP_DATA EQU 0E2H
ISP_ADDRH EQU 0E3H
ISP_ADDRL EQU 0E4H
ISP_CMD EQU 0E5H
ISP_TRIG EQU 0E6H
ISP_CONTR EQU 0E7H
;定义 ISP/IAP 命令
ISP_IAP_BYTE_READ EQU 1H ;字节读
ISP_IAP_BYTE_PROGRAM EQU 2H ;字节编程,可以将 1 写成 0,要将 1 变成 0,必须执行字节编程
ISP_IAP_SECTOR_ERASE EQU 3H ;扇区擦除,可以将 0 擦成 1,要将 0 变成 1,必须擦除整个扇区
;定义 Flash 操作等待时间及允许 IAP/ISP/EEPROM 操作的常数
;ENABLE_ISP EQU 80H ;系统工作时钟<30MHz 时,对 ISP_CONTR 寄存器设置此值
;ENABLE_ISP EQU 81H ;系统工作时钟<24MHz 时,对 ISP_CONTR 寄存器设置此值
ENABLE_ISP EQU 82H ;系统工作时钟<20MHz 时,对 ISP_CONTR 寄存器设置此值
;ENABLE_ISP EQU 83H ;系统工作时钟<12MHz 时,对 ISP_CONTR 寄存器设置此值
;ENABLE_ISP EQU 84H ;系统工作时钟<6MHz 时,对 ISP_CONTR 寄存器设置此值
;ENABLE_ISP EQU 85H ;系统工作时钟<3MHz 时,对 ISP_CONTR 寄存器设置此值
;ENABLE_ISP EQU 86H ;系统工作时钟<2MHz 时,对 ISP_CONTR 寄存器设置此值
;ENABLE_ISP EQU 87H ;系统工作时钟<1MHz 时,对 ISP_CONTR 寄存器设置此值

DEBUG_DATA EQU 5AH ;是本测试程序选定的 EEPROM 单元的数值如正确应等于的数值
;-----
;选择 MCU 型号
;DATA_FLASH_START_ADDRESS EQU 1000H ;STC12C2052AD 系列单片机的 EEPROM 测试起始地址
DATA_FLASH_START_ADDRESS EQU 2800H ;STC12C5410AD 系列单片机的 EEPROM 测试起始地址
;-----
ORG 0000H
LJMP MAIN
;-----
ORG 0100H
MAIN:
MOV P1,#0F0H ;演示程序开始工作,让 P1.0/P1.1/P1.2/P1.3 控制的灯亮
LCALL Delay ;延时
MOV P1,#0FH ;演示程序开始工作,让 P1.7/P1.6/P1.5/P1.4 控制的灯亮
```

```

        LCALL Delay          ;延时
        MOV     SP, #7FH      ;堆栈指针指向 7FH 单元
;*****
;将 EEPROM 测试起始地址单元的内容读出
MAIN1:
        MOV     DPTR, #DATA_FLASH_START_ADDRESS ;将 EEPROM 测试起始地址送 DPTR 数据指针
        LCALL   Byte_Read
        MOV     40H, A        ;将 EEPROM 的值送 40H 单元保存
        CJNE    A, #DEBUG_DATA, DATA_NOT_EQU_DEBUG_DATA ;如果数据比较不正确,就跳转
DATA_IS_DEBUG_DATA:
;数据是对的,亮 P1.7 控制的灯,然后在 P1 口上将 EEPROM 的数据显示出来
        MOV     P1, #01111111B ;如 (DATA_FLASH_START_ADDRESS)的值等于 #DEBUG_DATA, 亮 P1.7
        LCALL   Delay          ;延时
        MOV     A, 40H        ;将保存在 40H 单元中 EEPROM 的值从 40H 单元送累加器 A
        CPL     A             ;取反的目的是相应的灯亮代表 1, 不亮代表 0
        MOV     P1, A         ;数据是对的, 送 P1 显示
WAIT1:
        SJMP    WAIT1        ;数据是对的, 送 P1 显示后, CPU 在此无限循环执行此句

