高等教育电子信息类规划教材

电子技术专业英语

丁向荣　赖金志　饶瑞福　编　著
林俊良　姚永平　主　审

電子工業出版社
Publishing House of Electronics Industry
北京・BEIJING

内 容 简 介

本教材选取了电子技术类专业的基础内容（也可以说是主题内容）：电子元器件、电子产品生产工艺、电子产品检验、电子系统设计共计 4 个部分作为教材内容，其中，前 3 个部分采用图解的方式，“ 望图知义 ”达到巩固电子元器件、电子产品生产工艺、电子产品检验核心知识的目的，同时掌握好相关英文词汇；“ 电子系统设计 ” 部分则选用单片机应用技术作为学习内容，系统地以英文形式学习单片机的原理与应用，“ 温故而知英 ”，既可进一步提升单片机应用能力，又能较系统地掌握电子技术的专业英语知识，提升电子技术专业英语能力。

本书可作为高职（含中高职衔接）应用电子技术、电子信息工程、电子通信技术、声像工程技术等专业的 “ 专业英语 ” 课程教材，也可作为 “ 单片机应用技术 ” 课程的双语教材。此外，本书也可作为电子行业工作者在查阅英文技术资料、对外技术交流时的参考用书。

图书在版编目（CIP）数据

电子技术专业英语：汉英对照 / 丁向荣，赖金志，饶瑞福编著. —北京：电子工业出版社，2018.1

ISBN 978-7-121-33279-1

Ⅰ. ①电… Ⅱ. ①丁… ②赖… ③饶… Ⅲ. ①电子技术－英语－高等学校－教材 Ⅳ. ①TN

中国版本图书馆 CIP 数据核字（2017）第 308656 号

责任编辑：郭乃明　　特约编辑：范　丽
印　　刷：
装　　订：
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编　100036
开　　本：787×1 092　1/16　印张：14　字数：359 千字
版　　次：2018 年 1 月第 1 版
印　　次：2018 年 1 月第 1 次印刷
定　　价：36.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）88254561，QQ 34825072。

前　言

在“工业 4.0”、“物联网”、“互联网+”等新概念不断涌现的大背景下，第四次工业革命的进程中，国际交流、国际贸易将更加繁荣，市场急需具有国际视野、国际交流技能的电子技术应用型人才。

传统电子技术专业英语教材通常选择几十篇电子领域各方向的科普性文章，按照生词学习、课文疑难句解析、课文阅读形式进行组织，在内容上稍显枯燥、乏味，组织形式稍显呆板，加上高职学生英语基础水平较差、兴趣不高，使得高职电子技术专业英语普遍教学效果较差，学生不想学，教师也不愿意开课。从教师到学生都明知该课程的重要性与必要性，面对教与学两难的尴尬，该课程成了电子技术类专业课程中的“鸡肋”。

受朱派龙教授编写“机械工程专业英语图解教程”的启发，我们萌生了创新性编写“电子技术专业英语”的想法，围绕朱派龙教授的“望图知义”与“温故而知英”的核心理念，对电子技术专业英语在内容和组织形式上进行大刀阔斧的改革。在教材内容上，设置了电子元器件、电子产品生产工艺、电子产品检验、电子系统设计共四个部分，这四部分内容为电子技术类专业的基础内容，也可以说是主题内容，在电子技术类专业学习占据极其重要的位置；在组织形式上，前三部分采用图解的方式，“望图知义”达到巩固电子元器件、电子产品生产工艺、电子产品检验核心知识的目的，同时掌握好相关英文词汇；“电子系统设计”部分则选用单片机应用技术作为学习内容，系统地以英文形式学习单片机的原理与应用。单片机作为现代电子系统设计的核心，其原理与应用课程在电子技术类专业中是核心课程，学生对单片机有较深刻的认知，对单片机会有较高的学习动机与兴趣，“温故而知英”，既可进一步提升单片机应用能力，又能较系统地掌握电子技术的专业英语知识，提升电子技术专业英语能力。

本教材主要由丁向荣负责规划与统稿，由丁向荣、赖金志、饶瑞福编著，其中第 4 单元至第 9 单元由赖金志、丁向荣共同完成，第 1 单元至第 3 单元由饶瑞福、丁向荣共同完成。由台湾昆山科技大学林俊良教授、STC 创始人姚永平先生担任主审，对全书进行了审阅，并提出了宝贵意见。尤其感谢朱派龙教授，“望图知意”与“温故而知英”是我们新编教材的主体思想。

本书在写作过程中参考了大量书籍，同时也引用了互联网上的资料，在此向这些书籍和资料的原作者表示衷心的感谢。在写作过程中，资料收集和技术交流方面都得到了国内外专业学者和同行的支持，在此向他们也表示衷心的感谢。也可能有些引用资料的出处，基于各种原因未能出现在参考文献中，在此表示歉意与感谢！

由于编者水平有限，书中定有疏漏和不妥之处，敬请读者不吝指正！您有什么建议，可发电子邮件到：dingxiangrong65@163.com，与作者进一步沟通及交流。

编　者

目　　录

PART 1　Electronic Component　电子元器件

PART 2　Manufacturing Process of Electronic Products
电子产品生产工艺

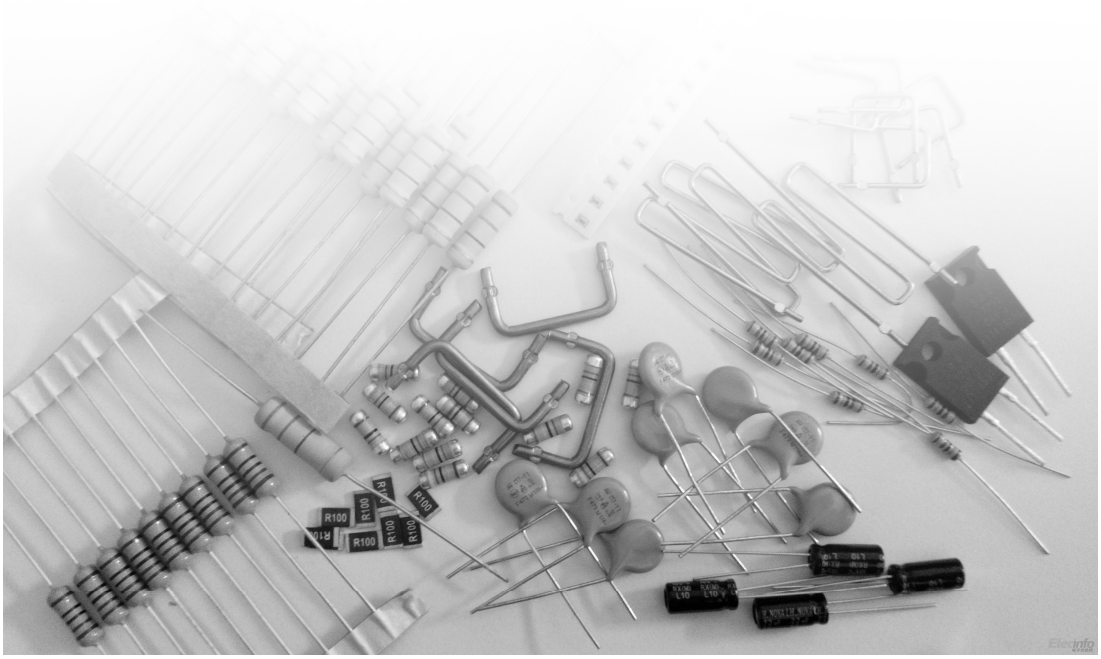
PART 3　Inspection of Electronic Products　电子产品检验

PART 4　Electronic System Design　电子系统设计

PART 1

Electronic Component

电子元器件

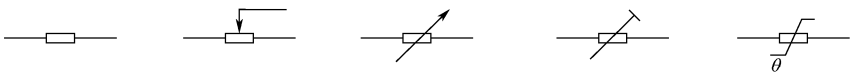


Unit 1　Electronic Component

第 1 单元　电子元器件

1.1　Resistor　电阻

1）Fixed Resistor　固定电阻（Figure 1.1～1.5）



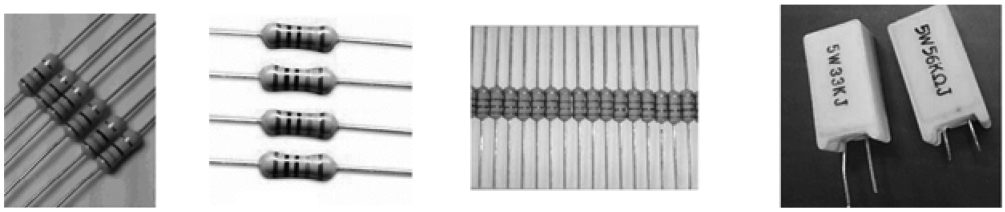
Resistor(symbol) 电阻（一般符号）　Potentiometer 电位器　Adjustable resistor 可调电阻　Trimming potentiometer 微调电位器　Thermistor 热敏电阻

Figure 1.1　Common Circuit Symbols Of Resistors　电阻的常用电路符号

Types of resistors 电阻的种类

- Carbon film resistor　碳膜电阻（RT）
- Metal film resistor　金属膜电阻（RJ）
- Synthetic film resistor　合成膜电阻（WTH）
- Wire-wound resistor　线绕电阻（RX）
- Fuse resistor　熔断电阻
- Cement resistor　水泥电阻
- NTC PTC resistor　NTC、PTC热敏电阻

Figure 1.2　Types of Resistors　电阻的种类



Carbon film resistor 碳膜电阻　Metal film resistor 金属膜电阻　Synthetic film resistor 合成膜电阻　Cement resistor 水泥电阻

Figure 1.3　Common Various Types of Resistors　常用电阻的外形

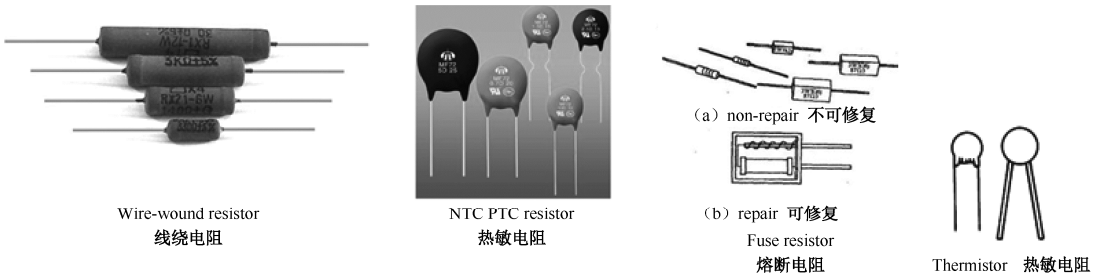


Figure 1.3　Common Various Types of Resistors　常用电阻的外形（续）

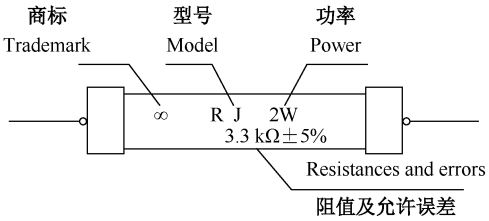
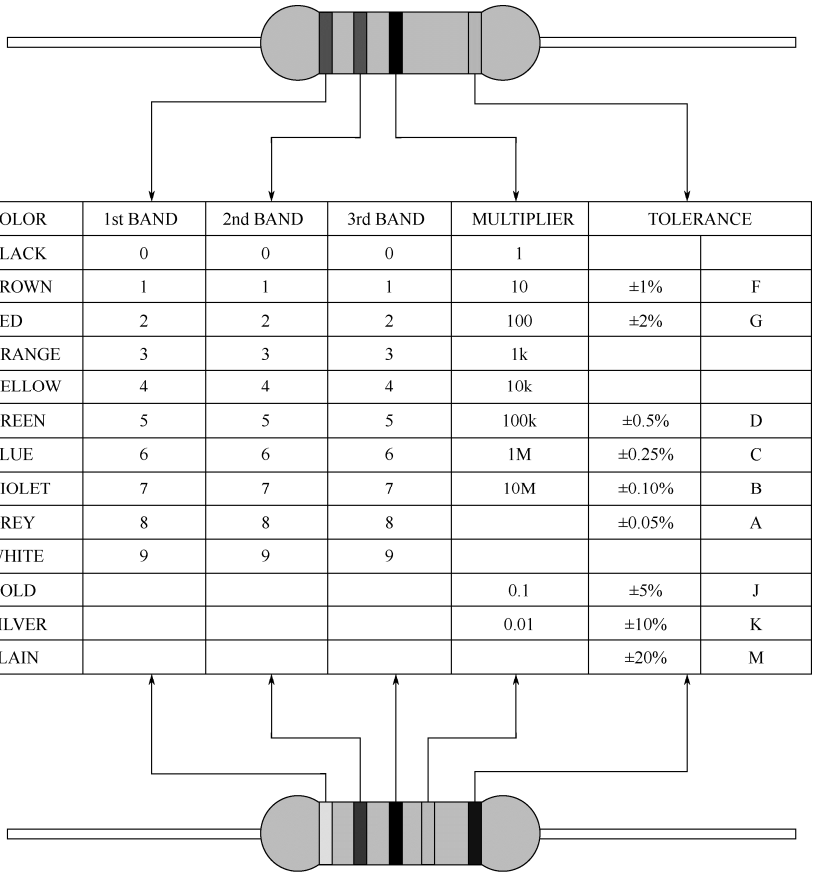


Figure 1.4　The Indication of Resistor　电阻的直标法



COLOR	1st BAND	2nd BAND	3rd BAND	MULTIPLIER	TOLERANCE	
BLACK	0	0	0	1		
BROWN	1	1	1	10	±1%	F
RED	2	2	2	100	±2%	G
ORANGE	3	3	3	1k		
YELLOW	4	4	4	10k		
GREEN	5	5	5	100k	±0.5%	D
BLUE	6	6	6	1M	±0.25%	C
VIOLET	7	7	7	10M	±0.10%	B
GREY	8	8	8		±0.05%	A
WHITE	9	9	9			
GOLD				0.1	±5%	J
SILVER				0.01	±10%	K
PLAIN					±20%	M

Figure 1.5　Recognition of Color Ring of Resistor　色环电阻的识别

2）Adjustable Resistor　可变电阻（Figure 1.6～1.7）

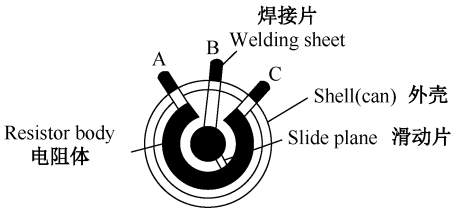
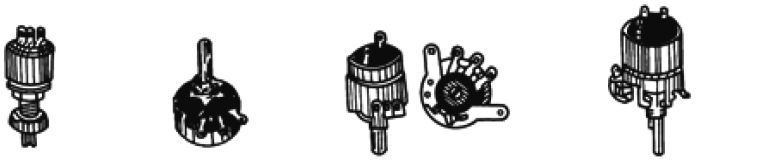
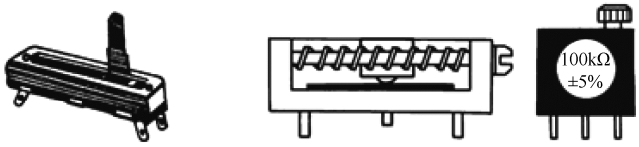


Figure 1.6　Structure of Potentiometer　电位器的结构

Adjustable resistor　可调电阻



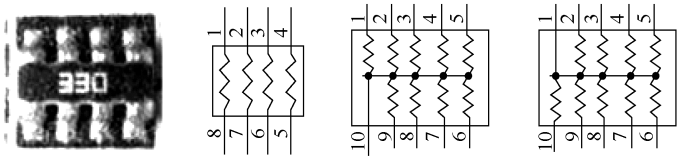
Organic solid potentiometer 有机实心电位器　Carbon film resistor 碳膜电位器　Potentiometer with switch 带开关电位器　Push-pull type potentiometer 推拉式电位器



Straight slide type potentiometer 直滑式电位器　Multi-turn type trimming potentiometer 多圈微调电位器

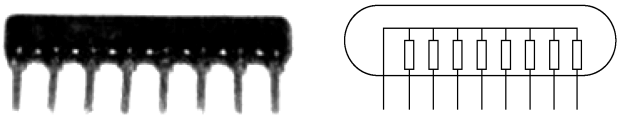
Figure 1.7　Common Various Types of Potentiometers　常见可变电位器外形

3）Line of Resistors (Direct Insertion Resistors)　排电阻（Figure 1.8）



（a）Surface-mount technology (SMT) resistors and structures

贴片排电阻及其结构



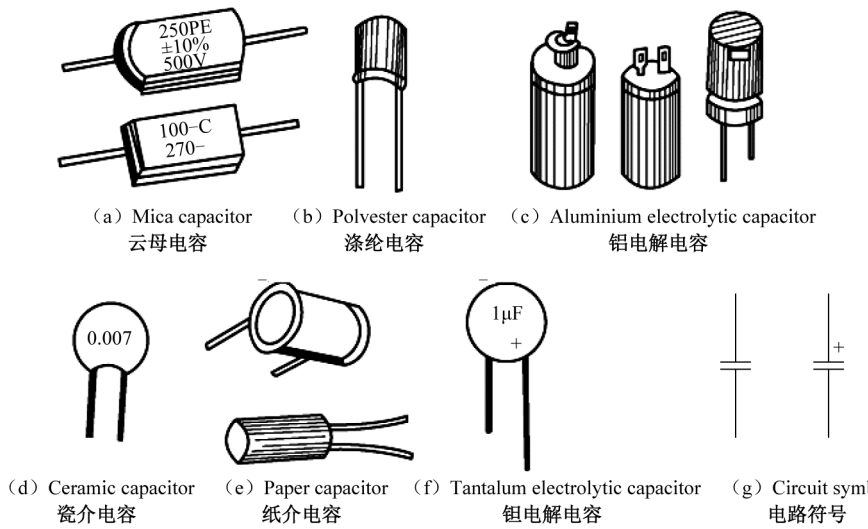
（b）Line of resistors and its structure

直插式排电阻及其结构

Figure 1.8　The Line of Resistors and Structures　排电阻及其内部结构

1.2　Capacitor　电容

1）Fixed Capacitor　固定电容（Figure 1.9，Table 1.1）



（a）Mica capacitor　云母电容
（b）Polyester capacitor　涤纶电容
（c）Aluminium electrolytic capacitor　铝电解电容
（d）Ceramic capacitor　瓷介电容
（e）Paper capacitor　纸介电容
（f）Tantalum electrolytic capacitor　钽电解电容
（g）Circuit symbol　电路符号

Figure 1.9　Common Various Types of Capacitors　常见电容外形

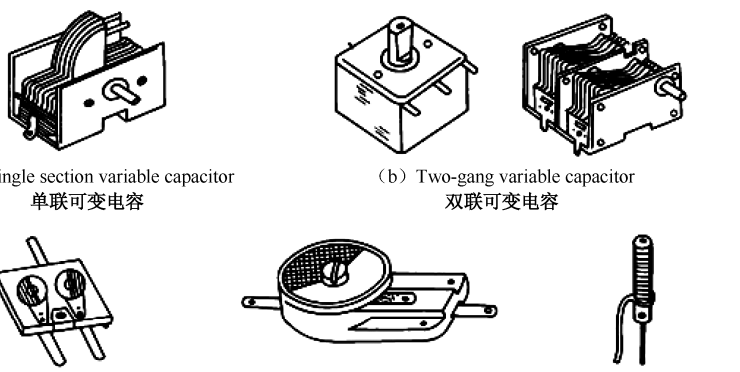
Table 1.1　The Features of Capacitors　常见电容的特点

Name 名称	Feature 特点
Metalized paper capacitor 金属化纸介质电容器（CJ）	Withstanding high voltage (Several tens to one thousand of volts) 耐压高（几十伏~1千伏） Large capacitance 容量大 Self-healing 具有“自愈”能力
Polyester capacitor 涤纶电容器（CL）	Small volume 体积小 Large capacitance 容量大 Good wet-heat resistant 耐热耐湿性好 Small stray inductance 寄生电感小
Mica capacitor 云母电容器（CY）	High precision 精确度高 Heat resistant 耐高温 Corrosion resistant 耐腐蚀 Low loss 介质损耗小 The defect is the low capacitance 缺点是容量较小
Monolithic capacitor 独石电容器	Large capacitance 容量大 Small volume 体积特别小 Heat resistant 耐高温 Good reliability 可靠性好 Low cost 成本低

Continue

Name 名称	Feature 特点
Ceramic capacitor 瓷介电容器 [High frequency 高频（CC） Low frequency 低频（CT）]	Small volume 体积小 Stability 性能稳定 Corrosion resistant 耐腐蚀 Heat resistant 耐热性好 Low loss 损耗小 Good insulation 绝缘电阻高 Used in low loss and high frequency circuit 用于低损耗及高频电路中 The defects are low strength, fragile, and crackable 缺点是机械强度低、易碎易裂
Aluminium electrolytic capacitor 铝电解电容器（CD）	Large capacitance 电容量特别大 Large deviation of capacitance 容量偏差大 Leakage of electricity 漏电 High loss 介质损耗大 Low cost 价格低廉

2）Adjustable Capacitor 可变电容（Figure 1.10）



（a）Single section variable capacitor
单联可变电容

（b）Two-gang variable capacitor
双联可变电容

（c）Organic thin film tuning capacitor
有机薄膜介质微调电容

（d）Fine tuning capacitor
瓷介微调电容

（e）Fine tuning capacitor
拉线微调电容

Figure 1.10 Common Various Types of Adjustable Capacitors 常见可变电容的外形

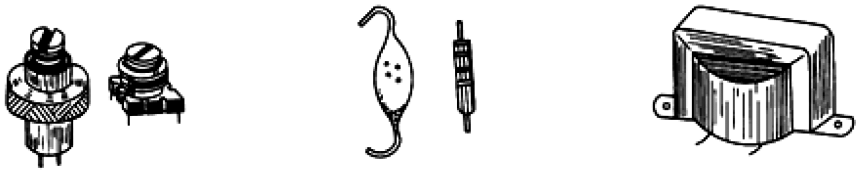
1.3 Inductor 电感

As Shown in Figure 1.11～1.12.

（a）Hollow coil
空心线圈

（b）Core coil
磁芯线圈

Figure 1.11 Common Various Types of Coil of Inductors 常见电感线圈外形

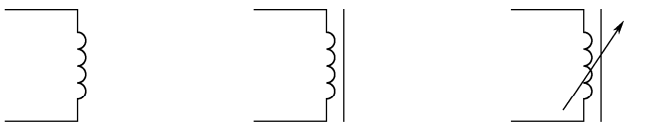


（c）Adjustable core coil
可调磁芯线圈

（d）Color code (color ring) inductor
色码（色环）电感

（e）Iron core
铁芯线圈

Figure 1.11　Common Various Types of Coil of Inductors　常见电感线圈外形（续）



（a）Symbol
一般符号

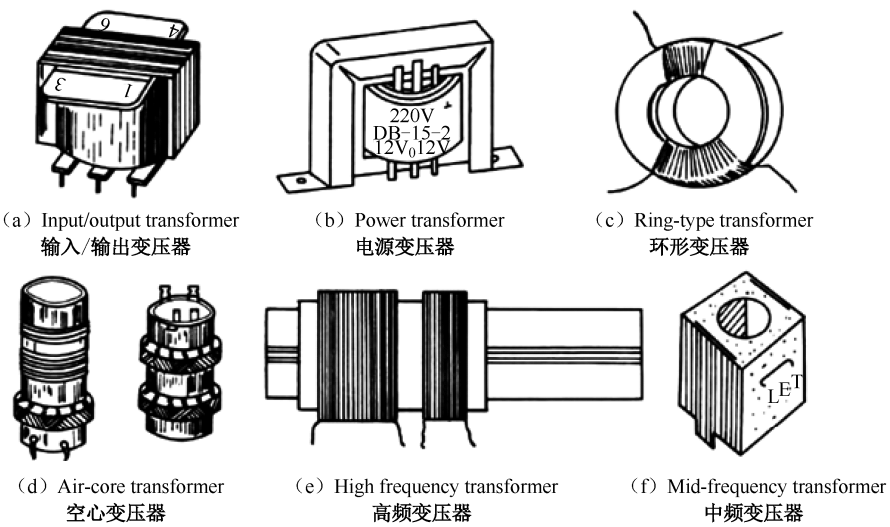
（b）Core (iron) coil inductor
带磁（铁）芯电感

（c）Adjustable inductor
可调电感

Figure 1.12　Common Circuit Symbols of Coil Inductors　电感线圈的电路符号

1.4　Transformer　变压器

As Shown in Figure 1.13～1.14.



（a）Input/output transformer
输入/输出变压器

（b）Power transformer
电源变压器

（c）Ring-type transformer
环形变压器

（d）Air-core transformer
空心变压器

（e）High frequency transformer
高频变压器

（f）Mid-frequency transformer
中频变压器

Figure 1.13　Common Various Types of Transformer　常用变压器外形

（a）Transformer with center cap
带中心抽头

（b）Adjustable core coil
磁芯可调

（c）Same polarity
带同名端

（d）Shielding
带有屏蔽

Figure 1.14　Common Circuit Symbols of Transformer　常用变压器符号

1.5 Relay 电磁继电器

As Shown in Figure 1.15～1.16.

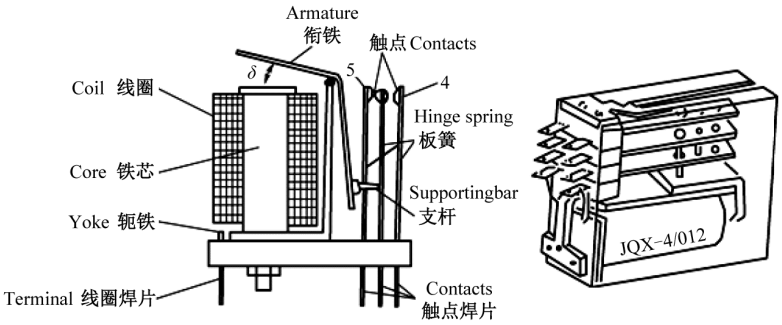


Figure 1.15 Structure of Relay 电磁继电器结构

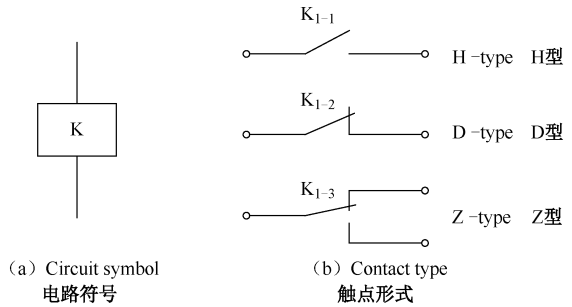


Figure 1.16 Common Circuit Symbols of Relays 电磁继电器的常用电路符号

1.6 Diode 半导体二极管

As Shown in Figure 1.17～1.18.

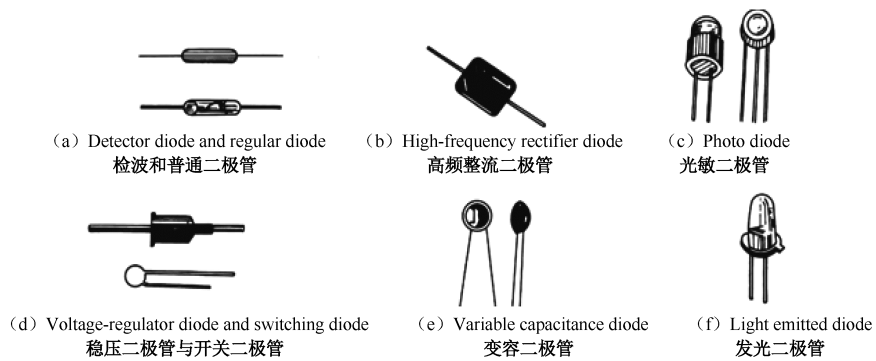
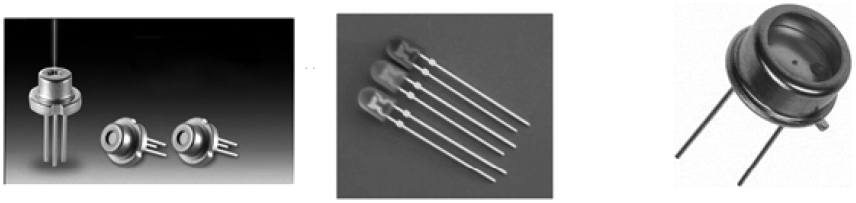


Figure 1.17 Common Various Types of Diodes 常见半导体二极管的外形

（a）Detector diode
检波二极管

（b）Rectifier diode
整流二极管

（c）Voltage regulator diode
稳压二极管



（d）Laser diode
激光二极管

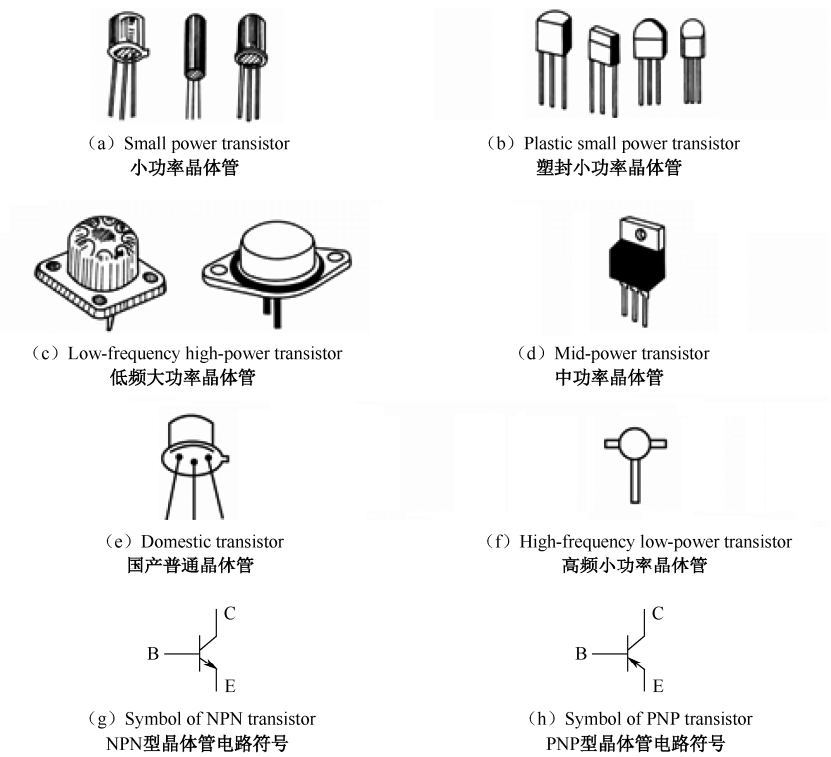
（e）Light emitted diode
发光二极管

（f）Photo diode
光敏二极管

Figure 1.18　Some Common Diode　常见二极管实物图

1.7　Transistor　半导体晶体管

As Shown in Figure 1.19～1.20.



（a）Small power transistor
小功率晶体管

（b）Plastic small power transistor
塑封小功率晶体管

（c）Low-frequency high-power transistor
低频大功率晶体管

（d）Mid-power transistor
中功率晶体管

（e）Domestic transistor
国产普通晶体管

（f）High-frequency low-power transistor
高频小功率晶体管

（g）Symbol of NPN transistor
NPN型晶体管电路符号

（h）Symbol of PNP transistor
PNP型晶体管电路符号

Figure 1.19　The Types of Bipolar Transistor and Circuit Symbol　双极型晶体管的外形与电路符号

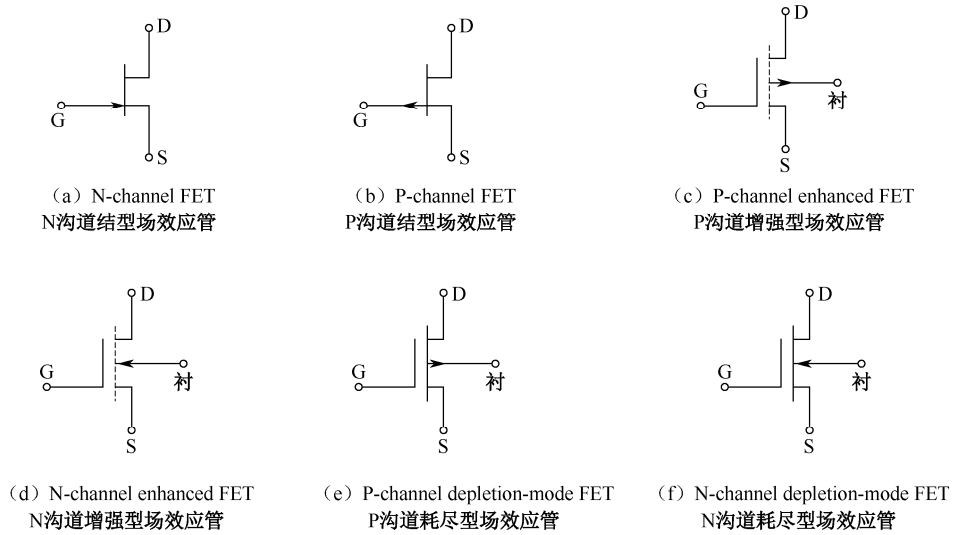


Figure 1.20　Circuit Symbols of Field Effect Transistor　场效应管电路符号

1.8　Thyristor　晶闸管

As Shown in Figure 1.21～1.22.

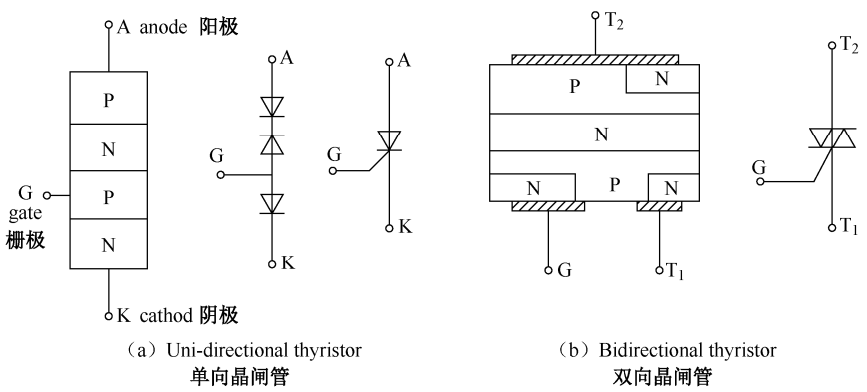


Figure 1.21　Structures of Thyristors and Circuit Symbols　晶闸管的结构与电路符号

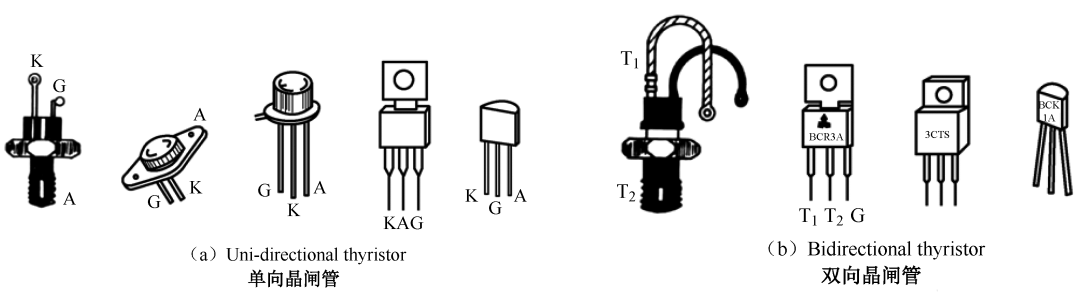


Figure 1.22　Common Various Types of Thyristors　常见晶闸管的外形

1.9　Integrated Circuit　集成电路

As Shown in Figure 1.23～1.25，Table 1.2.

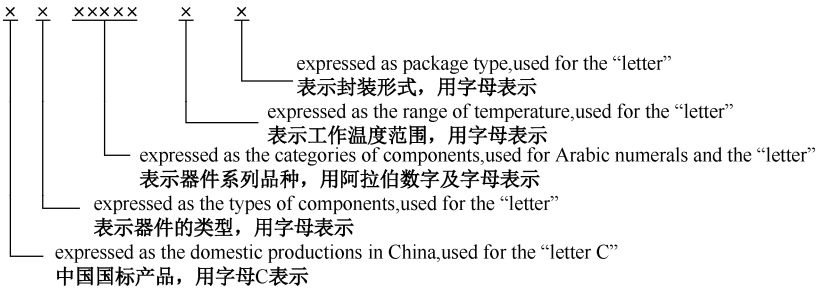


Figure 1.23　International Nomenclatures of IC　国标集成电路型号的命名方法

Table1.2　List of The Types of Integrated Circuits　国标集成电路型号命名的含义

Part 1: Country name 第一部分：国标		Part 2: Circuit type 第二部分：电路类型		Part 3: Circuit and code 第三部分：电路系列和代号	Part 4: Range of temperature 第四部分：温度范围		Part 5: Packaging type 第五部分：封装类型	
Letter 字母	Country 国别	Letter 字母	Circuit 电路（名称）		Letter 字母	Temperature （使用）温度	Letter 字母	Package 封装
lC	Made in China 中国制造	B	Nonlinear circuit 非线性电路	Code or mixing of numbers and letters 用数字或数字与字母混合表示电路的系列与代号	C	0～70℃	B	Plastic flat package 塑料扁平
		C	CMOS circuit CMOS 电路				C	Leadless ceramic chip carrier 陶瓷芯片载体封装
		D	Audio and TV circuit 音响、电视电路		G	-25～70℃	D	Ceramic dual in-line package 多层陶瓷双列直插
		E	ECL circuit ECL 电路				E	Plastic leaded chip carrier 塑料芯片载体封装
		F	Linear circuit 线性电路					
		H	HTL circuit HTL 电路					
		J	Interface circuit 接口电路		L	-25～85℃	F	Ceramic flat package 多层陶瓷扁平
		M	Memory 存储器				G	Pin grid array 网络阵列封装
		W	Voltage stabilizer 稳压器					
		T	TTL circuit TTL 电路		E	-40～85℃	H	Black ceramic package 黑瓷扁平

Continue

Part 1: Country name 第一部分：国标		Part 2: Circuit type 第二部分：电路类型		Part 3: Circuit and code 第三部分：电路系列和代号	Part 4: Range of temperature 第四部分：温度范围		Part 5: Packaging type 第五部分：封装类型	
Letter 字母	Country 国别	Letter 字母	Circuit 电路（名称）		Letter 字母	Temperature （使用）温度	Letter 字母	Package 封装
IC	Made in China 中国制造	μ	Microprocessor 微处理器电路	Code or mixing of numbers and letters 用数字或数字与字母混合表示电路的系列与代号			J	Black ceramic dual in-line package 黑瓷双列直插封装
		AD	A/D conversion circuit A/D 转换器		R	-55～85℃	K	Metal diamond-type 金属菱形封装
		DA	D/A conversion circuit D/A 转换器				P	Plastic dual in-line package 塑料双列直插封装
		SC	Communication circuit 通信专用电路		M	-55～125℃	S	Plastic single in-line package 塑料单列直插封装
		SS	Sensitive circuit 敏感电路				T	TO-8 金属圆形封装
		SW	Clock circuit 钟表电路					

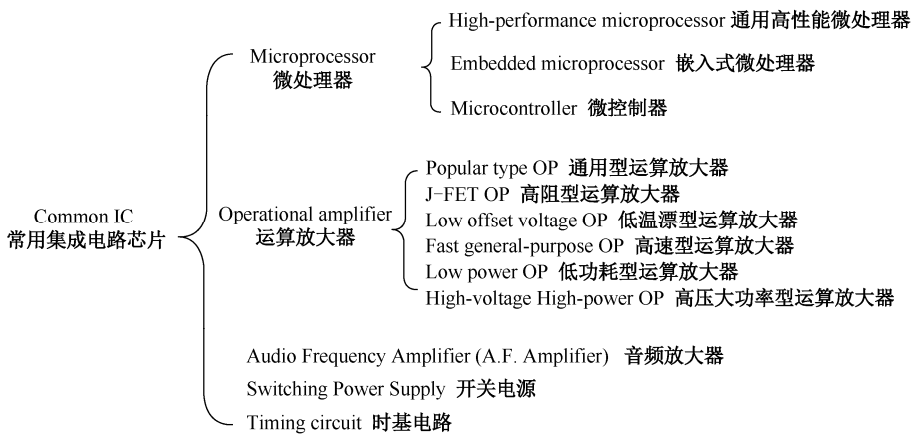
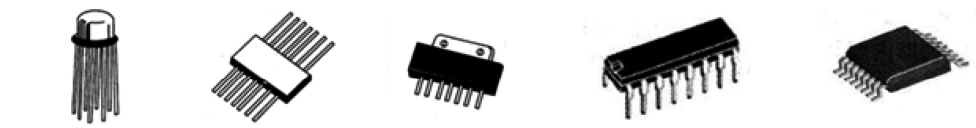


Figure 1.24 Common IC 常用集成电路芯片



（a）TO-8金属壳圆形 （b）Flat扁平形 （c）SIP （d）DIP （e）SOP

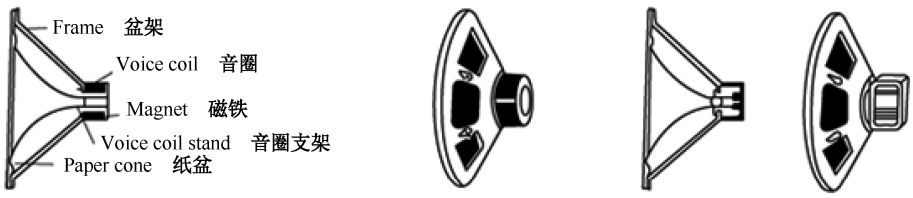
Figure 1.25 Common Various Types of Packages of ICs IC 的常用封装类型

（f）SOJ　（g）QFP　（h）PLCC　（i）BGA

Figure 1.25　Common Various Types of Packages of ICs　IC 的常用封装类型（续）

1.10　Electroacoustic Device　电声器件

1）Loudspeaker　扬声器（Figure 1.26～1.30）



（a）Outer magnet type　外磁式　（b）Inner magnet type　内磁式

Figure 1.26　Dynamic Loudspeaker Magnet Type　电动式扬声器

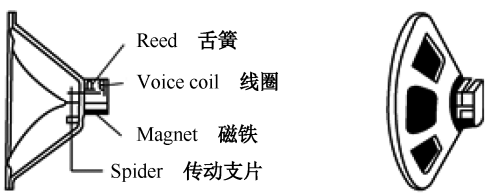


Figure 1.27　Reed-type Loudspeaker　舌簧式扬声器

Figure 1.28　Crystal Loudspeaker　晶体式扬声器

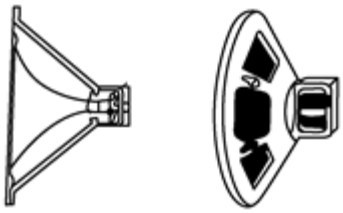


Figure 1.29 Excitation Type Loudspeaker
励磁式扬声器

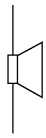
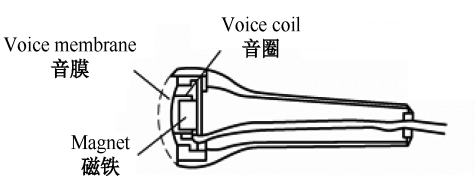
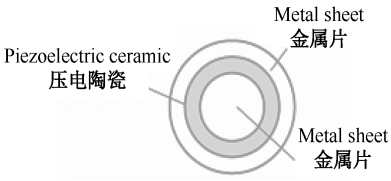


Figure 1.30 Loudspeaker Circuit Symbol
扬声器电路符号

2）Microphone 传声器（Figure 1.31）



（a）Moving coil microphone 动圈式传声器



（b）Piezoelectric ceramic 压电陶瓷片

（c）Electret condenser microphone 驻极体电容式传声器

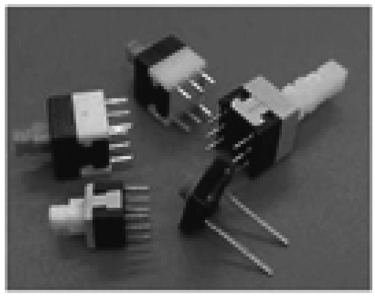
（d）Microphone circuit symbol 传声器的电路符号

Figure 1.31 Common Various Types of Microphones and Circuit Symbols 常见传声器的外形与电路符号

1.11 Switch, Connector, and Fuse
开关件、接插件与熔断器

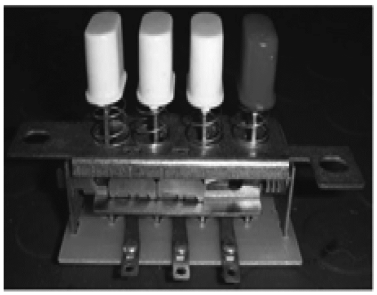
1）Switch 开关件（Figure 1.32）

（a）Toggle switch 钮子开关



（b）Button switch 按键开关

Figure 1.32 Common Various Types of Switches 常见开关的外形



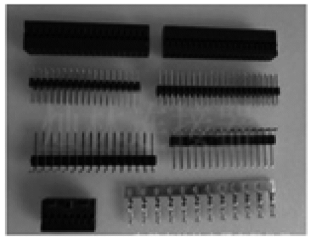
（c）Key switch　琴键开关

（d）Slide switch　滑动开关

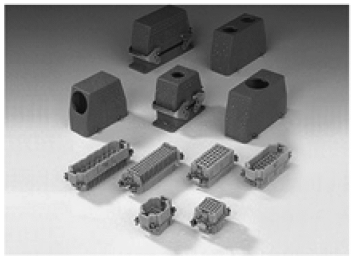
Figure 1.32　Common Various Types of Switches　常见开关的外形（续）

2）Connector　接插件（Figure 1.33）

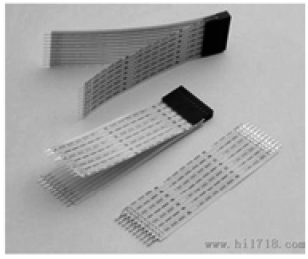
（a）Circular connector　圆形接插件



（b）Dupont series pin header　杜邦系列插针



（c）Rectangular connector　矩形接插件



（d）Flexible flat cable　扁平软排线

Figure 1.33　Common Various Types of Connectors　常见接插件的外形

3）Fuse　熔断器（Figure 1.34）

（a）Glass tube fuse　玻璃式熔断器

（b）Low voltage ceramic plug-in type fuse
插入式陶瓷低压熔断器

Figure 1.34　Common Various Types of Fuses　常见熔断器的外形

（c）Screw type fuse **螺旋式熔断器**

（d）High voltage fuse **高压熔断器**

Figure 1.34　Common Various Types of Fuses　常见熔断器的外形（续）

1.12　Surface Mounted Component　表面安装元器件

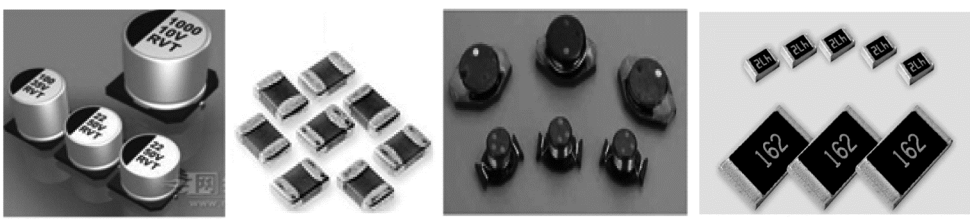
As Shown in Table 1.3，Figure 1.35.