DATA_NOT_EQU_DEBUG_DATA:
;EEPROM 里的数据是错的,亮 P1.3 控制的灯,然后在 P1 口上将错误的数据显示出来,
;再将该 EEPROM 所在的扇区整个擦除,将正确的数据写入后,亮 P1.5 控制的灯
        MOV     P1, #11110111B ;如 (DATA_FLASH_START_ADDRESS)的值不等于 #DEBUG_DATA, 亮 P1.3
        LCALL   Delay          ;延时
        MOV     A, 40H        ;将保存在 40H 单元中 EEPROM 的值从 40H 单元送累加器 A
        CPL     A             ;取反的目的是相应的灯亮代表 1, 不亮代表 0
        MOV     P1, A         ;数据不对, 送 P1 显示
        LCALL   Delay          ;延时

        MOV     DPTR, #DATA_FLASH_START_ADDRESS ;将 EEPROM 测试起始地址送 DPTR 数据指针
        ACALL   Sector_Erase ;擦除整个扇区
        MOV     DPTR, #DATA_FLASH_START_ADDRESS ;将 EEPROM 测试起始地址送 DPTR 数据指针
        MOV     A, #DEBUG_DATA ;写入 EEPROM 的数据为 #DEBUG_DATA
        ACALL   Byte_Program ;字节编程
        MOV     P1, #11011111B ;将先前亮的 P1.3 灯关闭,再亮 P1.5 灯,代表数据已被修改
WAIT2:
        SJMP    WAIT2        ;字节编程后,CPU 在此无限循环执行此句
;*****
;

```

;读一字节,调用前需打开 IAP 功能,入口:DPTR = 字节地址,返回:A = 读出字节

Byte_Read:

```

MOV     ISP_CONTR, #ENABLE_ISP      ;打开 IAP 功能, 设置 Flash 操作等待时间
MOV     ISP_CMD,   #ISP_IAP_BYTE_READ ;设置为 IAP/ISP/EEPROM 字节读模式命令
MOV     ISP_ADDRH, DPH               ;设置目标单元地址的高 8 位地址
MOV     ISP_ADDRL, DPL               ;设置目标单元地址的低 8 位地址
;CLR    EA
MOV     ISP_TRIG,  #46H               ;先送 46h,再送 B9h 到 ISP/IAP 触发寄存器,每次都需如此
MOV     ISP_TRIG,  #0B9H              ;送完 B9h 后, ISP/IAP 命令立即被触发起动
NOP
MOV     A,    ISP_DATA                ;读出的数据在 ISP_DATA 单元中,送入累加器 A
;SETB   EA
ACALL   IAP_Disable ;关闭 IAP 功能, 清相关的特殊功能寄存器,使 CPU 处于安全状态,
;一次连续的 IAP 操作完成之后建议关闭 IAP 功能,不需要每次都关

RET

```

;字节编程,调用前需打开 IAP 功能,入口:DPTR = 字节地址, A= 须编程字节的数据

Byte_Program:

```

MOV     ISP_CONTR, #ENABLE_ISP      ;打开 IAP 功能, 设置 Flash 操作等待时间
MOV     ISP_CMD,   #ISP_IAP_BYTE_PROGRAM ;设置为 IAP/ISP/EEPROM 字节编程模式命令
MOV     ISP_ADDRH, DPH               ;设置目标单元地址的高 8 位地址
MOV     ISP_ADDRL, DPL               ;设置目标单元地址的低 8 位地址
MOV     ISP_DATA,  A                 ;要编程的数据先送进 ISP_DATA 寄存器
;CLR    EA
MOV     ISP_TRIG,  #46H               ;先送 46h,再送 B9h 到 ISP/IAP 触发寄存器,每次都需如此
MOV     ISP_TRIG,  #0B9H              ;送完 B9h 后, ISP/IAP 命令立即被触发起动
NOP
;SETB   EA
ACALL   IAP_Disable ;关闭 IAP 功能, 清相关的特殊功能寄存器,使 CPU 处于安全状态,
;一次连续的 IAP 操作完成之后建议关闭 IAP 功能,不需要每次都关