Table1.3　Classification of Surface Mounted Components　表面安装元器件的分类

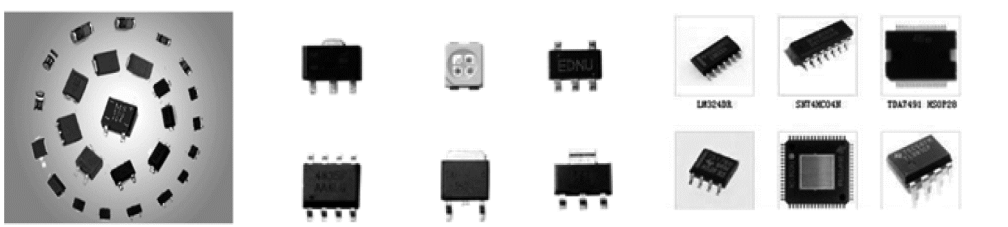
Category 种　类		Rectangular type 矩　形	Cylindrical type 圆柱形
Passive component 片式无源元件	Chip resistor 片式电阻器	Thick film resistor 厚膜电阻器 Thin film resistor 薄膜电阻器 Thermistor 热敏电阻器	Carbon film resistor 碳膜电阻器 Metal film resistor 金属膜电阻器
	Chip capacitor 片式电容器	Monolithic ceramic capacitor 陶瓷独石电容器 Film capacitor 薄膜电容器 Mica capacitor 云母电容器 Trimmer capacitor 微调电容器 Electrolytic capacitor 铝电解电容器 Tantalum electrolytic capacitor 钽电解电容器	Ceramic capacitor 陶瓷电容器 Solid tantalum electrolytic capacitor 固体钽电解电容器
	Chip potentiometer Patch-potentiometer 片式电位器	Potentiometer 电位器 Trimmer potentiometer 微调电位器	
	Chip inductor 片式电感器	Wound inductor 绕线电感器 Stacked inductor 叠层电感器 Variable inductor 可变电感器	Wound inductor 绕线电感器
	Chip sensitive component 片式敏感元件	Voltage dependent resistor (varistor) 压敏电阻器 Thermistor (thermal resistor) 热敏电阻器	
	Chip composite component 片式复合元件	Resistance network 电阻网络 Filter 滤波器 Resonator 谐振器 Ceramic capacitance network 陶瓷电容网络	

Continue

Category 种　类		Rectangular type 矩　形	Cylindrical type 圆柱形
Active component 片式有源器件	Small package diode 小型封装二极管	Plastic stabilizer 塑封稳压 Rectifier 整流 Switch 开关 Zener diode 齐纳二极管 Varactor diode 变容二极管	Glass sealing stabilize 玻封稳压 Rectifier 整流 Switch 开关 Zener diode 齐纳二极管 Varactor diode 变容二极管
	Small package transistor 小型封装晶体管	Plastic PNP 塑封 PNP NPN type transistor NPN 晶体管 Plastic FET 塑封场效应管	
	Miniature integrated circuit 小型集成电路	Flat package 扁平封装 Chip carrier 芯片载体	
	Bare chip 裸芯片	Package with guard ring 带环载体 Flip-chip 倒装芯片	



（a）Chip capacitor 贴片电容　（b）Chip inductor 贴片电感　（c）Chip resistor 贴片电阻



（d）Chip diode 贴片二极管　（e）Chip transistor 贴片三极管　（f）Chip integrated circuit 贴片集成电路

Figure 1.35　Common Various Types of Chip Components　常见贴片元件的外形

Exercise　练习题

1. To provide the query data describes the roles of resistors in the circuits.
2. To provide the query data describes the roles of capacitors in the circuits.
3. To provide the query data describes the roles of inductors in the circuits.
4. To provide the query data describes the features of diode unidirectional conductions.
5. To provide the query data describes the features of amplification of the transistors.
6. Summary of the main types of integrated circuit package.

PART 2

Manufacturing Process of Electronic Products

电子产品生产工艺

Unit 2　Manufacturing Process of Electronic Products

第 2 单元　电子产品生产工艺

2.1　Assembly Process　整机装配工艺

1. Manufacture Procedure of Assembly Process　整机装配工艺流程（Figure 2.1）

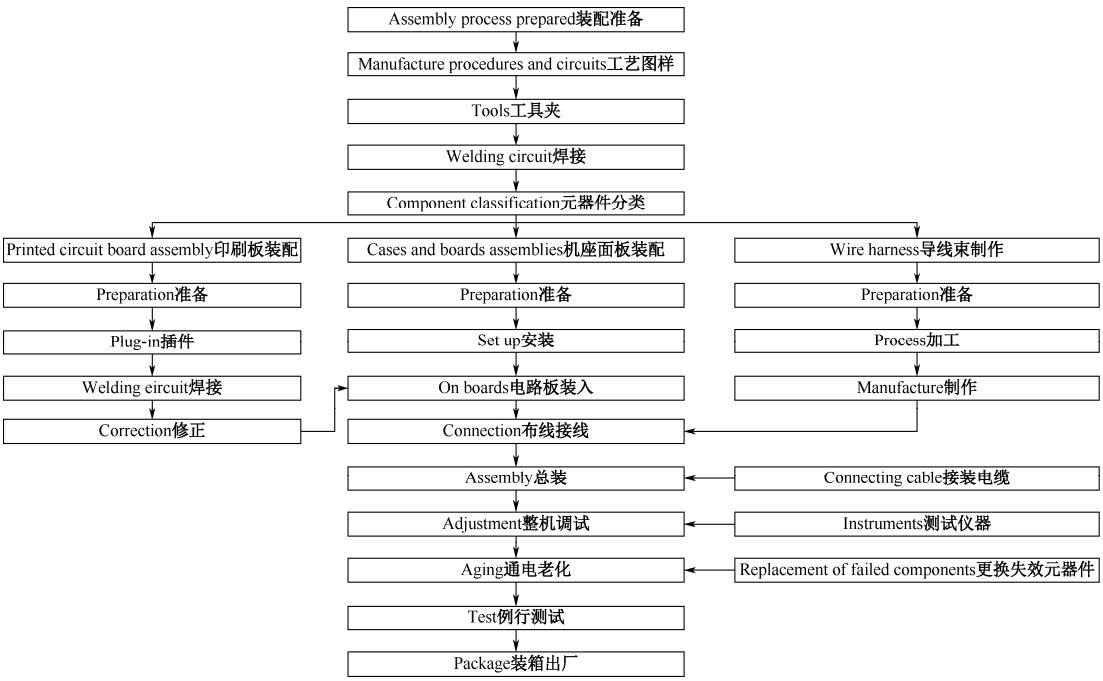


Figure 2.1　Manufacture Procedure of Assembly Process　整机装配工艺流程

2. Sorting The Complete Machine　整机装配顺序（Figure 2.2）

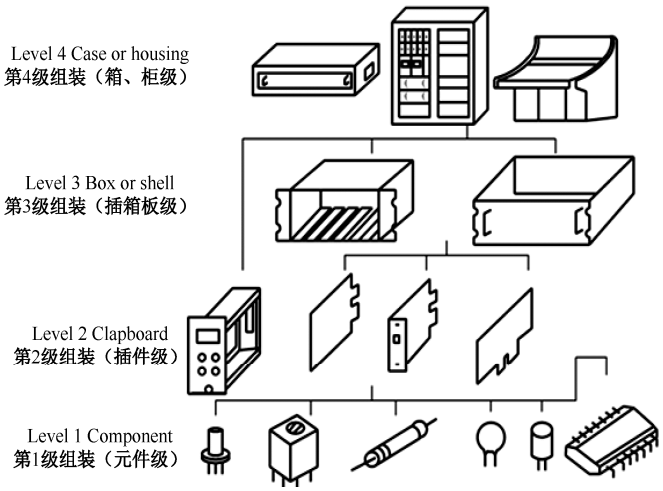


Figure 2.2　Sorting The Complete Machine　整机装配顺序

3. Assembly Process Graphics　整机装配工程图

1）Diagram of Circuit Block　电路框图（Figure 2.3）

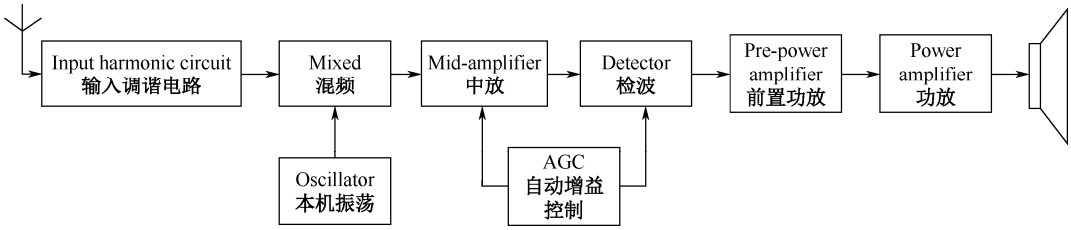


Figure 2.3　Diagram of Super-heterodyne Radio　超外差式收音机框图

2）Schematic Circuit　电路原理图（Figure 2.4）

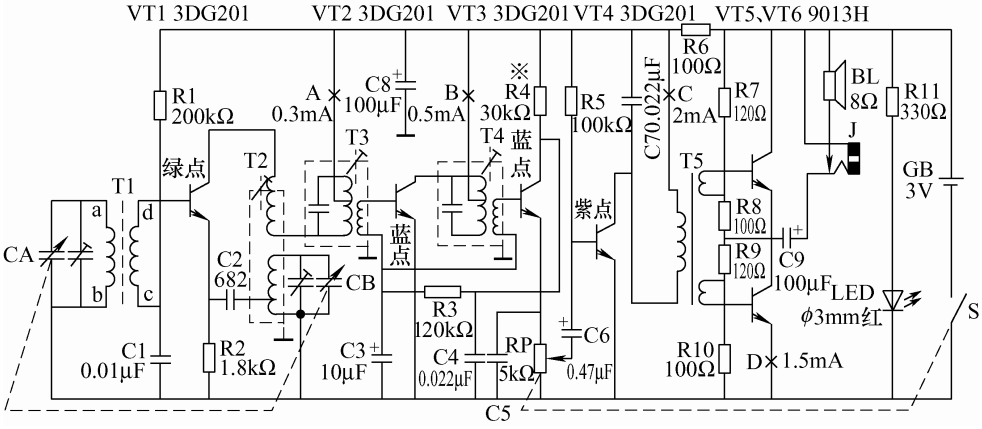
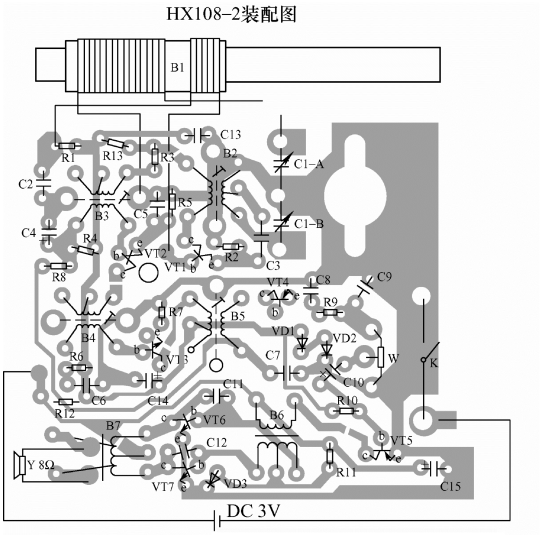


Figure 2.4　The Schematic Circuit of Super-heterodyne Radio　超外差收音机原理图

3）PCB Lay Out 印制电路板装配图（Figure 2.5）



9018H β97~146
9013H β144~202

Color ring
色环电阻色标数

1	2	3	4	5	6
brown 棕	red 红	orange 橙	yellow 黄	green 绿	blue 蓝
7	8	9	0	±5%	±10%
purple 紫	gray 灰	white 白	black 黑	golden 金	silver 银

例：棕 黑 橙 金

表示 1　0　10^3 偏差:±5%
该电阻为$10\times10^3\times(1\pm5\%)\Omega$
即为$10\times(1\pm5\%)k\Omega$

List of components元器件位号目录						List of components结构件清单		
Numbers 位号	Specifications 名称规格		Numbers 位号	Specifications 名称规格		序号	Specifications 名称规格	数量
R1	Resistance 电阻	100kΩ	C11	Chip capacitor 瓷片电容	0.022μF	1	Front frame前框	1
R2		2kΩ	C12		0.022μF	2	Cover后盖	1
R3		1000kΩ	C13		0.022μF	3	Calibration board周率板	1
R4		20kΩ	W	Potentiometer 电位器	5Ω	4	Tuning disc调谱盘	1
R5		150kΩ	C1	Double双联	CBM223P	5	Electric potential disc 电位盘	1
R6		62kΩ	B1	Magnetic bar 磁棒	B,5mm*13mm*55mm	6	Magnetic bar bracket 磁棒支架	1
R7		51kΩ		Antenna coil 天线线圈		7	Printing circuit board印制板	1
R8		1kΩ	B2	Oscillating coil 振荡线圈	Red红	8	Positive film正极板	2
R9		680kΩ	B3	中周	Yellow黄	9	Negative spring负极簧	2
R10		51kΩ	B4		White白	10	拎带	1
R11		1kΩ	B5		Black黑	11	Adjusting screw 调节盘螺钉	
R12		220kΩ	B6	Input Transformer (Blue Green)	输入变压器（蓝、绿）		沉头M2.5*4	1
R13		24kΩ	B7	Output Transformer (Yellow Red)	输出变压器（黄、红）	12	双连螺钉	
C2	Chip capacitor 瓷片电容	0.022μF	VD1	Diode二极管	1N4148		M2.5*5	2
C3		0.01μF	VD2		1N4148	13	机芯自攻螺钉	
C5		0.22μF	VD3		1N4148		M2.5*6	1
C6		0.022μF	VT1	Transistor 三极管	9018H	14	Potentiometer screw 电位器螺钉	
C7		0.022μF	VT2		9018H		M1.7*4	1
C8		0.022μF	VT3		9018H	15	正极导线（9cm）	1
C9		0.022μF	VT4		9018H	16	负极导线（10cm）	1
C4	Electrical capacitor 电解电容	4.7μF	VT5		9014C	17	Speaker wire 扬声器导线	2
C10		4.7μF	VT6		9013H			
C14		100μF	VT7		9013H			
C15		100μF	Y	Speaker扬声器	$2\frac{1}{2}$,8Ω			

Figure 2.5　PCB Lay Out　印制电路板装配图

4）Circuit Lay-out 电路接线图（Figure 2.6～2.7）

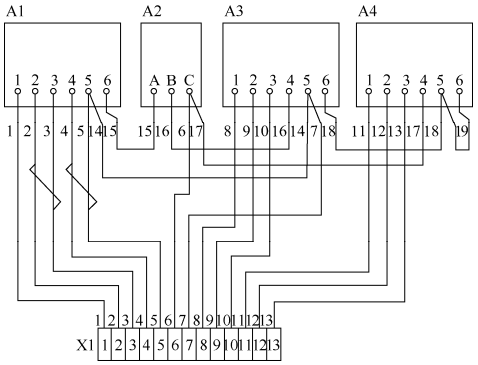


Figure 2.6　Circuit Lay-out　电路接线图

Cable number(s) 线缆号	Wire number 线号	Connecting point1 连接点 1		Connecting point2 连接点 2		Cable number(s) and specification (s) 导线电缆型号及规格	Length/mm 长度/mm	Note(s) 备注
		Part code 项目代号	Terminal code 端子代号	Part code 项目代号	Terminal code 端子代号			
	1	X1	1	A1	1	Installation line 安装线 VAR0.5mm² YE	200	绞合
	2	X1	2	A1	2	VAR0.5mm² RD	200	绞合
	3	X1	3	A1	3	VAR0.5mm² BU	200	绞合
	4	X1	4	A1	4	VAR0.5mm² GN	200	绞合
	5	X1	5	A1	5	VAR0.5mm² BN	200	
	6	X1	6	A2	C	VAR0.5mm² BK	250	
	7	X1	7	A3	5	VAR0.5mm² BN	300	
	8	X1	8	A3	1	VAR0.5mm² GY	300	
	9	X1	9	A3	2	VAR0.5mm² WH	300	
	10	X1	10	A3	3	VAR0.5mm² RD	300	
	11	X1	11	A4	1	VAR0.5mm² GN	220	
	12	X1	12	A4	2	VAR0.5mm² YE	220	
	13	X1	13	A4	3	VAR0.5mm² RD	220	
	14	A1	5	A3	5	VAR0.5mm² BN	300	
	15	A1	6	A2	A	VAR0.5mm² WH	100	
	16	A2	B	A3	4	VAR0.5mm² GY	200	
	17	A2	C	A4	4	VAR0.5mm² BK	200	
	18	A3	6	A4	5	VAR0.5mm² BU	150	
	19	A4	5	A4	6	Installation line 安装线 VAR0.5mm² BU	20	

Serial number(s) 序号	Code(s) 代号	Name(s) 名称	Number(s) 数量	Note(s) 备注
1	SJ2086-82	Installation line 安装线 VAR0.5mm2 YE	420	
2	SJ2086-82	VAR0.5mm2 RD	720	
3	SJ2086-82	VAR0.5mm2 BU	370	
4	SJ2086-82	VAR0.5mm2 GN	420	
5	SJ2086-82	VAR0.5mm2 BN	800	
6	SJ2086-82	VAR0.5mm2 BK	450	
7	SJ2086-82	VAR0.5mm2 WH	400	
8	SJ2086-82	VAR0.5mm2 GY	500	

××装置
单元接线图（表）

第1张　　共1张

Figure 2.7　Installation Line　接线

5）Making Wiring-drawing 线扎图（Figure 2.8～2.9）

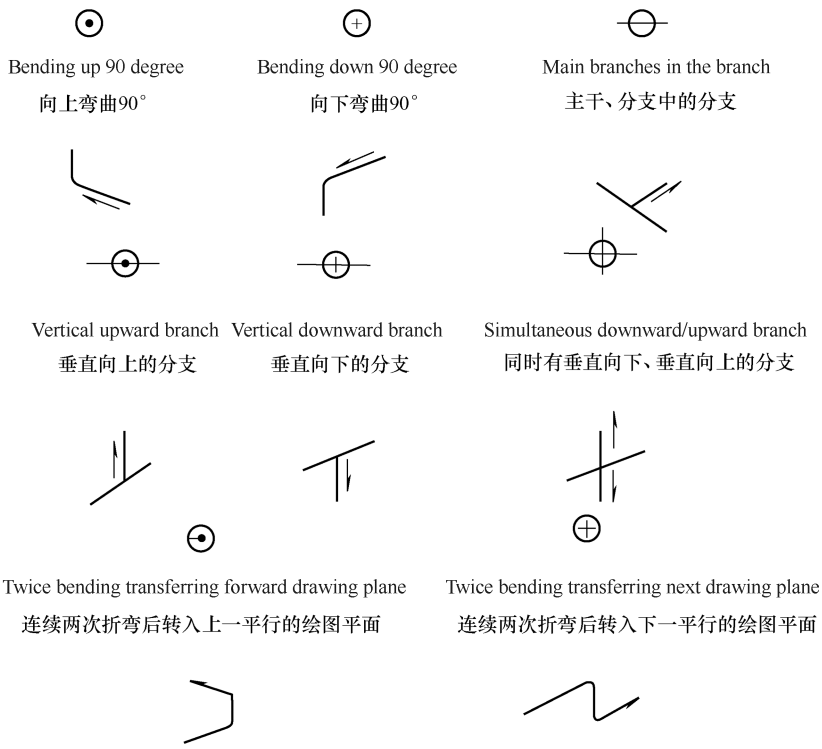


Figure 2.8　Wiring-drawing Symbol　接线扎符号

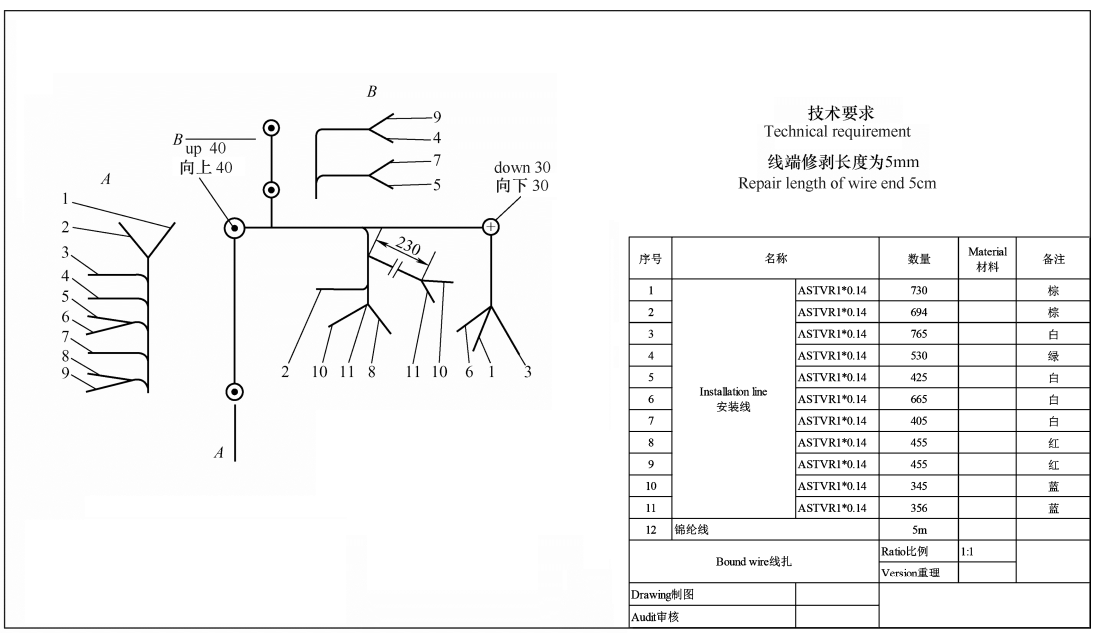


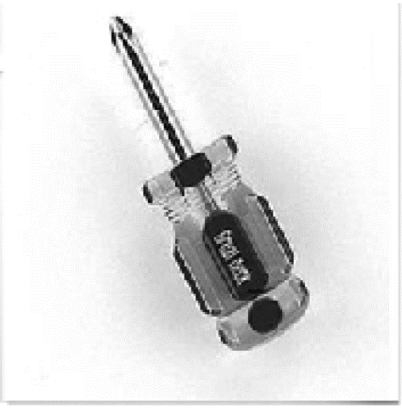
Figure 2.9　Making Wiring-drawing　线扎图

2.2 Common Tools for Electrical Production
电子生产常用工具

1. Hardware Tool 五金工具（Figure 2.10～2.22）

1）Screw Driver 螺丝刀

Slotted screw driver 一字螺丝刀



Phillips screw driver 十字螺丝刀

Hand pressing plate automatic 手压式板自动（螺丝刀）

Electric screw driver 电动螺丝刀

Figure 2.10 Categories of Screw Driver 螺丝刀的种类

2）Needle-nose Plier　尖嘴钳

Figure 2.11　Needle-nose Plier　尖嘴钳

3）Diagonal Plier　斜口钳

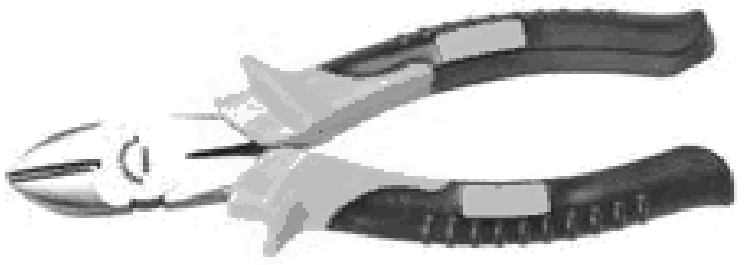


Figure 2.12　Diagonal Plier　斜口钳

4）Plier/Vice　钢丝钳

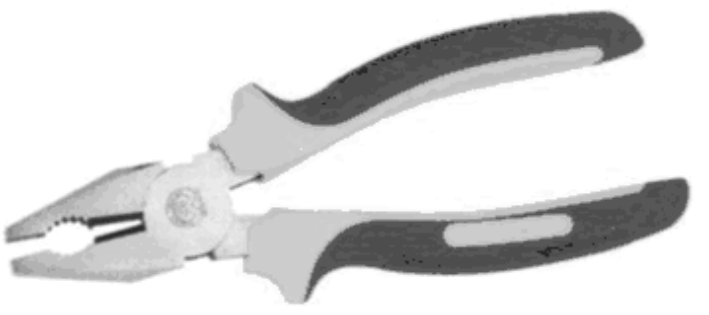


Figure 2.13　Plier/Vice　钢丝钳

5）Tweezer　镊子

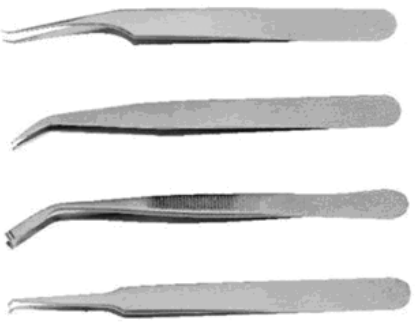


Figure 2.14　Tweezer　镊子

6）Scissor　剪刀

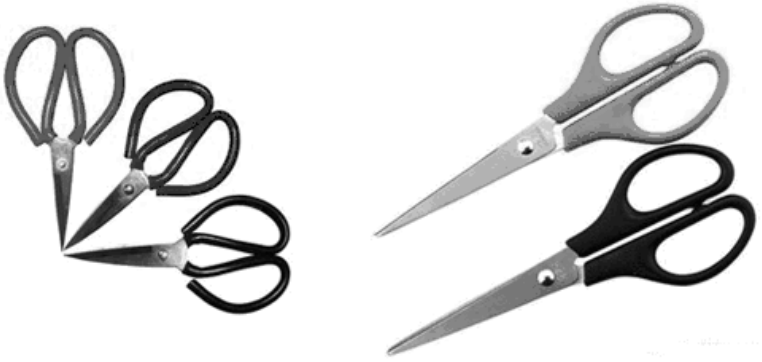
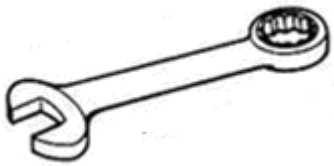