RET

```

;擦除扇区, 入口:DPTR = 扇区地址

Sector_Erase:

```

MOV     ISP_CONTR, #ENABLE_ISP      ;打开 IAP 功能, 设置 Flash 操作等待时间
MOV     ISP_CMD,   #03H               ;设置为 IAP/ISP/EEPROM 扇区擦除模式命令
MOV     ISP_ADDRH, DPH               ;设置目标单元地址的高 8 位地址
MOV     ISP_ADDRL, DPL               ;设置目标单元地址的低 8 位地址
;CLR    EA
MOV     ISP_TRIG,  #46H               ;先送 46h,再送 B9h 到 ISP/IAP 触发寄存器,每次都需如此
MOV     ISP_TRIG,  #0B9H              ;送完 B9h 后, ISP/IAP 命令立即被触发起动
NOP
;SETB   EA
ACALL   IAP_Disable ;关闭 IAP 功能, 清相关的特殊功能寄存器,使 CPU 处于安全状态,
;一次连续的 IAP 操作完成之后建议关闭 IAP 功能,不需要每次都关

RET

```

```
;-----
IAP_Disable:
;关闭 IAP 功能, 清相关的特殊功能寄存器,使 CPU 处于安全状态,
;一次连续的 IAP 操作完成之后建议关闭 IAP 功能,不需要每次都关
    MOV    ISP_CONTR, #0          ;关闭 IAP 功能
    MOV    ISP_CMD,  #0          ;清命令寄存器,使命令寄存器无命令,此句可不用
    MOV    ISP_TRIG, #0          ;清命令触发寄存器,使命令触发寄存器无触发,此句可不用
    RET

;-----
Delay:
    CLR    A
    MOV    R0, A
    MOV    R1, A
    MOV    R2, #20H
Delay_Loop:
    DJNZ   R0, Delay_Loop
    DJNZ   R1, Delay_Loop
    DJNZ   R2, Delay_Loop
    RET