Figure 2.15　Scissor　剪刀

7）Wrench Set　扳手

Double Open End Wrench　扳手

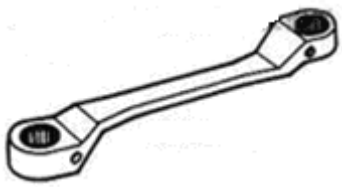
C-wrench　钩形扳手



Conbination Wrench　两用扳手

Socket Wrench　套筒扳手

Figure 2.16　Wrench Set　扳手



Double Spline Wrench　梅花扳手

Hex Wrench　内六角扳手

Adjustable Wrench　活动扳手

Torque Wrench　扭力扳手

Figure 2.16　Wrench Set　扳手（续）

8）File　整形锉

Figure 2.17　File　整形锉

9）Screwdriver with Molded Handle　无感螺丝刀

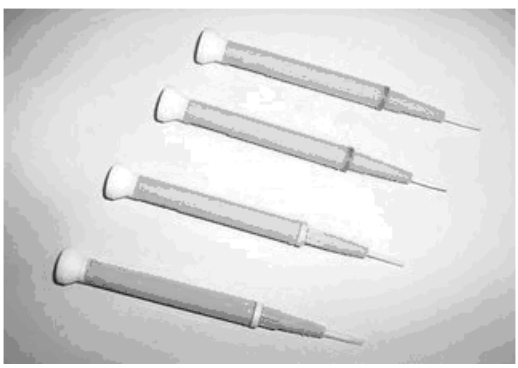


Figure 2.18　Screwdriver with Molded Handle　无感螺丝刀

10）Driver　钟表起子

Figure 2.19　Driver　钟表起子

11）Wire Stripper　剥线钳

Figure 2.20　Wire Stripper　剥线钳

12）Crimping Pincher　压接钳

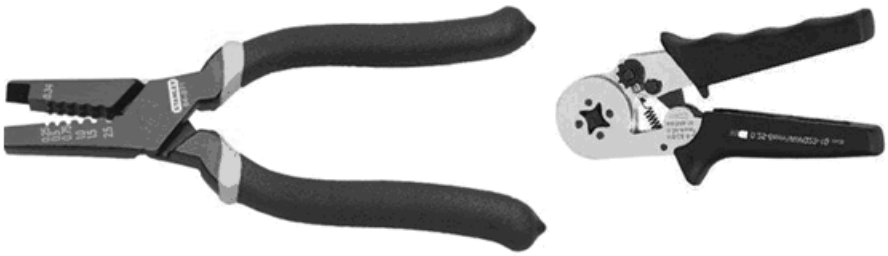


Figure 2.21　Crimping Pincher　压接钳

13）Hot Melt Glue Gun　热熔胶枪

Figure 2.22　Hot Melt Glue Gun　热熔胶枪

2. Welding tool　焊接工具（Figure 2.23～2.29）

1）Electric Soldering Iron　电烙铁

External heating electric soldering iron　外热式电烙铁

Internal heating electric soldering iron
内热式电烙铁

Figure 2.23　Electric Soldering Iron　电烙铁

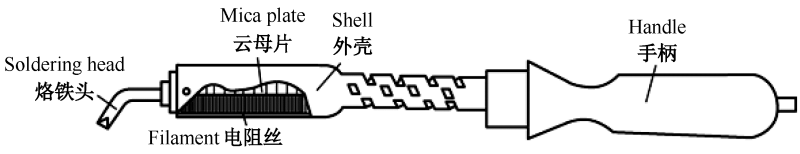


Figure 2.24　Structure of External Heating Electric Soldering Iron　外热式电烙铁结构

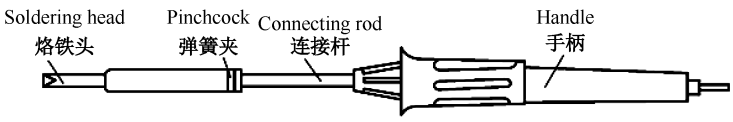


Figure 2.25　Structure of Internal Heating Electric Soldering Iron　内热式电烙铁结构

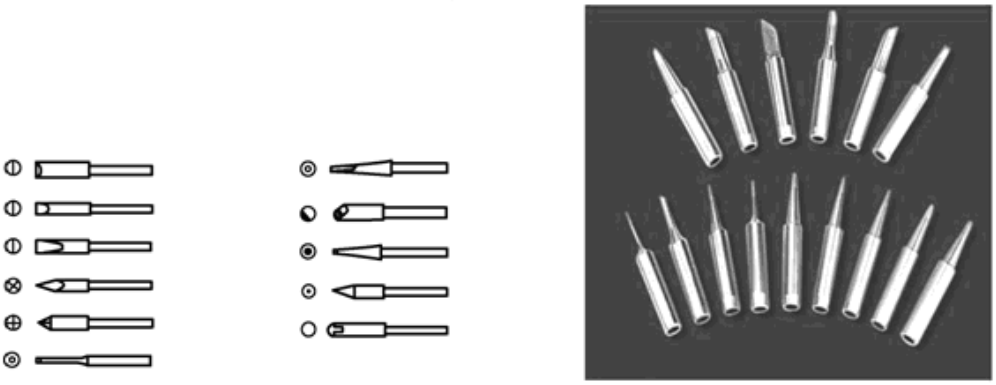


Figure 2.26　Common Solder Soldering Iron Tip　常用烙铁头

2）Temperature-controlled Soldering Iron　恒温电烙铁

Figure 2.27　Temperature-controlled Soldering Iron　恒温电烙铁

3）Automatic Tin Feeding Machine　自动送锡电烙铁

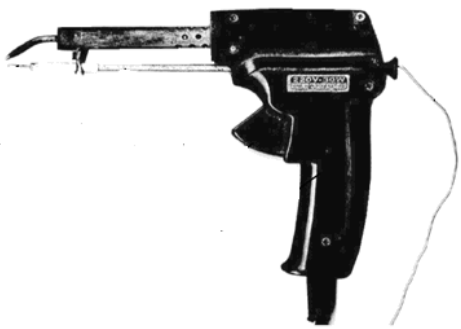


Figure 2.28　Automatic Tin Feeding Machine　自动送锡电烙铁

4）Solder Sucker　吸锡器

Figure 2.29　Solder Sucker　吸锡器

2. Hot Air Gun　电热风枪（Figure 2.30）

Figure 2.30　Hot Air Gun　电热风枪

3. Common Equipment　常用专用设备（Figure 2.31～2.37）

1）Automatic Wire Stripping Machine　自动剥线机

Figure 2.31　Automatic Wire Stripping Machine　自动剥线机

2）Automatic Aircraft　自动切脚机

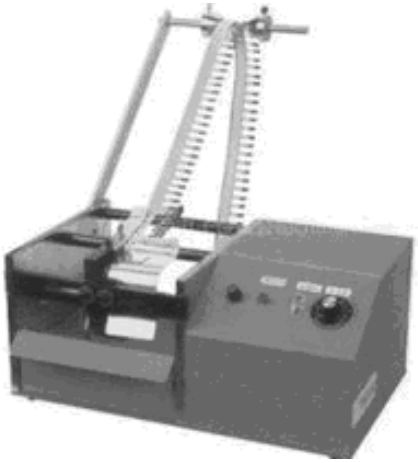


Figure 2.32　Automatic Aircraft　自动切脚机

3）Tin immersion Equipment　浸锡设备

Hand Baptist Tin Heater Tool Grinder　普通手浸锡炉

Ultrasonic Tin Dipping Machine　超声波浸锡机

Figure 2.33　Tin Immersion Equipment　浸锡设备

4）Ultrasonic Cleaning Machine　超声波清洗机

Figure 2.34　Ultrasonic Cleaning Machine　超声波清洗机

5）Automatic Wave Soldering Machine　全自动波峰焊接机

Figure 2.35　Automatic Wave Soldering Machine　全自动波峰焊接机

6）Reflow Soldering Machine　回流焊接机

Figure 2.36　Reflow Soldering Machine　回流焊接机

7）In-circuit test Equipment　在线测试仪

Circuit Board Test Equipment　电路板测试仪　　Fly Probe Test Equipment　飞针在线测试仪

Figure 2.37　In-circuit Test Equipment　在线测试仪

2.3 Production Process for Printed Circuit Board Fabrication 印制电路板制作工艺

1. Typical Double-sided PCB Fabrication 典型双面板制作工艺（Figure 2.38）

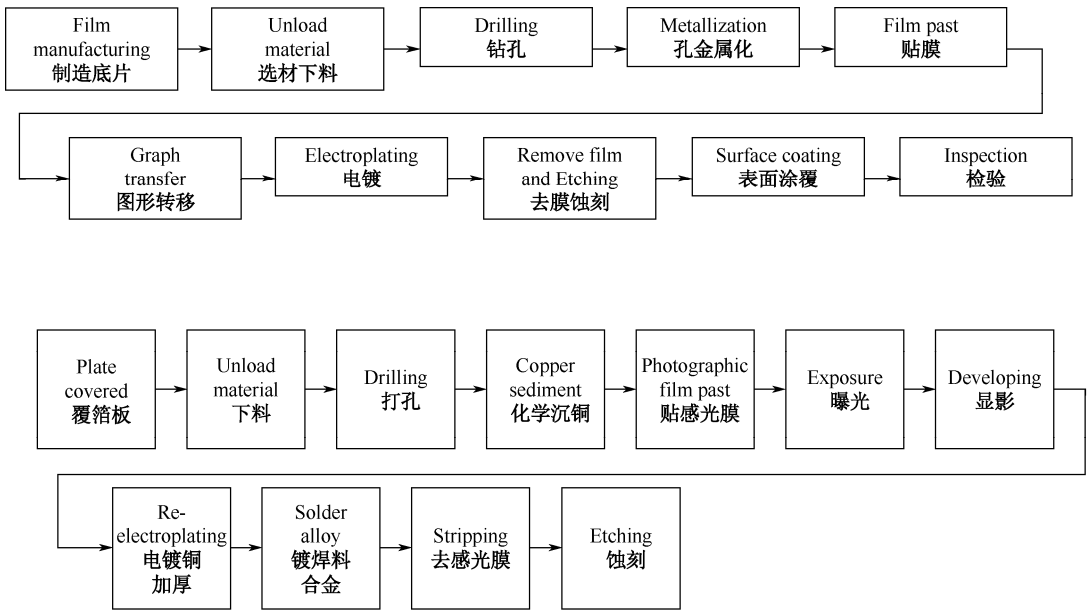


Figure 2.38 Typical Double-sided PCB Fabrication 典型双面板制作工艺

2. Multilayer PCB Fabrication 多层 PCB 制造工艺（Figure 2.39～2.41）

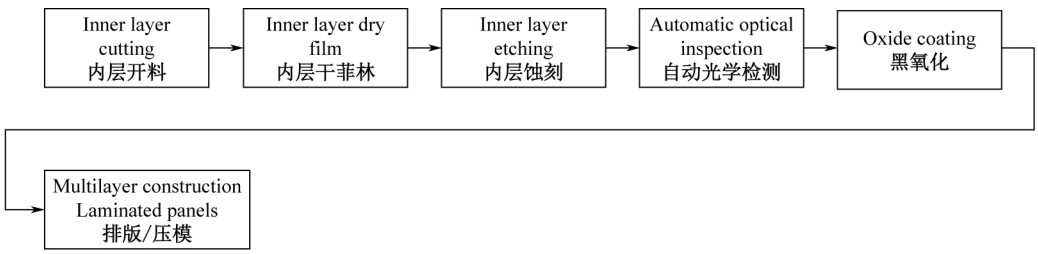


Figure 2.39 Multilayer PCB Fabrication 多层 PCB 制造工艺

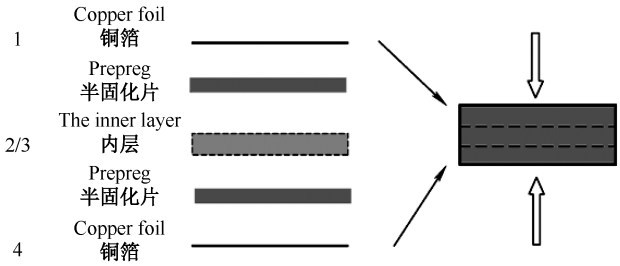


Figure 2.40　4-layer PCB Fabrication and Lamination　4 层板坯料结构与压制

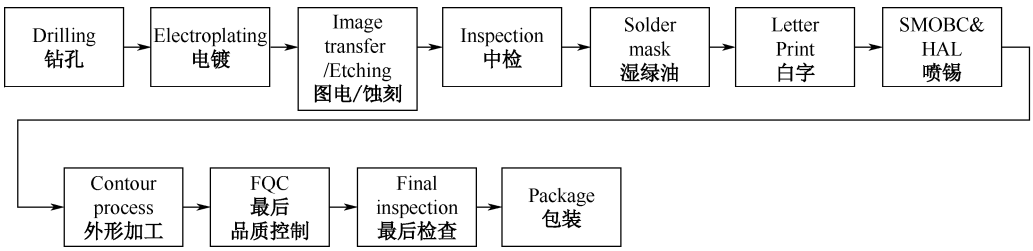


Figure 2.41　The Outer Layer of PCB Process Flow Chart　多层 PCB 外层加工工艺流程

3. 印制电路板手工制作工艺（Figure 2.42～2.44）

1）Heat Transfer Process　热转印工艺

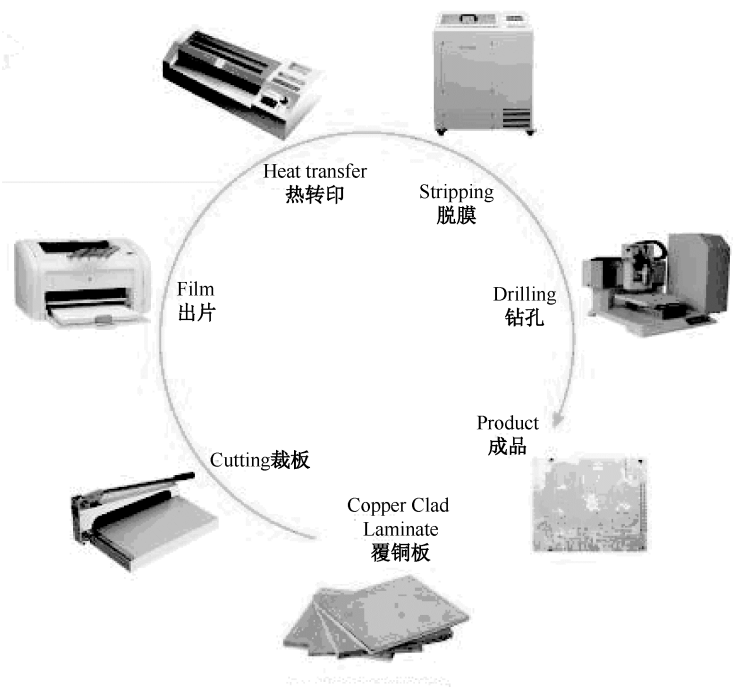


Figure 2.42　Heat Transfer Process　热转印工艺

2）Photosensitive Process　感光工艺

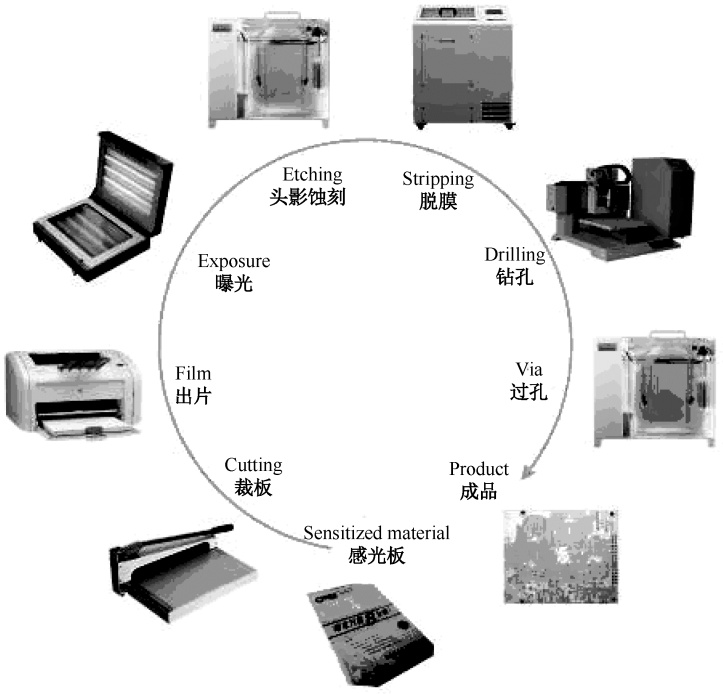


Figure 2.43　Photosensitive Process　感光工艺

3）Engraving Fabrication　雕刻制作工艺

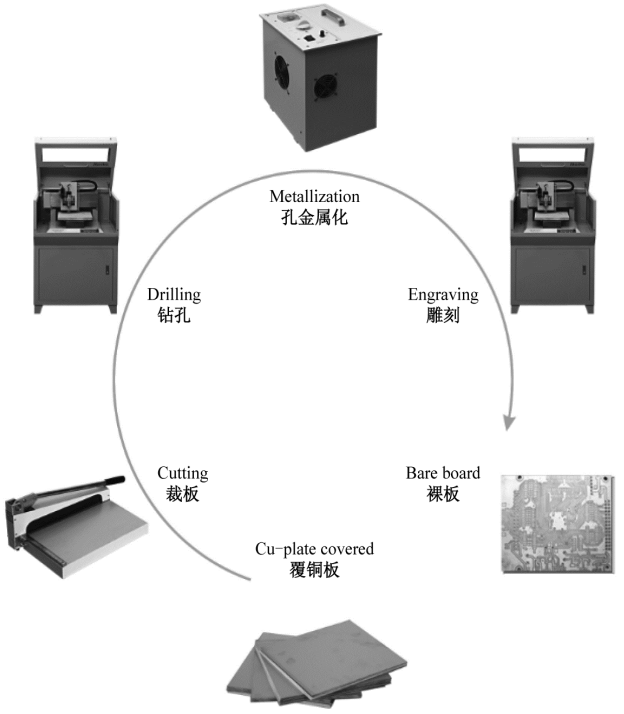


Figure 2.44　Engraving Fabrication　雕刻制作工艺

2.4　Welding Process　电子焊接工艺

1. Plug-in Welding Process　插件焊接工艺（Figure 2.45～2.48）

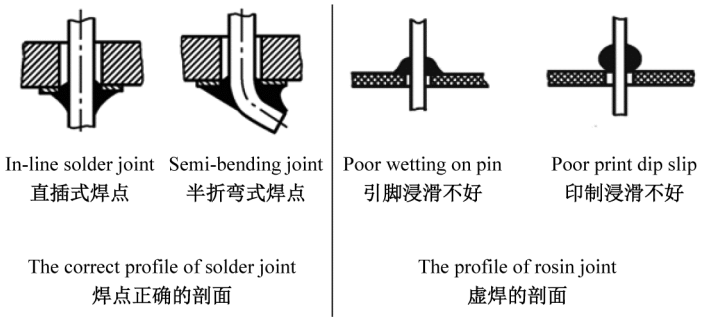


Figure 2.45　The Correct Solder and Rosin Joint　正确焊点与虚焊

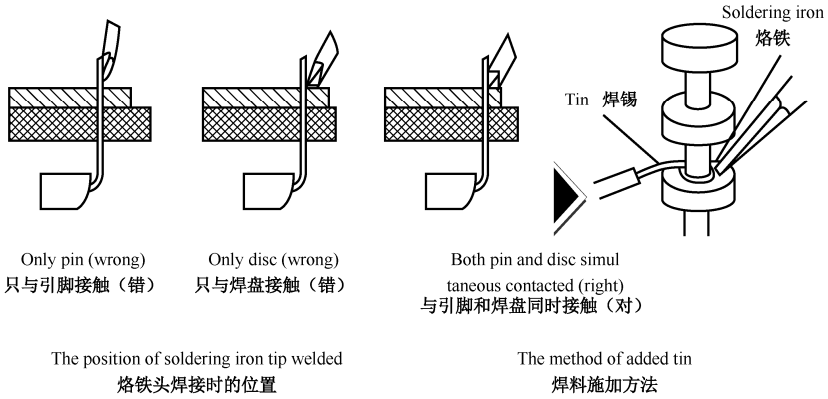


Figure 2.46　The Method of Added Tin and The Position　焊料施加方法和放置位置

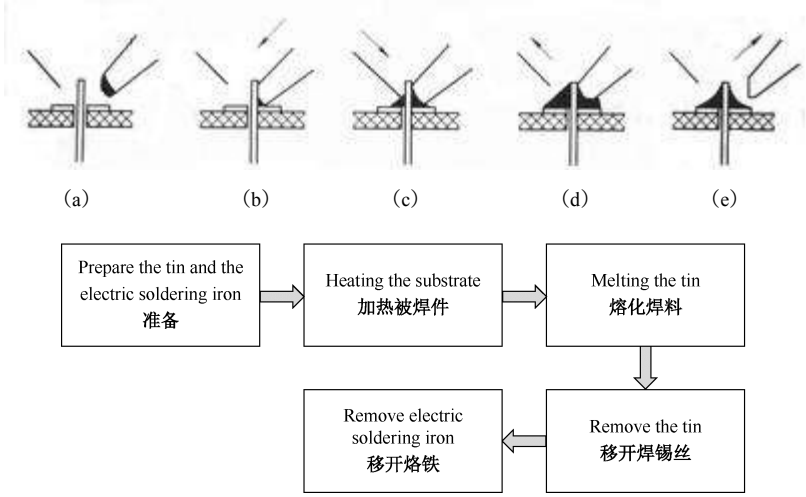


Figure 2.47　5 Steps of Welding Process　五步焊接工艺

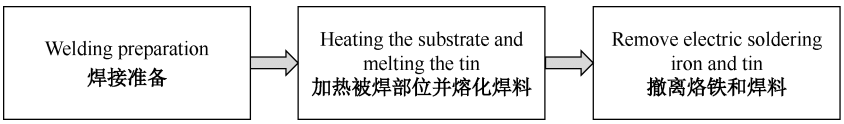


Figure 2.48　3 Steps of Welding Process　三步焊接工艺

2. Surface Mounting Technology (SMT) Welding Process　SMT 焊接工艺（Figure 2.49～2.56）

1）Single-side mixed assemblies processes　单面混装工艺流程

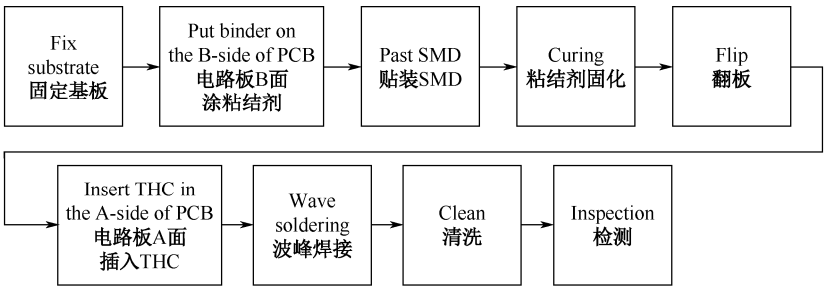


Figure 2.49　First-pasted Process　先贴法工艺流程

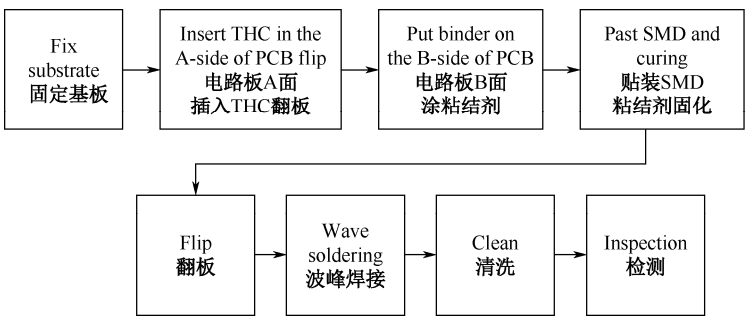


Figure 2.50　Rear-pasted Process　后贴法工艺流程

2）Double-side mixed Assemblies Processes　双面混装工艺流程

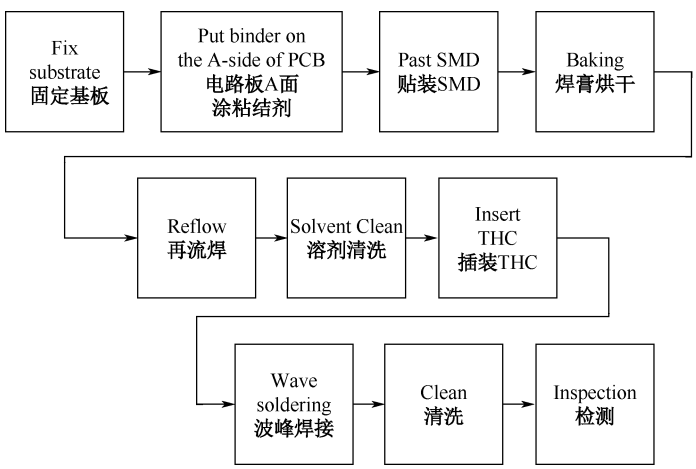


Figure 2.51　Double-side Mixed Assemblies Processes(1)　双面混装工艺流程（1）

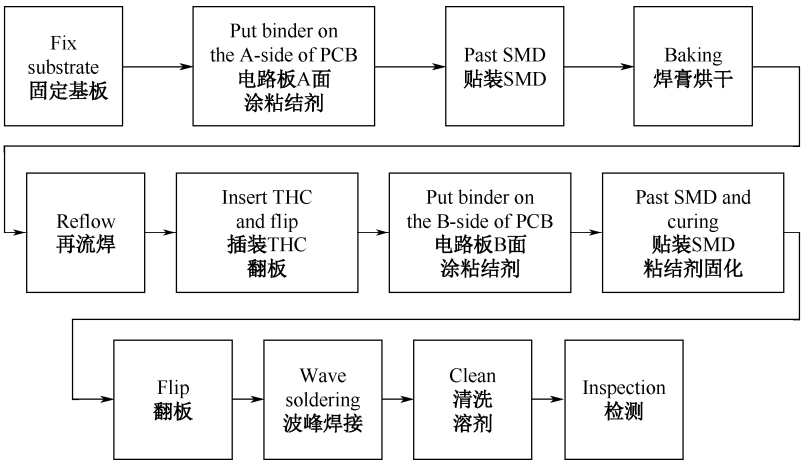


Figure 2.52　Double-side Mixed Assemblies Processes(2)　双面混装工艺流程（2）

3）Complete Surface Mount Process　完全表面安装工艺流程

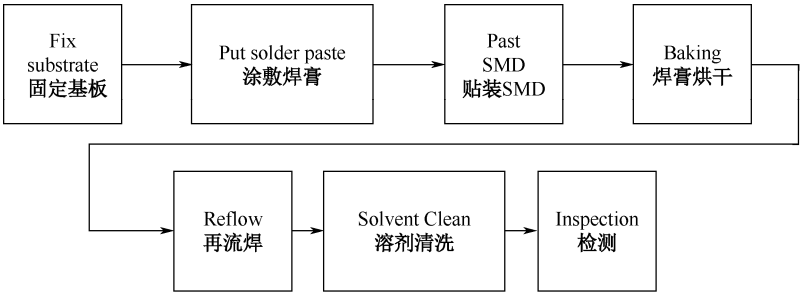


Figure 2.53　One-side Complete Surface Mount Process　单面表面安装工艺流程

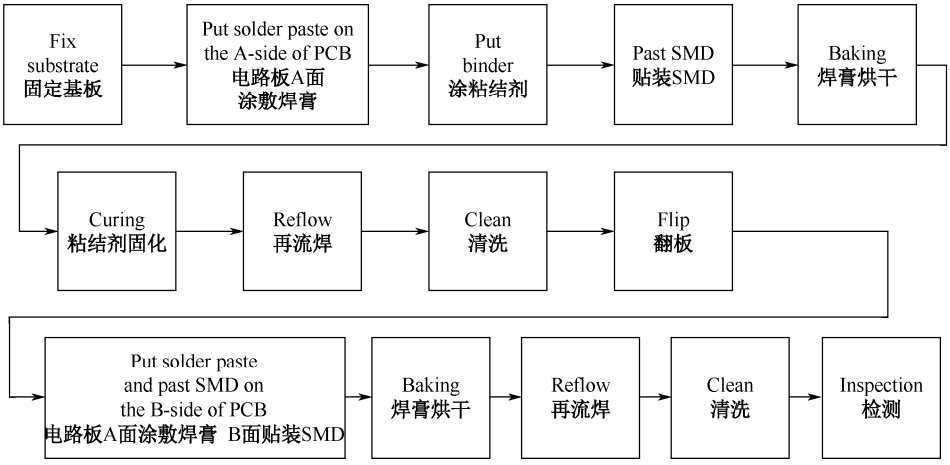


Figure 2.54　Double-side Complete Surface Mount Process(1)　双面表面安装工艺流程（1）

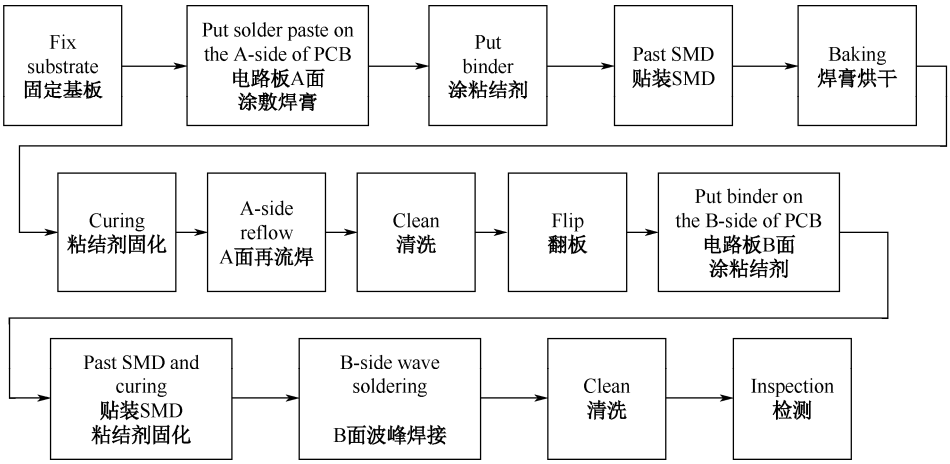


Figure 2.55 Double-side Complete Surface Mount Process(2) 双面表面安装工艺流程（2）

4）SMT Welding Technique SMT 焊接技术

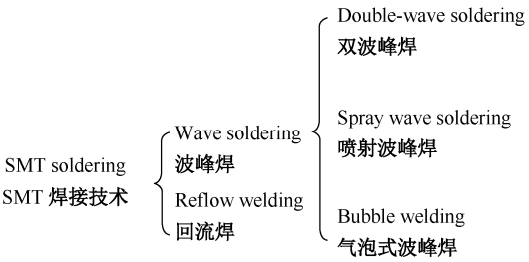


Figure 2.56 SMT Welding Technique SMT 焊接技术

3. Contact-type Technology 接触焊接技术（Figure 2.57～2.61）

1）Crimping 压接

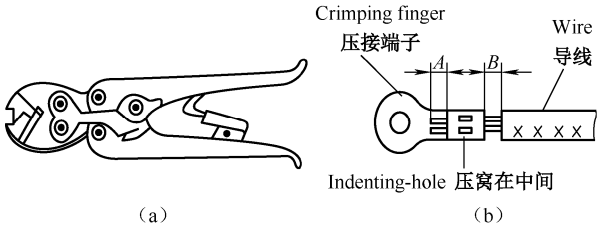


Figure 2.57 Crimping Pliers and Schematic of Crimping 手动压接钳及压接示意图

2）Winding　绕接

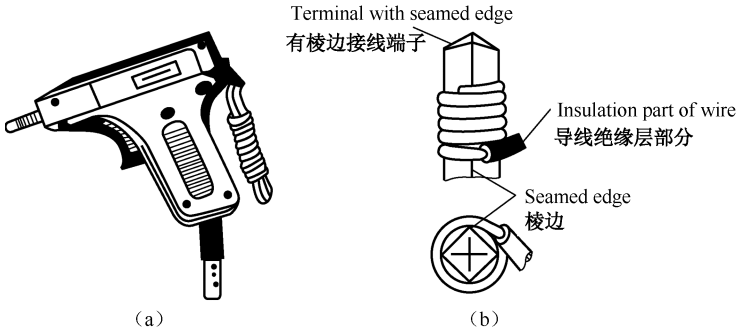


Figure 2.58　Electric Type Winding Gun and Schematic of Winding　电动型绕接枪及示意图

3）Puncture Welding　穿刺

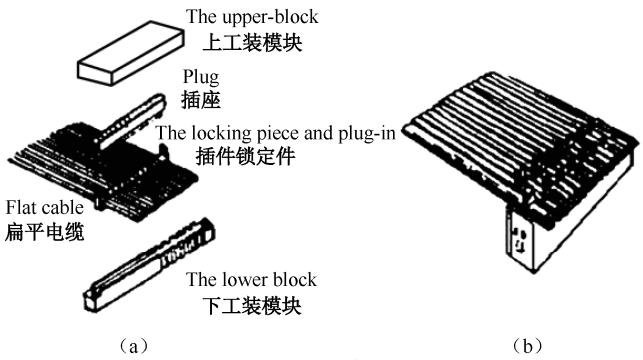


Figure 2.59　Puncture Welding　穿刺

4）Threads　螺纹连接

Slotted Cylinder Screw　一字槽圆柱螺钉

Cone End Locking Screw　锥端紧定螺钉

Cross Recessed Countersuck Head Screw　十字槽平圆头螺钉

Hex Nut　六角螺母

Slotted Head Screw　一字槽沉头螺钉

Spring Washer　弹簧垫片

Cross Recessed Countersuck Head Tapping Screw　十字槽平圆头自攻螺钉

Figure 2.60　Schematics of Some Fasteners in Common Usage　常用紧固件意示图

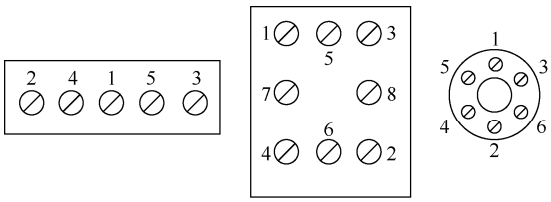


Figure 2.61　Sequence of Lock and Unlock of Screws　螺钉的紧固与拆卸顺序

2.5　Electronic Measurement Process　电子调试工艺

1．Instruments in Common Usage　常用仪器仪表（Figure 2.62～2.68）

1）Universal Meter　万用表

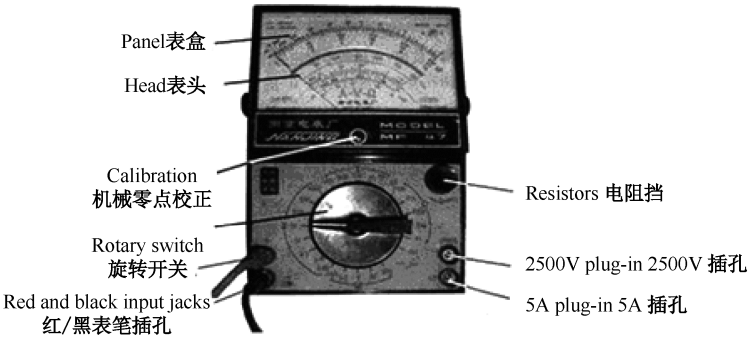


Figure 2.62　Pointer-type Universal Meter　指针式万用表

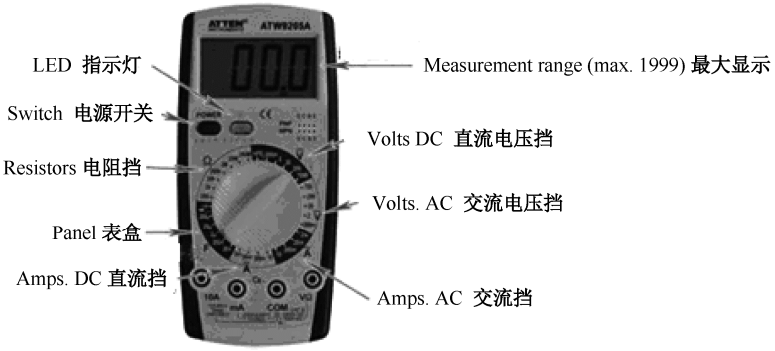


Figure 2.63　Digital-type Universal Meter　数字万用表

2）Milivolt Meter　毫伏表

Figure 2.64　Milivolt Meter　毫伏表

3）Signal Generator　信号发生器

低频信号发生器

Low Frequency Signal Generator

高频信号发生器

High Frequency Signal Generator

Figure 2.65　Signal Generator　信号发生器

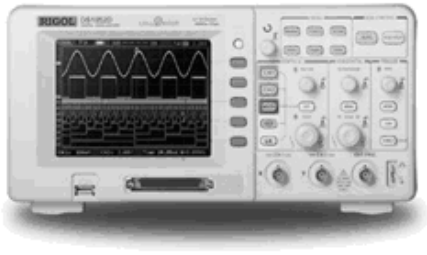
4）General Oscilloscope　通用示波器

单踪示波器

Single-channel Oscilloscope

双踪示波器

Double-channel Oscilloscope



数字示波器

Digital Oscilloscope

Figure 2.66　General Oscilloscope　通用示波器

5）Distortion Meter　失真度仪

Figure 2.67　Distortion Meter　失真度仪

6）Frequency Meter　频率计

Figure 2.68　Frequency Meter　频率计

2. Flow Chart of Electronic Adjustment　电子调试工艺流程（Figure 2.69～2.71）

1）General flow chart of electronic adjustment　电子调试工艺的一般流程

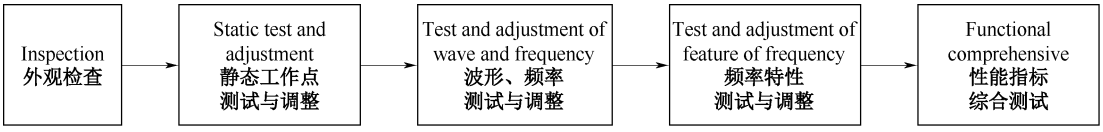


Figure 2.69　Unit Or Component Adjustment Flow Chart　单元部件调试工艺流程

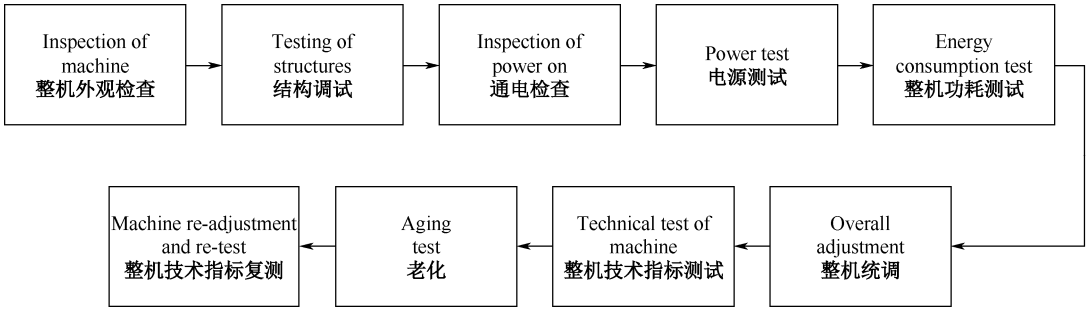


Figure 2.70　Overall Machine Adjustment Flow Chart　整机产品调试工艺流程

2）Adjusted Electronic Technology　电子调试技术

General test：　　常规检测方法：

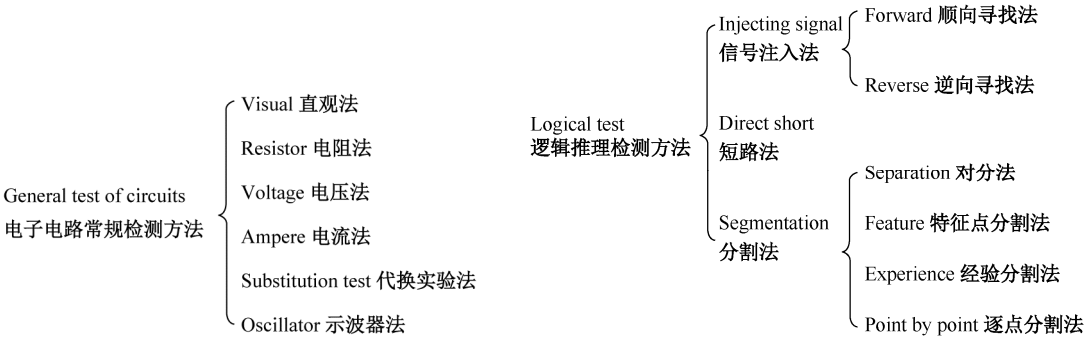


Figure 2.71　General Test　常规检测方法

Exercise　练习题

1. To provide the query data describes the important roles of manufacture procedures of the assembly processes.

2. To provide the query data describes the notice rules of PCB lay out.

3. To provide the query data describes the roles of common tools for electrical productions.

4. To provide the query data describes the difference from the welding tools.

5. To provide the query data describes the features of the electronic measurement process.

PART 3

Inspection of Electronic Products

电子产品检验

Unit 3　Inspection of Electronic Products

第 3 单元　电子产品检验

3.1　Inspection Requirements for Electronic Products
电子产品的检验要求

As Shown in Figure 3.1 and Table 3.1.

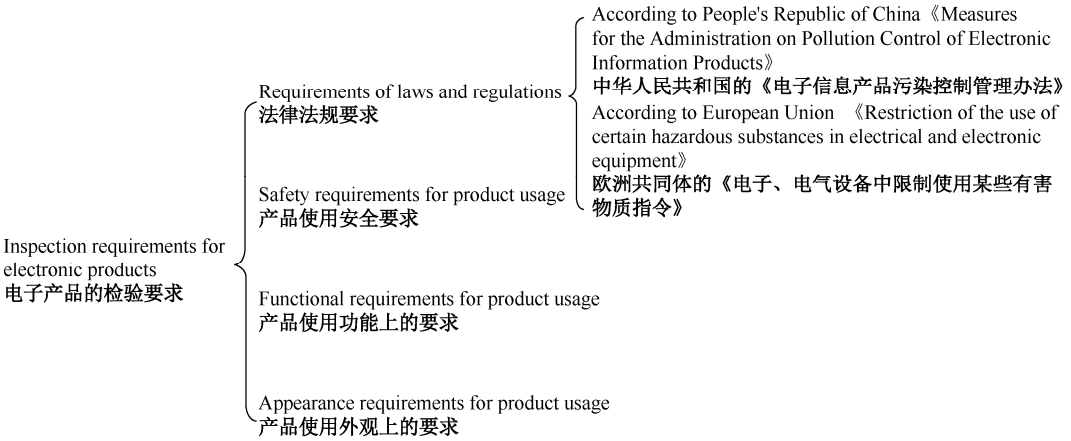


Figure 3.1　Inspection Requirements for Electronic Products　电子产品的检验要求

Table 3.1　Six Major Categories of Hazardous Substances Content Standard Table

ROHS 六大类有害物质含量标准表

Chemical test 化学测试分类	Test item 测试项目	Threshold level/ppm 检测要求（限定值，单位：ppm）
ROHS ROHS 指令	Cd 镉	＞100　≤100
	Pb 铅	＞1000　≤1000

Continue

Chemical test 化学测试分类	Test item 测试项目	Threshold level/ppm 检测要求（限定值，单位：ppm）
ROHS ROHS 指令	Hg 汞	＜ 1000　1000 以内
	Cr(VI) 六价铬	＜ 1000　1000 以内
	PBB 多溴联苯	＜ 1000　1000 以内
	PBDE 多溴联苯醚	＜ 1000　1000 以内
Package 包材指令	94/62/EC 包材指令	Ref. 94/62/EC EU. 《Package》 参考第 94/62/EC 号欧盟《有关包装材料指令》要求
Battery 电池指令	2006/66/EC 新电池指令	Ref. 2006/66/EC 《Battery》 参考第 2006/66/EC 号《关于电池及蓄电池、废弃电池及蓄电池以及废止 91/157/EEC 的指令》要求

3.2 Defects of Electronic Products 电子产品的缺陷

As Shown in Table 3.2～3.3 and Figure 3.2.

Table 3.2 Defect Levels and Descriptions of The Electronic Products
产品的缺陷等级与缺陷描述

List 序号	Defect level 缺陷等级	Symbol 符号	Defect description 缺陷描述
1	Safe 安全问题	S	Product design is not in conformity with IEC or ISO regulations. The product is in a dangerous, so that it is harmful to people or the environment 产品设计与 IEC 或 ISO 法规不符；产品处于一种危险状态，以至于对人或周围环境有所伤害和损害
2	Critical 严重问题	A	Defects will lead to great dissatisfaction with the customer's products, and the defect is easy to be found 缺陷会引起客户对产品的极大不满，且缺陷极易被发现
3	Major 主要问题	B	The main function of the system failure, no collapse. But it will lead to the subsequent operation or work can not continue to carry on. Defects are not acceptable to the customers 系统的主要功能失效，没有崩溃。但会导致后续操作或工作不能继续进行。缺陷是客户无法接受的

Continue

List 序号	Defect level 缺陷等级	Symbol 符号	Defect description 缺陷描述
4	Secondary 次要问题	C	The secondary function of the system fails, but the follow-up operation or work can continue to be carried out. Defects are acceptable to the customers 系统的次要功能失效，但后续操作或工作仍可以继续进行。缺陷是客户可以接受的
5	Minor 微不足道的问题	D	Not worth mentioning failure function, customers may become aware of the failure, the failure does not affect the diagnosis and examination. Will not cause customer dissatisfaction, the defect is not easy to be detected 微不足道的功能失效，客户也许觉察不到该功能失效，该功能失效不影响诊断和检查。不会引起客户不满，缺陷不易被察觉
6	Enhancements 需要改善的问题	E	Did not cause functional failure, is not a defect. Customer use process recommendations to improve the problem. This problem may be related to the manufacturing, service, product cost and other factors 没有引起功能失效，不是一个缺陷。客户使用过程建议改善的问题。这个问题可能涉及到可制造性、可服务性、产品的成本等因素

Table 3.3　Defect States and Descriptions of The Electronic Products

电子产品的缺陷状态与缺陷状态描述

Defect state 缺陷状态	Defect description 缺陷状态描述
Root cause is unknown 原因未知	Newly discovered defects or legacy defects of previous versions of the product. Causes of defects have not yet been detected. For the current state, the team corresponds to the activities are: to find the root cause of defects 新发现的缺陷或以前版本产品的遗留缺陷。引起缺陷的原因尚未查出。当处于当前状态时，团队所对应的活动是：查找引起缺陷的根本原因
Assigned 指派	Has been judged a defect caused by reasons, defects were assigned to specific personnel to resolve, but also specific solutions to planning or implementation process. For the current state, the team corresponds to the activities are: planning solutions, implementation of solutions, component / unit test 已判断出引起缺陷的原因，缺陷被指派给具体人员进行解决，但具体的解决方案还在策划或实施过程中。当处于当前状态时，团队所对应的活动是：策划解决方案、实施解决方案、部件/单元测试
Resolved 已解决	The cause of the defect has been identified, the solution has been implemented and passed the component / unit test. For the current state, the team corresponds to the activities are: system validation or confirmation 引起缺陷的原因已确定，解决方案已经被实施并通过了部件/单元测试。当处于当前状态时，团队所对应的活动是：系统验证或内部确认
Verified 已验证	Causes of defects have been identified, the implementation of the solution has been verified by the system and the confirmation. The status of the product can be submitted for design validation, manufacturing process validation and service validation. For the current state, the team corresponds to the activities are: confirm design, process of the production and service 引起缺陷的原因已确定，实施解决方案后已通过系统验证及内部确认。产品状态可以提交进行设计确认、生产过程确认和服务确认。当处于当前状态时，团队所对应的活动：设计确认、生产过程确认和服务确认

Continue

Defect state 缺陷状态	Defect description 缺陷状态描述
Validated 已确认	Solution has been confirmed by design, process , service. For the current state, the team corresponds to the activities are: solution release to production, service, quality and other related departments 解决方案已经通过设计确认，生产过程确认和服务确认。当处于当前状态时，团队所对应的活动：解决方案发布至生产、服务、质量等相关部门
Closed 已关闭	Solution has been officially released to the production, service, supply chain, quality, and other related departments. The related work has been updated, for example, BOM of product, procurement contracts, etc 解决方案已经正式发布至生产、服务、供应链、质量等相关部门。并且相关工作都已更新完毕。比如产品 BOM、采购合同等
Postponed 暂不解决	Defects caused by the root causes have been identified, There are no solutions or in abeyance, and extension. The defect can be accepted 引起缺陷的根本原因已确定，但暂无解决方案或解决方案暂不实施，须延期解决，且缺陷可以被接受
Inefficacy 无效	Repeated, unable to reproduce such invalid defects 重复、无法重现等无效缺陷

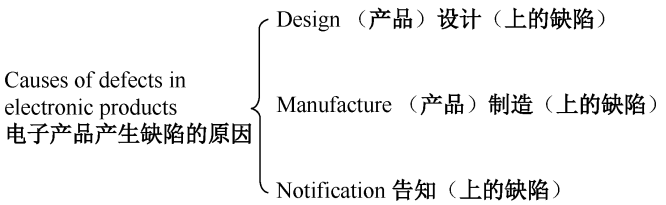


Figure 3.2 Causes of Defects in Electronic Products 电子产品产生缺陷的原因

3.3 Influencing Factors and Classification of Electronic Products Inspection 电子产品检验的影响因素与分类

As Shown in Figure 3.3～3.4.

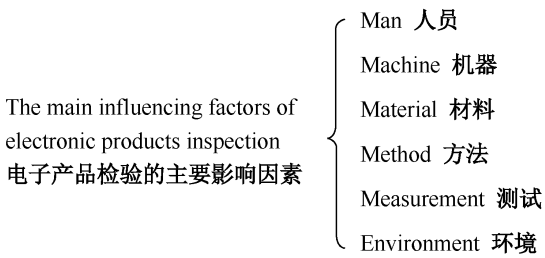


Figure 3.3 The Main Influencing Factors of Electronic Products Inspection
电子产品检验的主要影响因素

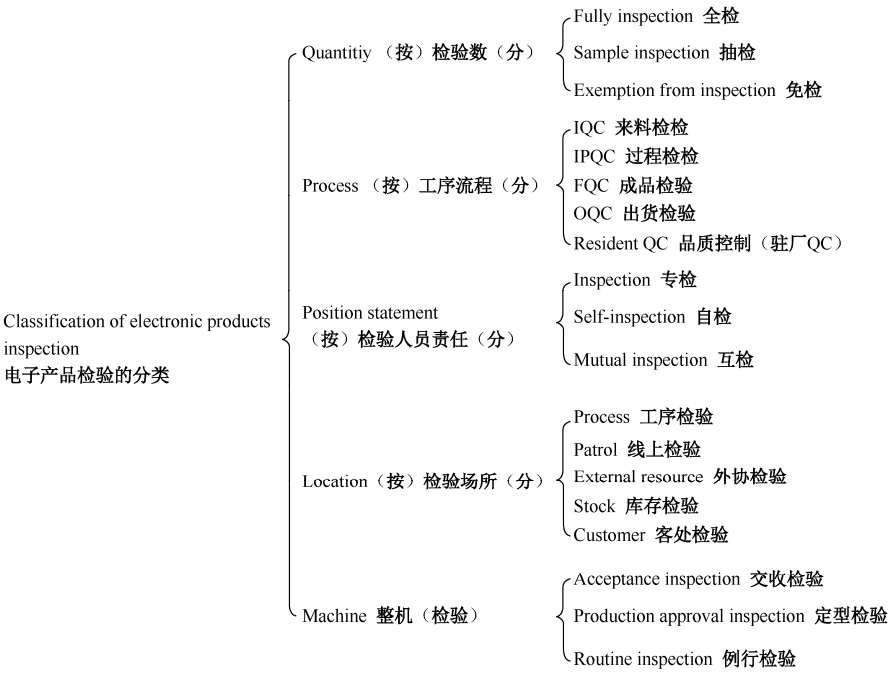


Figure 3.4　Classification of Electronic Products Inspection　电子产品检验的分类

3.4　Contents of Electronic Products Inspection 电子产品检验的内容

As Shown in Figure 3.5～3.7.

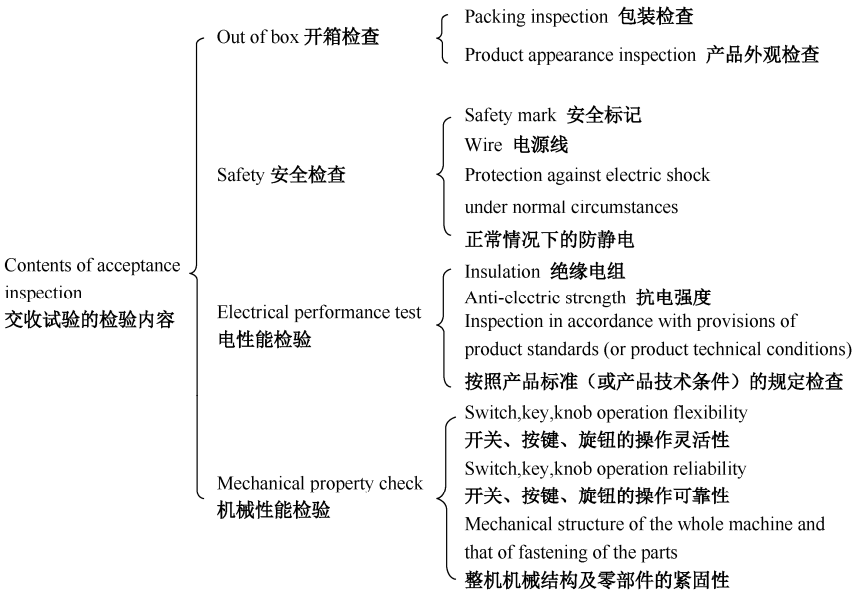


Figure 3.5　Contents of Acceptance Inspection　交收试验的检验的内容

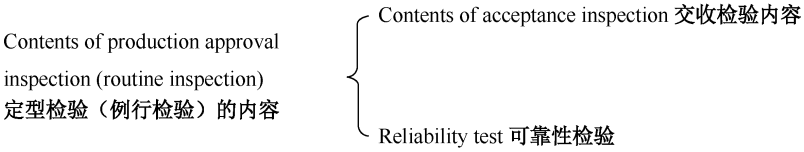
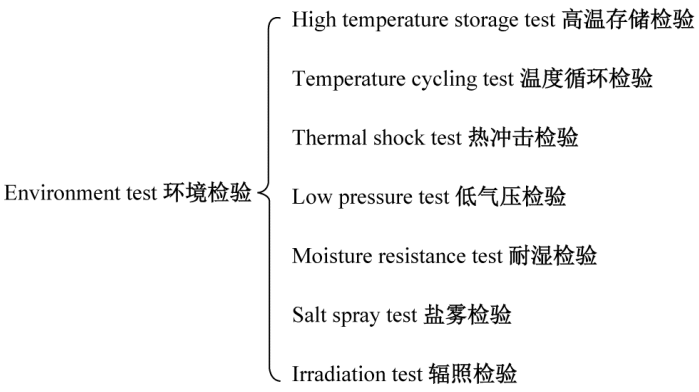
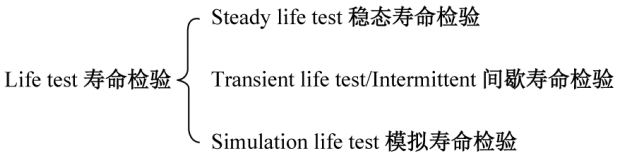


Figure 3.6　Contents of Production Approval Inspections
定型检验内容





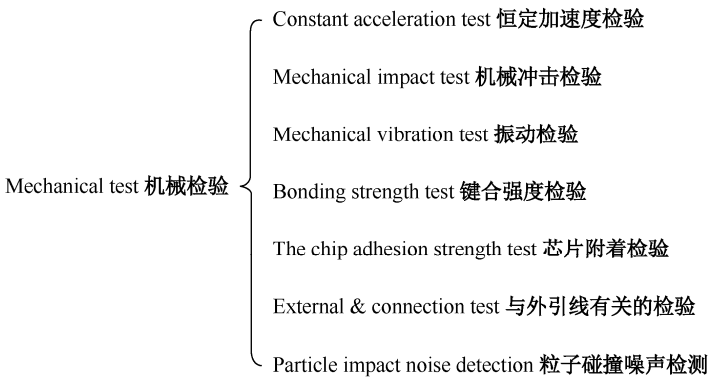


Figure 3.7　Reliability Test　可靠性检验

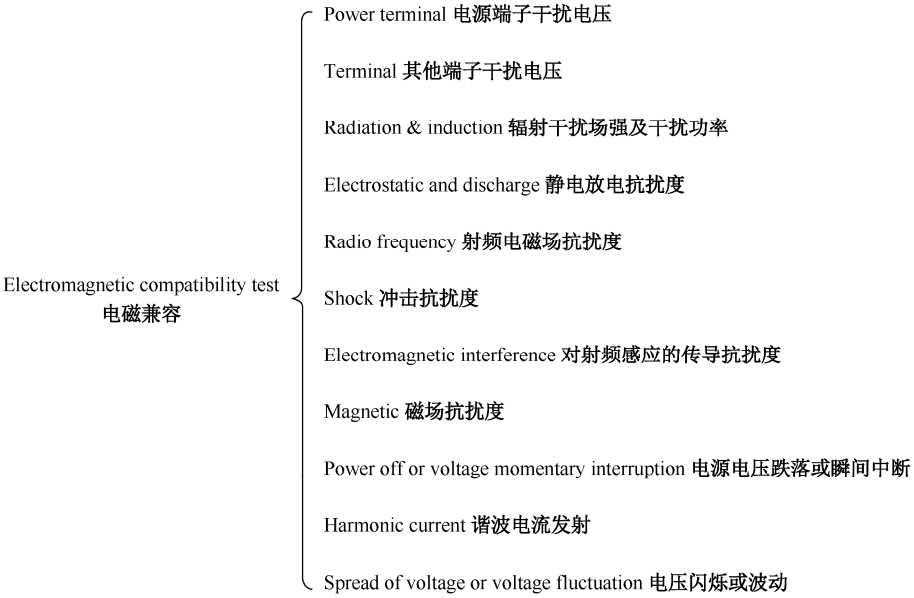


Figure 3.7　Reliability Test　可靠性检验（续）

3.5　Inspection Processes of Electronic Products　电子产品检验工艺

1. Inspection Flow Chart of Electronic Products　电子产品检验的工艺流程（Figure 3.8～3.9）

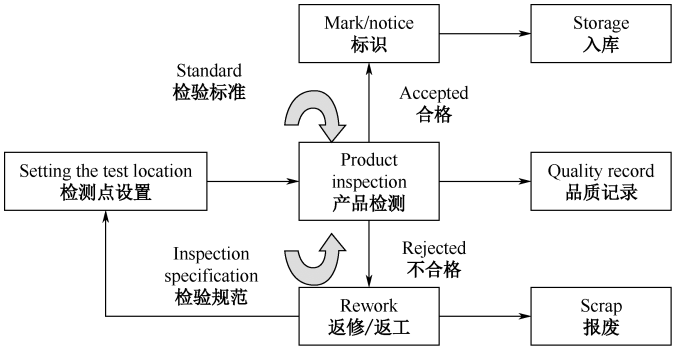


Figure3.8　Inspection Processes of Electronic Products　电子产品检验的一般流程

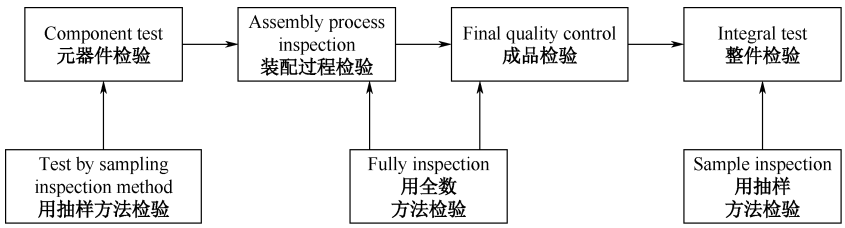


Figure 3.9　Inspection Processes and Required of Electronic Products　电子产品一般检验工艺与检验要求

2. Sample Inspection 抽样检验

1）Method of Sample Inspection

According to a batch in the sample test results, they are determined whether to accept the batch called "sampling inspection". Sample test results depend on the sample *N* corresponding to the number of accepted "*Ac*" and the number of rejected "*Re*". The number of defects or defective products found in the sample is called "*r*". If *r* is less than or equal to *Ac*, one can receive the batch; if *r* is greater than or equal to *Re*, the batch should be rejected.

2）GB—2828 Standard

GB — 2828 standard can also be used for continuous batch sampling. The production process is consistent and stable for the premise. Production process is in the effective and strict quality system assurance. In order to control the quality of batches of products due to the impact of some of the control factors caused by changes. GB—2828 strictly according to the degree of divided into three sampling schemes: normal inspection, strict inspection, and relaxation test, according to certain conditions can be mutual conversion.

（1）Sample code. GB—2828 standard A, C, B... R said sample code. According to batch *N* and inspection level specified. for example: *N* = 1000, inspection level is I, from Table 3.4 can check sample code.