;-----

    END

;*****
```

STC12C5410AD 系列单片机定时器的使用

定时器 0 和 1

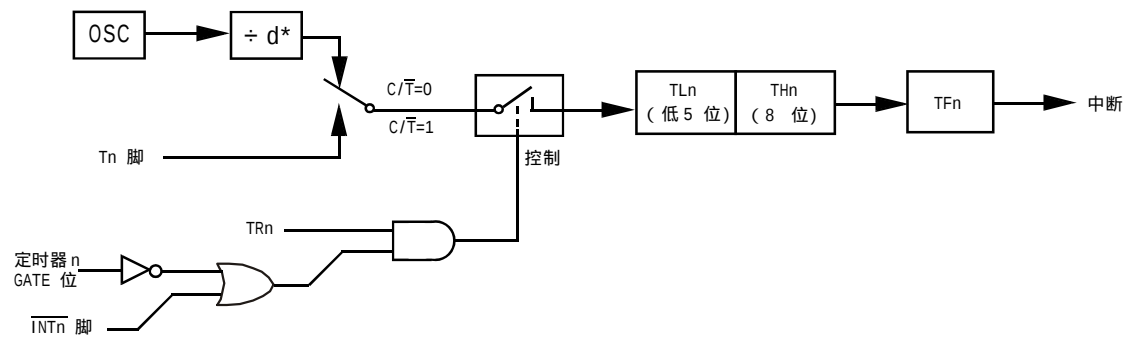
定时和计数功能由特殊功能寄存器 TMOD 的控制位 $C/\overline{T}$ 进行选择，TMOD 寄存器的各位信息如下表所列。可以看出，2 个定时 / 计数器有 4 种操作模式，通过 TMOD 的 M1 和 M0 选择。2 个定时 / 计数器的模式 0、1 和 2 都相同，模式 3 不同，各模式下的功能如下所述。

寄存器 TMOD 各位的功能描述

TMOD	地址：89H		复位值：00H																									
不可位寻址																												
<table><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td>GATE</td><td>C/$\overline{T}$</td><td>M1</td><td>M0</td><td>GATE</td><td>C/$\overline{T}$</td><td>M1</td><td>M0</td></tr><tr><td colspan="4">定时器 1</td><td colspan="4">定时器 0</td></tr></table>					7	6	5	4	3	2	1	0	GATE	C/ $\overline{T}$	M1	M0	GATE	C/ $\overline{T}$	M1	M0	定时器 1				定时器 0			
7	6	5	4	3	2	1	0																					
GATE	C/ $\overline{T}$	M1	M0	GATE	C/ $\overline{T}$	M1	M0																					
定时器 1				定时器 0																								
位	符号	功能																										
TMOD.7/	GATE	TMOD.7 控制定时器 1, 置 1 时只有在在 $\overline{INT1}$ 脚为高及 TR1 控制位置 1 时才 可打开定时器 / 计数器 1。																										
TMOD.3/	GATE	TMOD.3 控制定时器 0, 置 1 时只有在在 $\overline{INT0}$ 脚为高及 TR0 控制位置 1 时才 可打开定时器 / 计数器 0。																										
TMOD.6/	C/ $\overline{T}$	TMOD.6 控制定时器 1 用作定时器或计数器，清零则用作定时器（从内 部系统时钟输入），置 1 用作计数器（从 T1/P3.5 脚输入）																										
TMOD.2/	C/ $\overline{T}$	TMOD.2 控制定时器 0 用作定时器或计数器，清零则用作定时器（从内 部系统时钟输入），置 1 用作计数器（从 T0/P3.4 脚输入）																										
TMOD.5/TMOD.4	M1、M0	定时器 / 计数器 1 模式选择																										
	0 0	13 位定时器 / 计数器，兼容 8048 定时器模式，TL1 只用低 5 位参与分 频，TH1 整个 8 位全用。																										
	0 1	16 位定时器 / 计数器，TL1、TH1 全用																										
	1 0	8 位自动重装载定时器，当溢出时将 TH1 存放的值自动重装入 TL1。																										
TMOD.1/TMOD.0	M1、M0	定时器 / 计数器 1 此时无效（停止计数）。																										
	M1、M0	定时器 / 计数器 0 模式选择																										
	0 0	13 位定时器 / 计数器，兼容 8048 定时器模式，TL0 只用低 5 位参与分 频，TH0 整个 8 位全用。																										
	0 1	16 位定时器 / 计数器，TL0、TH0 全用																										
	1 0	8 位自动重装载定时器，当溢出时将 TH0 存放的值自动重装入 TL0。																										
	1 1	定时器 0 此时作为双 8 位定时器 / 计数器。TL0 作为一个 8 位定时器 / 计 数器，通过标准定时器 0 的控制位控制。TH0 仅作为一个 8 位定时器， 由定时器 1 的控制位控制。																										