1）抽样检验的方式

根据从批中抽取的样本的检验结果决定是否接收该批，称为“抽样检验”，抽样检验结果取决于样本 *N* 对应的接收数 *Ac* 及拒收数 *Re*，样本中检验发现的缺陷或缺陷产品数为 *r*。如果 *r*<*Ac*，则认为可接收此批；如果 *r*≥*Re*，则认为应拒收此批。

2）GB—2828 标准

GB—2828 标准也可用于连续批抽样，前提是：产品的生产过程是一致的、稳定的，生产过程是在有效的、严格的品质保证体系下进行的，为了控制产品的批次质量由于某些未受控制因素的影响而产生的变化，GB—2828 按严格程度分为三种抽样方案：正常检验、加严检验及放宽检验，在一定条件下可以相互转换。

（1）样本量代码。GB—2828 标准用英文大写字母 A、B、C…R 表示样本量代码。根据批量 *N* 及指定的检验水平（进行检验）例如：设 *N*=1000，采用检验水平 I，从表 3.4 中可查得样本量代码。

Table 3.4 Sample Size Code Table 样本量代码对照表

Batch *N* 批量 *N*	Special inspection level 特别检验水平				Usually test level 通常检验水平		
	S-1	S-2	S-3	S-4	I	II	III
2～8	A	A	A	A	A	A	B
9～15	A	A	A	A	A	B	C
16～25	A	A	B	B	B	C	D
26～50	A	B	B	B	B	D	E
51～90	B	B	C	C	C	E	F
91～150	B	B	C	D	D	F	G
151～280	B	C	D	E	F	G	H

Continue

Batch *N* 批量 *N*	Special inspection level 特别检验水平				Usually test level 通常检验水平		
	S-1	S-2	S-3	S-4	I	II	III
281～500	B	C	D	E	F	H	J
501～1200	C	C	E	F	G	J	K
1201～3200	C	D	E	G	H	K	L
3201～10000	C	D	F	G	J	L	M
10001～35000	C	D	F	H	K	M	N
35001～150000	D	E	G	J	L	N	P
150001～500000	D	E	G	J	M	P	Q
500001 以上	D	E	H	K	N	Q	R

（2）Usage of sampling scheme GB2828. It has been clarified the batch (*N*) of productions, *AQL*, and the strict degree of the inspection level.

The first step: according to *N* and inspection level to find the sampling scheme code.

The second step: according to the selection of strict examination of inspection of corresponding sample inspection table.

Check the normal inspection "sub sampling scheme", see Table 3.5; Search a strict inspection of the "sub sampling scheme ", see Table 3.6; Relaxation inspection "sub sampling scheme ", see Table 3.7.

The third step: according to the sampling scheme code to the right in the table, in the sample bar check the sample *n*, and then from the code location, *AQL*, where the column of the cross grid, read [*Ac*, *Re*]. If the intersection is an arrow not a number, then enter the fourth step.

The fourth step: along the direction of the arrow, read the arrow point to the first [*Ac*, *Re*] and the [*Ac*, *Re*] that of left to the correspond *n*. If *n* is larger than batch *N*, it will be been inspection all and cancelled *n* in the third step.

（2）抽样方案的使用。假设已经明确产品的批量 *N*、*AQL* 值、检验水平检验的严格程度。

第一步：根据批量 *N* 及检验水平查得抽样方案代码。

第二步 :依据选定的检验的严格程度查相应的抽样检查表。

检索正常检验“ 一次抽样方案 ”，见表 3.5；检索加严检验“ 一次抽样方案 ”，见表 3.6 ;检索放宽检验‘ 一次抽样方案 ”，见表 3.7。

第三步 :在表中，根据抽样方案代码向右，在样本栏查得样本量 *n*，再从代码所在行、*AQL* 所在列的交叉格中，读出 [*Ac*，*Re*]。如果该交叉格中不是数字而是箭头，则进入第四步。

第四步：沿着箭头方向，读出箭头所指第一个[*Ac*，*Re*]，然后由此[*Ac*，*Re*]所在行向左，在样本栏读出相应样本量 *n*, 这个时候第三步所得的样本量 *n* 作废，当样本量 *n* 大于批量 *N* 时，则进行全数检验。

Table 3.5 Normal Inspection: A Sampling Test 正常检验一次抽样方案（主表）

样本大小字码	样本量	合格质量水平（AQC）																																					
		0.01		0.015		0.025		0.04		0.065		0.10		0.15		0.25		0.40		0.65		1.0		1.5		2.5		4.0		6.5		10		15		25		40		65		100	
		Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ac	Re
A	2	↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		1	2	2	3	3	4	5	6
B	3	↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		0	1	↑		↓		1	2	2	3	3	4	5	6
C	5	↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		0	1	↑		↓		1	2	2	3	3	4	5	6	7	8
D	8	↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		0	1	↑		↓		1	2	2	3	3	4	5	6	7	8	10	11	14	15
E	13	↓		↓		↓		↓		↓		↓		↓		↓		↓		↓		0	1	↑		↓		1	2	2	3	3	4	5	6	7	8	10	11	14	15
F	20	↓		↓		↓		↓		↓		↓		↓		↓		↓		0	1	↑		↓		1	2	2	3	3	4	5	6	7	8	10	11	14	15	21	22
G	32	↓		↓		↓		↓		↓		↓		↓		↓		0	1	↑		↓		1	2	2	3	3	4	5	6	7	8	10	11	14	15	21	22	↑	
H	50	↓		↓		↓		↓		↓		↓		↓		0	1	↑		↓		1	2	2	3	3	4	5	6	7	8	10	11	14	15	21	22	↑		↑	
J	80	↓		↓		↓		↓		↓		↓		0	1	↑		↓		1	2	2	3	3	4	5	6	7	8	10	11	14	15	21	22	↑		↑		↑	
K	125	↓		↓		↓		↓		↓		0	1	↑		↓		1	2	2	3	3	4	5	6	7	8	10	11	14	15	21	22	↑		↑		↑		↑	
L	200	↓		↓		↓		↓		0	1	↑		↓		1	2	2	3	3	4	5	6	7	8	10	11	14	15	21	22	↑		↑		↑		↑		↑	
M	315	↓		↓		↓		0	1	↑		↓		1	2	2	3	3	4	5	6	7	8	10	11	14	15	21	22	↑		↑		↑		↑		↑		↑	
N	500	↓		↓		0	1	↑		↓		1	2	2	3	3	4	5	6	7	8	10	11	14	15	21	22	↑		↑		↑		↑		↑		↑		↑	
P	800	↓		0	1	↑		↓		1	2	2	3	3	4	5	6	7	8	10	11	14	15	21	22	↑		↑		↑		↑		↑		↑		↑		↑	
Q	1250	0	1	↑		↓		1	2	2	3	3	4	5	6	7	8	10	11	14	15	21	22	↑		↑		↑		↑		↑		↑		↑		↑		↑	
R	2000	↑		↑		1	2	2	3	3	4	5	6	7	8	10	11	14	15	21	22	↑		↑		↑		↑		↑		↑		↑		↑		↑		↑	

样本大小字码	样本量	合格质量水平（AQC）																																					
		100		150		250		400		650		1000																													
		Ac	Re	Ac	Re	Ac	Re	Ac	Re	Ae	Re	Ac	Re																												
A	2	7	8	10	11	14	15	21	22	30	31																														
B	3	7	8	10	11	14	15	21	22	30	31	44	45																												
C	5	10	11	14	15	21	22	30	31	44	45	↑																													
D	8	14	15	21	22	30	31	44	45	↑		↑																													
E	13	21	22	30	31	44	45	↑		↑		↑																													
F	20	↑		↑		↑		↑		↑		↑																													
G	32	↑		↑		↑		↑		↑		↑																													
H	50	↑		↑		↑		↑		↑		↑																													
J	80	↑		↑		↑		↑		↑		↑																													
K	125	↑		↑		↑		↑		↑		↑																													
L	200	↑		↑		↑		↑		↑		↑																													
M	315	↑		↑		↑		↑		↑		↑																													
N	500	↑		↑		↑		↑		↑		↑																													
P	800	↑		↑		↑		↑		↑		↑																													
Q	1250	↑		↑		↑		↑		↑		↑																													
R	2000	↑		↑		↑		↑		↑		↑																													

Table 3.6 Strict Inspection: A Sampling Test 加严检验一次抽样方案（主表）

样本大小字码	样本量	合格质量水平（AQC）																									
		0.01	0.015	0.025	0.04	0.065	0.10	0.15	0.25	0.40	0.65	1.0	1.5	2.5	4.0	6.5	10	15	25	40	65	100	150	250	400	650	1000
		Ac Re	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*
A	2	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	8 9	12 13	18 19	27 28
B	3	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	8 9	18 19	27 28	27 28	41 42
C	5	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	8 9	12 13	18 19	27 28	41 42	↑
D	8	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	8 9	12 13	18 19	27 28	41 42	↑	↑
E	13	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	8 9	12 13	18 19	27 28	41 42	↑	↑	↑
F	20	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	8 9	12 13	18 19	↑	↑	↑	↑	↑	↑
G	32	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	8 9	12 13	18 19	↑	↑	↑	↑	↑	↑	↑
H	50	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	8 9	12 13	18 19	↑	↑	↑	↑	↑	↑	↑	↑
J	80	↓	↓	↓	↓	↓	↓	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	8 9	12 13	18 19	↑	↑	↑	↑	↑	↑	↑	↑	↑
K	125	↓	↓	↓	↓	↓	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	8 9	12 13	18 19	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
L	200	↓	↓	↓	↓	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	8 9	12 13	18 19	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
M	315	↓	↓	↓	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	8 9	12 13	18 19	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
N	500	↓	↓	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	8 9	12 13	18 19	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
P	800	↓	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	8 9	12 13	18 19	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
Q	1250	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	8 9	12 13	18 19	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
R	2000	0 1	↑	↓	1 2	2 3	3 4	5 6	8 9	12 13	18 19	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑

Table 3.7 Relaxation Inspection: A Sampling Test 放宽检验一次抽样方案（主表）

样本大小字码	样本量	合格质量水平（AQC）																									
		0.01	0.015	0.025	0.04	0.065	0.10	0.15	0.25	0.40	0.65	1.0	1.5	2.5	4.0	6.5	10	15	25	40	65	100	150	250	400	650	1000
		Ac Re	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*	*Ac Re*
A	2	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↓	↓	1 2	2 3	3 4	5 6	7 8	10 11	14 15	21 22	30 31
B	2	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↑	↓	↓	1 2	2 3	3 4	5 6	6 7	10 11	14 15	21 22	30 31
C	2	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↑	↓	↓	1 2	2 3	3 4	5 6	6 7	8 9	10 11	14 15	21 22	↑
D	3	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↑	↓	↓	1 2	2 3	3 4	5 6	6 7	8 9	10 11	14 15	21 22	↑	↑
E	5	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↑	↓	↓	1 2	2 3	3 4	5 6	6 7	8 9	10 11	14 15	21 22	↑	↑	↑
F	8	↓	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↑	↓	↓	1 2	2 3	3 4	5 6	6 7	8 9	10 11	↑	↑	↑	↑	↑	↑
G	13	↓	↓	↓	↓	↓	↓	↓	↓	0 1	↑	↓	↓	1 2	2 3	3 4	5 6	6 7	8 9	10 11	↑	↑	↑	↑	↑	↑	↑
H	20	↓	↓	↓	↓	↓	↓	↓	0 1	↑	↓	↓	1 2	2 3	3 4	5 6	6 7	8 9	10 11	↑	↑	↑	↑	↑	↑	↑	↑
J	32	↓	↓	↓	↓	↓	↓	0 1	↑	↓	↓	1 2	2 3	3 4	5 6	6 7	8 9	10 11	↑	↑	↑	↑	↑	↑	↑	↑	↑
K	50	↓	↓	↓	↓	↓	0 1	↑	↓	↓	1 2	2 3	3 4	5 6	6 7	8 9	10 11	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
L	80	↓	↓	↓	↓	0 1	↑	↓	↓	1 2	2 3	3 4	5 6	6 7	8 9	10 11	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
M	125	↓	↓	↓	0 1	↑	↓	↓	1 2	2 3	3 4	5 6	6 7	8 9	10 11	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
N	200	↓	↓	0 1	↑	↓	↓	1 2	2 3	3 4	5 6	6 7	8 9	10 11	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
P	315	↓	0 1	↑	↓	↓	1 2	2 3	3 4	5 6	6 7	8 9	10 11	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
Q	500	0 1	↑	↑	↓	1 2	2 3	3 4	5 6	6 7	8 9	10 11	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
R	800	↑	↑	↑	1 2	2 3	3 4	5 6	6 7	8 9	10 11	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑

(3) Transfer principle. GB—2828 standard is for the transfer between normal inspection, strict inspection, and relaxation inspection, the beginning is always normal inspection.

① From normal to strict. In the normal inspection, if there are 2 batches of the sequence from 2 to 5, those of be rejected by the initial inspection, strict inspection should be applied from the next batch. (Not to be submitted again) Note: Before they can be submitted again, rejected batch must be through screening, rework and other measures.

② From strict to normal. In the strict inspection, if 5 consecutive batches after initial inspection is received, the next batch will be to normal inspection.

③ From normal to relaxation. In the normal inspection, if the following conditions are all satisfied, the relaxation inspection should be carried out from the next batch:

● 10 consecutive batches were received in the normal inspection.

● Conditions specified in the batch number of the extracted sample unqualified products (or rejected) total less than or equal to Table 3-2 in the prescribed limit number.

● Production is stable.

● Consider the expectation of relaxa -tion inspection to save the cost.

It should be noted that the relaxation inspection is not mandatory, if required, can be returned to the normal respection.

④ From relaxation to normal. When any of the following conditions are to be relaxed, the next batch should be to normal:

● A batch of rejected.

（3）转移原则。GB—2828 标准对正常检验、加严检验、放宽检验之间的转移规则进行了规定，一开始都是正常检验。

① 从正常到加严。在进行正常检验时，凡连续 2、3、4 或 5 批中有 2 批经初检验被拒收（不算再次提交），则从下一批起转为加严检验。注：再次提交批是指拒收后经采取筛选和返工等措施后再提交检验的批。

② 从加严到正常。在进行加严检验时，若连续 5 批经初检验被接收，则从下一批起转为正常检验。

③ 从正常到放宽。在进行正常检验时，若下列诸条件都满足，则从下一批起开始执行放宽检验：

● 连续 10 批进行正常检验，初检验均被接收。

● 上一条件中规定的批数所抽取的样本中，不合格品（或不合格）总数小于或等于表 3-2 中规定的界限数。

● 生产处于稳定状态。

● 考虑希望放宽检验以节省检验费用。

需要注意的是放宽检验是非强制性的，如果需要可转回正常检验。

④ 从放宽到正常。在进行放宽检验时，若出现下列任意情况，则从下一批起转回正常检验：

● 一批被拒收。

● Production starts abnormal or stagnant.

● For other reasons which are considered necessary to return to the normal test.

⑤ Suspension. In strict inspection, when the cumulative numbers do not reach 5 batches, the inspection should be suspended according to the standard until the suppliers take corrective measures to restore the inspection.

● 生产开始不正常或停滞。

● 由于其他原因，认为有必要转回正常检验。

⑤ 暂停检验。在进行加严检验时，在连续批的被检验累积数未达到 5 批时，本标准规定暂停接收检验，等供方采取纠正措施后，才能恢复检验。

Exercise 练习题

1. To provide the query data describes the specifications of inspection requirements for electronic products.

2. To provide the query data describes the hazardous substances limits of electronic products.

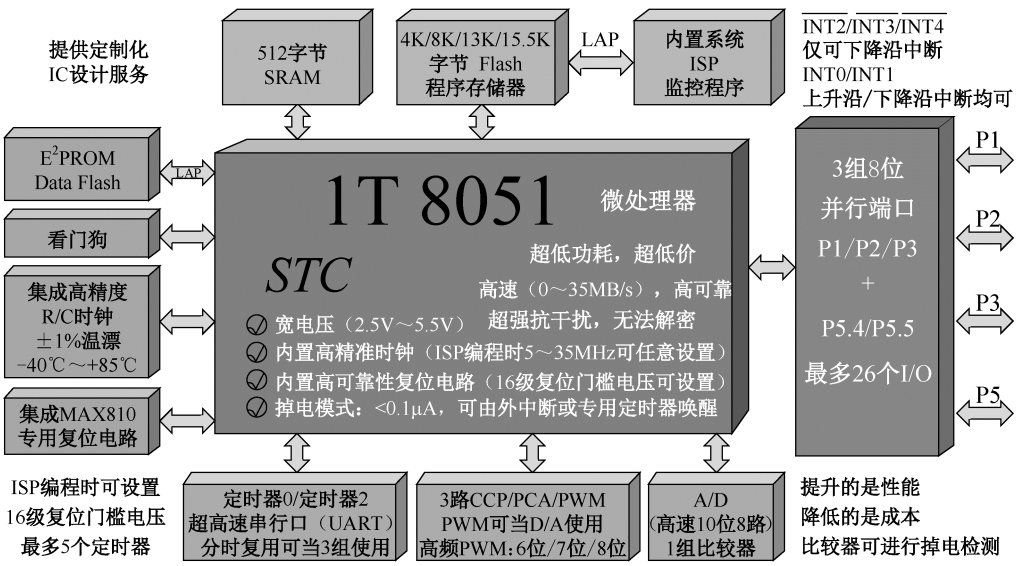
3. To provide the query data describes the roles of the influence factors of electronic products inspection.

4. Please list the inspection standards for electronic products.

PART 4

Electronic System Design

电子系统设计



Unit 4　Structure and Basic Principles of STC15W201S Series MCU

第 4 单元　STC15W201S 单片机结构与工作原理

4.1　Structure and Basic Principles　结构与工作原理

4.1.1　Basic Structure　基本结构

The basic structure of STC15W201S series MCU is shown in Figure 4.1. It comprises CPU, program memory (program flash), data memory (scratch-pad RAM, auxiliary RAM, and specific function registers), EEPROM (data flash), timer/counter, serial communication port, comparator, Watchdog, power monitor, specific reset circuit, internal RC oscillator and some other modules.

STC15W201S 单片机的基本结构如图 4.1 所示，STC15W201S 单片机包含 CPU、程序存储器（程序 Flash）、数据存储器（基本 RAM、扩展 RAM、特殊功能寄存器）、EEPROM（数据 Flash）、定时器/计数器、串行口、中断系统、比较器以及硬件看门狗、电源监控、专用复位电路、内部 RC 振荡器等模块。

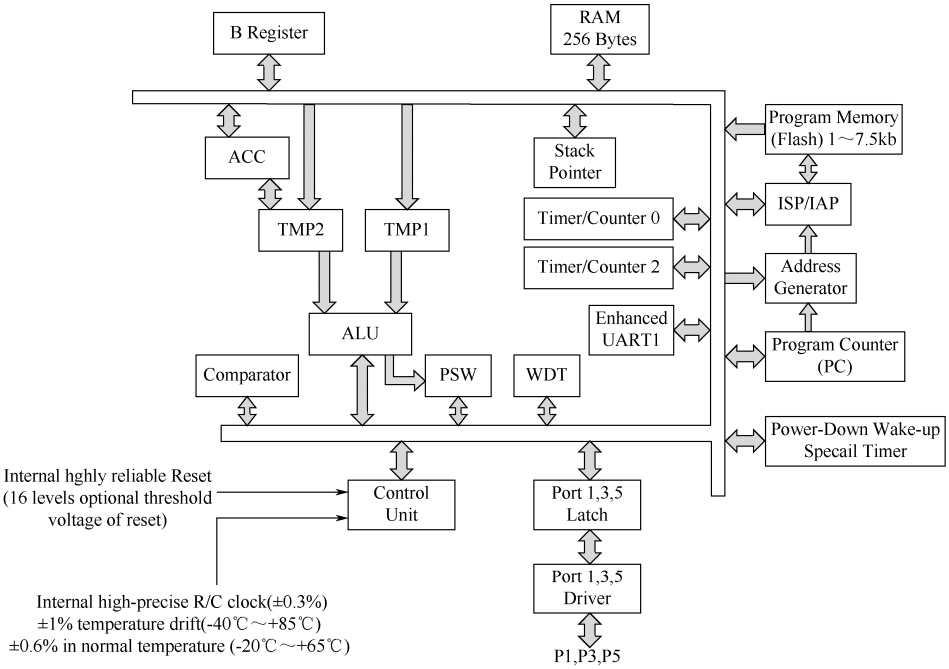


Figure 4.1　Basic structure of STC15W201S series MCU

STC15W201S 系列单片机的基本结构

4.1.2　Pins and Functional Characteristics　引脚与功能特性

There are 2 types of package in STC15W201S, SOP16 and DIP16, which are shown in Figure 4.2.

Pin descriptions of STC15W201S series MCU are shown in Table 4.1.

STC15W201S 单片机有 SOP16 和 DIP16 共 2 种封装形式，STC15W201S 单片机的引脚图如图 4.2 所示。

STC15W201S 单片机的引脚符号以及功能特性如表 4.1 所示。

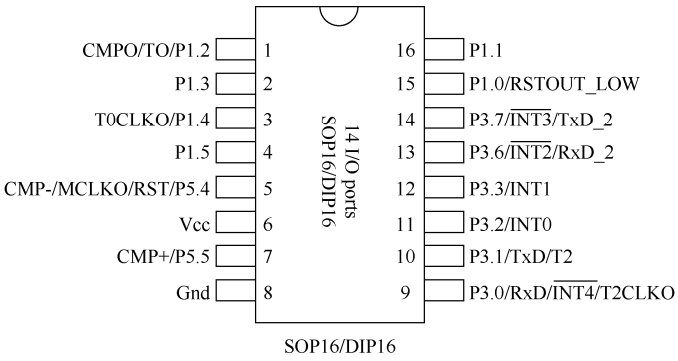


Figure 4.2　Pins of STC15W201S series MCU

STC15W201S 系列单片机的引脚图

Table 4.1 Pin descriptions of STC15W201S series MCU

STC15W201S 单片机的引脚符号以及功能特性

Pin Number	Name	Description	
1	CMPO/T0/P1.2	P1.2	Common I/O port, PORT1[2]
		T0	External input of Timer/Counter 0
		CMPO	Output port of comparator
2	P1.3	Common I/O port, PORT1[3]	
3	T0CLKO/P1.4	P1.4	Common I/O port, PORT1[4]
		T0CLKO	T0 clock output
4	P1.5	Common I/O port, PORT1[5]	
5	CMP-/MCLKO/RST/P5.4	P5.4	Common I/O port, PORT5[4]
		RST	Reset pin
		MCLKO	Master clock output
		CMP-	Comparator negative input
6	*V*cc	The positive pole of power	
7	CMP+/ P5.5	P5.5	Common I/O port, PORT5[5]
		CMP+	Comparator positive input
8	GND	The negative pole of power. Gound	
9	P3.0/RXD/ $\overline{\text{INT4}}$ /T2CLKO	P3.0	Common I/O port, PORT3[0]
		RXD	Receive data port of UART
		$\overline{\text{INT4}}$	External interrupt 4
		T2CLKO	T2 clock output
10	P3.1/TXD/T2	P3.1	Common I/O port, PORT3[0]
		TXD	Transit data port of UART
		T2	External input of Timer/Counter 2
11	P3.2/INT0	P3.2	Common I/O port, PORT3[2]
		INT0	External interrupt 0
12	P3.3/INT1	P3.3	Common I/O port, PORT3[3]
		INT1	External interrupt 1
13	P3.6/ $\overline{\text{INT2}}$ /RXD_2	P3.6	Common I/O port, PORT3[6]/INT1
		$\overline{\text{INT2}}$	External interrupt 2
		RXD_2	Receive data port of UART
14	P3.7/ $\overline{\text{INT3}}$ /TXD_2	P3.7	Common I/O port, PORT3[7]
		$\overline{\text{INT3}}$	External interrupt 3
		TXD_2	Transit data port of UART
15	P1.0	Common I/O port, PORT1[0]	
16	P1.1	Common I/O port, PORT1[1]	

4.1.3 Structure of CPU CPU 结构

The CPU of MCU is composed of arithmetic unit and controller, which is used for reading and analyzing every instruction, and manipulating all components specific computing or operations according to instructions.