1. 模式 0

将定时器设置成模式 0 时类似 8048 定时器，即 8 位计数器带 32 分频的预分频器。下图所示为模式 0 工作方式。此模式下，定时器配置为 13 位的计数器，由 TLn 的低 5 位和 THn 的 8 位所构成。TLn 低 5 位溢出向 THn 进位，THn 计数溢出置位 TCON 中的溢出标志位 TF_n (n=0, 1)。GATE=0 时，如 TR_n=1，则定时器计数。GATE=1 时，允许由外部输入 $\overline{INT1}$ 控制定时器 1， $\overline{INT0}$ 控制定时器 0，这样可实现脉宽测量。TR_n 为 TCON 寄存器内的控制位，TCON 寄存器各位的具体功能描述见 TCON 寄存器各位的具体功能描述表。



* 在 T0x12 = 0 模式下, d=12(12 时钟模式); 在 T0x12 = 1 模式下, d=1(1T)。

图 定时器 / 计数器 0 和定时器 / 计数器 1 的模式 0 : 13 位定时 / 计数器

表 寄存器 TCON 各位的功能描述

TCON 地址: 88H									
可位寻址 复位值: 00H		7	6	5	4	3	2	1	0
		TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
位	符号	功 能							
TCON.7	TF1	定时器 / 计数器 1 溢出标志位。当 T1 被允许计数后, T1 从初值开始加 1 计数, 最高位产生溢出时, 置“1”TF1, 并向 CPU 请求中断, 当 CPU 响应时, 由硬件清“0”TF1, TF1 也可以由程序查询或清“0”。							
TCON.6	TR1	定时器 T1 的运行控制位。该位由软件置位和清零。当 GATE (TMOD.7) = 0, TR1=1 时就允许 T1 开始计数, TR1=0 时禁止 T1 计数。当 GATE (TMOD.7) = 1, TR1=1 且 INT1 输入高电平时, 才允许 T1 计数。							
TCON.5	TF0	定时器 / 计数器 0 溢出标志位。当 T0 被允许计数后, T0 从初值开始加 1 计数, 最高位产生溢出时, 置“1”TF0, 并向 CPU 请求中断, 当 CPU 响应时, 由硬件清“0”TF0, TF0 也可以由程序查询或清“0”。							
TCON.4	TR0	定时器 T0 的运行控制位。该位由软件置位和清零。当 GATE (TMOD.3) = 0, TR0=1 时就允许 T0 开始计数, TR0=0 时禁止 T0 计数。当 GATE (TMOD.3) = 1, TR0=1 且 INTO 输入高电平时, 才允许 T0 计数。							
TCON.3	IE1	外部中断 1 中断请求标志位。当主机响应中断转向该中断服务程序执行时, 由内部硬件自动将 IE1 位清 0。							
TCON.2	IT1	外部中断 1 触发方式控制位。IT1=0 时, 外部中断 1 为低电平触发方式, 当 $\overline{INT1}$ (P3.3) 输入低电平时, 置位 IE1。采用低电平触发方式时, 外部中断源 (输入到 $\overline{INT1}$) 必须保持低电平有效, 直到该中断被 CPU 响应, 同时在该中断服务程序执行完之前, 外部中断源 <u>必须被清除</u> (P3.3 要变高), 否则将产生另一次中断。当 IT1=1 时, 则外部中断 1 ($\overline{INT1}$) 端口由“1”→“0”下降沿跳变, 激活中断请求标志位 IE1, 向主机请求中断处理。							
TCON.1	IE0	外部中断 0 中断请求标志位。当主机响应中断转向该中断服务程序执行时, 由内部硬件自动将 IE0 位清 0。							
TCON.0	IT0	外部中断 0 触发方式控制位。IT0=0 时, 外部中断 0 为低电平触发方式, 当 $\overline{INT0}$ (P3.2) 输入低电平时, 置位 IE0。采用低电平触发方式时, 外部中断源 (输入到 $\overline{INT0}$) 必须保持低电平有效, 直到该中断被 CPU 响应, 同时在该中断服务程序执行完之前, 外部中断源 <u>必须被清除</u> (P3.2 要变高), 否则将产生另一次中断。当 IT0=1 时, 则外部中断 0 ($\overline{INT0}$) 端口由“1”→“0”下降沿跳变, 激活中断请求标志位 IE1, 向主机请求中断处理。							

该 13 位寄存器包含 THn 全部 8 个位及 TLn 的低 5 位。TLn 的高 3 位不定, 可将其忽略。置位运行标志 (TRn) 不能清零此寄存器。模式 0 的操作对于定时器 0 及定时器 1 都是相同的。2 个不同的 GATE 位 (TMOD.7 和 TMOD.3) 分别分配给定时器 1 及定时器 0。