1. Arithmetic Unit

Arithmetic unit is composed of arithmetic logic unit (ALU), accumulator (ACC), register B, temporary register (TMP1, TMP2) and program status word register (PSW). It accomplishes arithmetic logical operation, bit variable process, transmitting and so on.

ALU is so powerful that it can accomplish arithmetic operations such as addition, subtraction, multiplication and division, logical operations such as AND, OR, NOT, XOR and circulation, and besides, bit process which is not available in other normal MCU.

Accumulator (ACC), also known as A, which is used for providing operand to ALU and storing ALU operating result, is the busiest register in CPU. Most executions of instruction are carried out by ACC.

Register B is the specialized register used for multiplication and division, and storing the operator and result, while it's a scratch pad register for other instructions.

Program status word register, PSW for short, is used for storing the feature of ALU result and the status of process, which are the requirement of transforming program and serving program differentiating and inquiry. The definition of PSW is shown in Table 4.2.

单片机的中央处理器（CPU）由运算器和控制器组成。它的作用是读入并分析每条指令，根据各指令功能控制单片机的各功能部件执行指定的运算或操作。

1. 运算器

运算器由算术/逻辑运算部件（ALU）、累加器（ACC）、寄存器 B、暂存器（TMP1，TMP2）和程序状态标志寄存器（PSW）组成。它所完成的任务是实现算术与逻辑运算、位变量处理与传送等操作。

ALU 功能极强，既可实现 8 位二进制数据的加、减、乘、除等算术运算和与、或、非、异或、循环等逻辑运算，同时还具有一般微处理器所不具备的位处理功能。

累加器（ACC），又记为 A，用于向 ALU 提供操作数和存放运算结果，是 CPU 中工作最繁忙的寄存器，大多数指令的执行都要通过累加器进行。

寄存器 B 是专门为乘法和除法运算设置的寄存器，用于存放乘法和除法运算的操作数和运算结果。对于其他指令，可作为普通寄存器使用。

程序状态标志寄存器（PSW），简称程序状态字。它用来保存 ALU 运算结果的特征和处理状态。这些特征和状态可以作为控制程序转移的条件，供程序判别和查询。PSW 的各位定义如表 4.2 所示。

Table 4.2 PSW

PSW	Address 地址	D7	D6	D5	D4	D3	D2	D1	D0	Reset Value 复位值
	D0H	CY	AC	F0	RS1	RS0	OV	F1	P	0000 0000

CY: Carry bit. When addition or subtraction is carried out, if there is carry in or out in the MSB (Most Significant Bit) in the result of operation, CY will be set to '1', otherwise cleared to '0'. CY will be cleared after multiplication。

AC: Auxiliary Carry Flag. When addition or subtraction is executed, if there is carry in or out from low order nibble to high order nibble, AC will be set to '1', otherwise cleared to '0'.

F0: User sign 0. Status sign defined by user.

RS1, RS0: Register bank select control bit.

OV: Overflow Flag. Indication of whether there is overflow or not in operation. (OV)=1, indicates there is overflow, while (OV) =0, no overflow.

F1: User sign 1. Status sign defined by user.

P: Parity flag. (P) =0 when the number of digit '1' is even, otherwise (P) =0. Parity check digit can be set according to P bit, that is, adopt NOT-P when odd parity, while P when even parity.

2. Control Unit

Control Unit, composed of Instruction Register (IR), Instruction Decoder (ID), Timer and Control Logic circuit, and Program Counter (PC), is the command center of CPU.

Program Counter (PC) is a counter of 16 bits, which doesn't belong to specific registers, and it always stores the program memory address of next instruction to be taken. Besides, PC will increase after one instruction is taken, preparing for the next

CY：进位标志位。执行加/减法指令时，如果操作结果的最高位 D7 出现进/借位，则 CY 置“ 1 ”，否则清零。执行乘法运算后，CY 清零。

AC：辅助进位标志位。当执行加/减法指令时，如果低四位数向高四位数产生进/借位，则 AC 置“ 1 ”，否则清零。

F0：用户标志 0。由用户定义的状态标志。

RS1、RS0：工作寄存器组选择控制位。

OV：溢出标志位。指示运算过程中是否发生了溢出。有溢出时，（OV）＝1；无溢出时，（OV）＝0。

F1：用户标志 1。由用户定义的状态标志。

P：奇偶标志位。如果累加器 ACC 中 1 的个数为偶数，（P）=0；否则（P）=1。在具有奇偶校验的串行数据通信中，可以根据 P 设置奇偶校验位。如奇校验，取 P 的非值；若偶校验，取 P 值。

2. 控制器

控制器是 CPU 的指挥中心，由指令寄存器（IR）、指令译码器（ID）、定时及控制逻辑电路以及程序计数器（PC）等组成。

程序计数器（PC）是一个 16 位的计数器（注意：PC 不属于特殊功能寄存器）。它总是存放着下一个要取指令字节所在程序存储器存储单元的地址。并且，每取完一个指令字

instruction. Therefore, generally, CPU executes program following the sequence of instructions. However, it's an exception when transition, subprogram instruction calling or interruption response is executed, in such situation that a new address is given to PC automatically by instruction or interruption response process. CPU will start to execute program from the place where the PC points.

Instruction Register (IR) is used for storing the instruction under execution. An instruction is put into IR from program memory before it's executed. Instruction includes opcode, which is delivered to ID and form corresponding micromanipulation signal, and operand, which is delivered to operand generator circuit in order to form a real operand address.

Timing and control are the core component, of MCU which controls instruction taken, instruction execution, operand or operating result access and so on, and delivers all kinds of micromanipulation signals to coordinate the components, and accomplishes the tasks appointed by instructions.

节后，PC 的内容自动加 1，为取下一个指令字节做准备。因此，一般情况下，CPU 是按指令顺序执行程序的。只有在执行转移、子程序调用指令和中断响应时例外，而是由指令或中断响应过程自动给 PC 置入新的地址。PC 指到哪里，CPU 就从那里开始执行程序。

指令寄存器（IR）保存当前正在执行的指令。执行一条指令，先要把它从程序存储器取到指令寄存器 IR 中。指令内容包含操作码和操作数两部分，操作码送至指令译码器（ID），并形成相应指令的微操作信号；操作数送至操作数形成电路，以便形成实际的操作数地址。

定时与控制是微处理器的核心部件，它的任务是控制取指令、执行指令、存取操作数或运算结果等操作，向其他部件发出各种微操作信号，协调各部件工作，完成指令指定的工作任务。

4.1.4 Clock 时钟

1. Selection of Clock

The clock of STC15W201S MCU is internal clock, which can be set by online program software STC-ISP. Its frequency ranges from 5 MHz to 35 MHz with ±1% temperature drift (-40℃～+85℃) and ±0.6% (room temperature).

Firstly, launch STC-ISP online program software, and then select the clock frequency of STC15W201S series MCU in the option "Input IRC frequency" shown in Figure 4.3. The MCU's clock will synchronize to the frequency chosen after the program is downloaded.

1. 时钟的选择

STC15W201S 单片机的时钟是内置时钟，是通过 STC-ISP 在线编程软件进行设置的。时钟频率范围为 5～35MHz，在 -40 ° ～+85 ° 温度环境下，温漂为±1%，在常温下，温漂为±0.6%。

首先打开 STC-ISP 在线编程软件，再单击硬件选项中第一行“输入用户程序运行时的 IRC 频率”的下拉按钮，如图 4.3 所示。待程序下载时，单片机的时钟频率将同步变更为自己所选择的时钟频率。

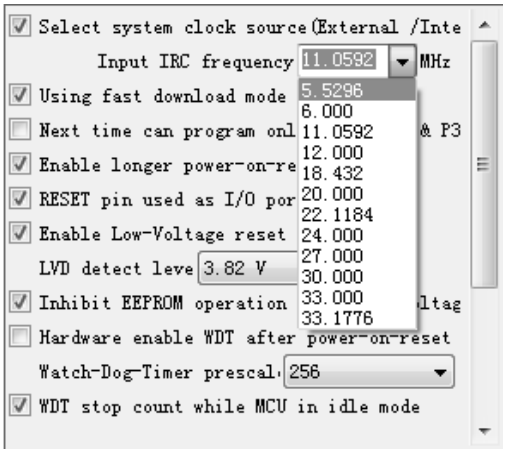


Figure 4.3 Selection of Clock 时钟的选择

2. System Clock and Clock Frequency Division Register

The output of clock source is provided to MCU CPU and inner port via a programmable clock frequency divider instead of connecting the clock of MCU CPU and inner port directly. In order to differentiate each other, the clock source is denoted as f_{OSC}, CPU clock and inner clock are denoted as f_{SYS}, where $f_{SYS}=f_{OSC}/N$, and N is the frequency division coefficient. STC15W201S series MCU can work at lower clock frequency using the clock frequency divider (CLK_DIV) for clock division, thus lower the power at the same time.

The division of system clock is shown in Table4.3.

2. 系统时钟与时钟分频寄存器

时钟源输出信号不是直接与单片机 CPU、内部接口的时钟信号相连的，而是经过一个可编程时钟分频器再提供给单片机 CPU 和内部接口的。为了区分时钟源时钟信号与 CPU、内部接口的时钟，时钟源（振荡器时钟）信号的频率记为 f_{OSC}，CPU、内部接口的时钟称为系统时钟，记为 f_{SYS}。$f_{SYS}=f_{OSC}/N$，其中 N 为时钟分频器的分频系数，利用时钟分频器（CLK_DIV）可进行时钟分频，从而使 STC15W201S 单片机在较低频率工作，降低 STC15W201S 单片机工作的功耗。

系统时钟的分频情况如表 4.3 所示。

Table 4.3 System Clock and Frequency Division Factor 系统时钟与分频系数

CLKS2	CLKS1	CLKS0	Division factor（N）	System clock of CPU
0	0	0	1	f_{OSC}
0	0	1	2	$f_{OSC}/2$
0	1	0	4	$f_{OSC}/4$
0	1	1	8	$f_{OSC}/8$
1	0	0	16	$f_{OSC}/16$
1	0	1	32	$f_{OSC}/32$

Continue

CLKS2	CLKS1	CLKS0	Division factor（*N*）	System clock of CPU
1	1	0	64	$f_{OSC}/64$
1	1	1	128	$f_{OSC}/128$

3. Output and Control of Master Clock

STC15W201S series MCU output master clock on pin P5.4. However it is MCKO_S1 and MCKO_S0 in CLK_DIV that determines whether it will output master clock or not, and what the division of clock is. See Table 4.4 for details.

3. 主时钟输出与主时钟控制

单片机从 P5.4 引脚输出主时钟，但是否输出、输出分频为多少是由 CLK_DIV 中的 MCKO_S1、MCKO_S0 控制的，详见表 4.4。

Table 4.4　Output of Master Clock　主时钟输出功能

MCKO_S1	MCKO_S0	The control bit of master clock output by dividing the frequency 主时钟输出功能的分频控制位
0	0	Master clock do not output external clock　禁止输出
0	1	Master clock output external clock, and the output clock frequency=MCLK/1 输出时钟频率=主时钟频率
1	0	Master clock output external clock, and the output clock frequency is divided by 2 (=MCLK/2) 输出时钟频率=主时钟频率/2
1	1	Master clock output external clock, and the output clock frequency is divided by 4 (=MCLK/4) 输出时钟频率=主时钟频率/4

4.1.5　Reset　复位

CPU and other components of MCU will start from a certain status after reset. Reset is classified into two types, Warm Boot Rest and Cold Boot Reset, whose differences are showed in Table 4.5.

The bit B4 in PCON register is Power-on/Power-off Reset Flag (POF). The POF will be set to "1" after cold boot reset, while will remain unchanged after hot boot reset. In practical application, POF is used to judge whether the reset is power-on reset (cold boot reset) or other reset, such as RST-pin reset, Watchdog reset and software reset. The POF bit should be cleared by software after judgment. Users can do different processes according to the status of POF in initialization as shown in Figure 4.4.

复位是单片机的初始化工作，复位后中央处理器（CPU）及单片机内的其他功能部件都处在一确定的初始状态，并从这个状态开始工作。复位分为热启动复位和冷启动复位两大类，它们的区别如表 4.5 所示。

PCON 寄存器的 B4 位是单片机的上电复位标志位（POF），冷启动后复位标志 POF 为 1，热启动复位后 POF 不变。在实际应用中，该位用来判断单片机复位是上电复位（冷启动复位），还是 RST 外部复位，或看门狗复位，或软复位，但应在判断出上电复位后及时将 POF 清零。用户可以在初始化程序中判断 POF 是否为 1，并对不同情况进行不同的处理，如图 4.4 所示。

Table 4.5 Warm Boot Reset and Cold Boot Reset 热启动复位和冷启动复位对照表

Reset type 复位种类	Reset source 复位源	POF 上电复位标志	Result 复位结果
Cold boot reset 冷启动复位	Cold boot reset namely Power-off/Power-on reset caused by the power of system be off or on 系统停电后再上电引起的硬复位	1	System will reset to ISP address 0000H and begin running ISP monitor program, and system will turn to the user program area automatically by soft reset when there is no legitimate ISP command detected 从系统 ISP 监控程序区开始执行程序，如果检测到 ISP 下载流，进入下载程序状态，结束后自动转到用户程序区执行用户程序；如果检测不到合法的 ISP 下载流，将执行软复位，并跳转到用户程序区执行用户程序
Warm boot reset 热启动复位	External RST Pin Reset 外部 RST 引脚产生的硬复位	Invariant 不变	
	Watch Dog-Timer Reset 内部看门狗复位	Invariant 不变	SWBS=1: System till reset to ISP address 0000H and begin running ISP monitor program； SWBS=0: System till reset to AP address 0000H and begin running user application program 若 SWBS 为 1，则复位到 ISP 监控程序区； 若 SWBS 为 0，则复位到用户程序区 0000H 处
	Soft Reset according to register IAP_CONTR 编程 IAP_CONTR 寄存器产生的软复位	Invariant 不变	

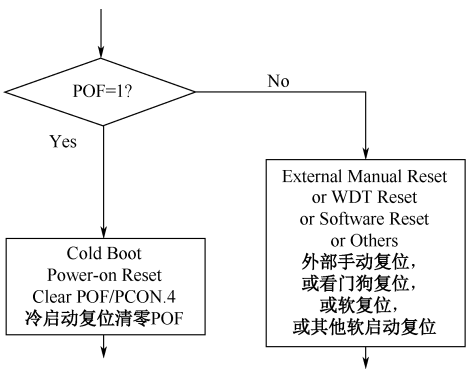


Figure 4.4 The Flow Diagram of "The Reset Type Judgement by User Software"

“用户软件判断复位种类”的流程图

1. The Implementation of Reset

There are several types of reset mode in STC15W201S series MCU: internal power-on reset circuit reset (MAX810 special circuit reset), Watchdog-Timer reset and software reset.

1. 复位的实现

STC15W201S 单片机有多种复位模式：内置上电复位（MAX810 专用复位）、看门狗复位与软件复位。

(1) Internal Power-on Reset Circuit (MAX810 specific reset circuit)

All the logic circuits will be reset when the V_{CC} is lower than the threshold of power-off/power-on reset detection voltage. The reset will be terminated after a delay of 8192 clocks when the V_{CC} goes up to a voltage higher than threshold of reset detection voltage.

The reset will be released a delay of 180ms after the power-off/power-on reset's termination, when the MAX810 special reset circuit is enable in ISP.

(2) External RST Pin Reset. External RST pin reset accomplishes the MCU reset by applying an active-high reset pulse to RST pin from external. The P5.4/RST pin is set as I/O port (default) at factory. If user need to configure it as reset function pin, one should enable the corresponding option in STC-ISP Writer/Programmer (Shown as Figure 4.3, remove the " √ " in option " RESET pin used as I/O port "). Asserting an active-high signal and lasting at least 24 cycles plus 20μs on the RST pin will generate a reset. If the signal on RST pin turns to active-low level, MCU will be withdrew from reset process and start to run from system ISP Monitor Program Section. MCU will be soft reset to User Program Section and start to run user program when legal ISP download command flow is not detected.

The principle and circuit of reset are the same as the traditional 8051 MCU, as shown in Figure 4.5.

（1）内置上电复位电路（MAX810 专用复位电路）

当电源电压低于掉电/上电复位检测门槛电压时，所有的逻辑电路都会复位。当内部 V_{CC} 上升到复位门槛电压以上，延迟 8192 个时钟，掉电复位/上电复位结束。

若 MAX810 专用复位电路在 ISP 编程时可用，则掉电复位/上电复位结束后产生约 180ms 复位延迟，复位才能被解除。

（2）外部 RST 引脚复位。外部 RST 引脚复位就是从外部向 RST 引脚施加一定宽度的高电平复位脉冲，从而实现单片机的复位。P5.4（RST）引脚出厂时被设置为 I/O 口，要将其配置为复位引脚，要在 STC-ISP 编程时设置（如图 4.3 所示，将 " 将复位引脚作为 I/O 口 " 前面方框中的 " √ " 去掉即可）。将 RST 引脚拉高并维持至少 24 个时钟加 20μs 后，单片机进入复位状态，将 RST 引脚拉回低电平，单片机结束复位状态并从系统 ISP 监控程序区开始执行程序，如果检测不到合法的 ISP 下载命令流，将软复位到用户程序区执行用户程序。

复位原理以及复位电路与传统的 8051 单片机的复位是一样的，如图 4.5 所示。

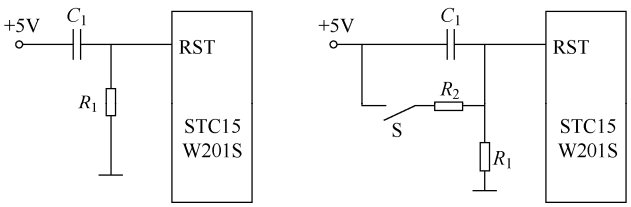


Figure 4.5 Reset circuit of MCU 单片机复位电路

(3) Watchdog Reset. The Watchdog circuit, which is usually used to monitor CPU, is a measure to improve the reliability of system. CPU will be considered working in an abnormal state and Watchdog circuit will reset CPU to force the system to start to run from User Program Area 0000H (when SWBS=0) or System ISP Monitor Area (when SWBS=1) when the CPU fails accessing Watchdog as required in a specific time. See section 4.1.7 for details.

(4) Software Reset. User may need to achieve MCU system soft reset (one of warm boot reset) in service sometimes. Due to the fact that hardware of traditional 8051 MCU does not support this feature, user has to use software to simulate it. Nowadays, STC15W201S series MCU achieves this function by using the ISP/IAP controlling register IAP_CONTR. User can control the 2 bits (SWBS/SWRST) in register to achieve system reset. The format of IAP_CONTR is showed in Table 4.6.

（3）看门狗复位。看门狗的基本作用就是监视 CPU 的工作。如果 CPU 在规定的时间内没有按要求访问看门狗，就认为 CPU 处于异常状态，看门狗就会强迫 CPU 复位，使系统重新从用户程序区（SWBS 为 0 时）0000H 处或系统 ISP 监控区（SWBS 为 1 时）开始执行用户程序，是一种提高系统可靠性的措施。详见 4.1.7 节。

（4）软件复位。在系统运行过程中，有时会根据特殊需求，实现单片机系统软复位（热启动复位之一）。传统的 8051 单片机由于硬件上未支持此功能，用户必须用软件模拟实现，实现起来较麻烦。STC15W201S 单片机利用 ISP/IAP 控制寄存器 IAP_CONTR 实现了此功能。用户只须简单地控制 ISP_CONTR 的其中两位（SWBS/SWRST）就可以实现系统复位了。IAP_CONTR 的格式如表 4.6 所示。

Table 4.6 IAP_CONTR

IAP_CONTR	Address 地址	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value 复位值
	C7H	IAPEN	SWBS	SWRST	CMD_FAIL	—	WT2	WT1	WT0	0000 x000

SWBS: Software Boot selection Control Bit.
0: Boot from User Program Section;
1: Boot from ISP Monitor Program Section.

SWRST: Software Rest Control.
0: Inactive;
1: Active, generate software system reset.

Run “IAP_CONTR=0x20;” to switch to User Program Section and start to run program.

Run “IAP_CONTR=0x60;” to switch to ISP Monitor Program Section and start to run program.

SWBS :软件复位程序启动区的选择控制位。（SWBS）＝0，从用户程序区启动；（SWBS）＝1，从 ISP 监控程序程序区启动。

SWRST ：软件复位控制位。（SWRST）＝0，不操作；（SWRST）＝1，产生软件复位。

若要切换到用户程序区起始处开始执行程序，执行“ IAP_CONTR= 0x20; ”语句；若要切换到 ISP 监控程序区起始处开始执行程序，执行“ IAP_CONTR= 0x60; ”语句。

2. Reset State

Except the boot section of program, the value of PC and the states of all specific function registers are the same, where PC=0000H, SP=07H, P1=P3=P5=FFH (see Table 4.7), both in cool boot reset and warm boot reset. The operation will not affect the status of on-chip RAM.

2. 复位状态

冷启动复位和热启动复位时，除程序的启动区域有所不同外，复位后 PC 值与各特殊功能寄存器的初始状态是一样的，具体见表 4.7。其中，PC=0000H, SP=07H, P1=P3=P5= FFH。复位不影响片内 RAM 的状态。

4.1.6 Power Management* 电源管理*

The normal working current of STC15W201S series MCU is 2.7mA~7mA when the V_{CC} is 5V. In order to lower the power consumption, STC15W201S series MCU can work in two low-power modes: idle and power-down mode. The typical value of working current is 1.8mA in idle mode and less than 0.1μA in power-down mode.

电源电压为 5V 时，STC15W201S 单片机的正常工作电流为 2.7～7mA。为了尽可能降低系统的功耗，STC15W201S 单片机可以运行在两种省电工作模式下：空闲模式和掉电模式。空闲模式下，STC15W201S 单片机的工作电流典型值为 1.8mA；掉电模式下，STC15W201S 单片机的工作电流小于 0.1μA。

1. Control of Idle Mode and Power-down Mode

Low-power mode is controlled by the corresponding bit in Power Control (PCON) register which is shown in Table 4.7.

1. 空闲模式与掉电模式的控制

省电工作模式的进入由电源控制寄存器 PCON 的相应位控制。PCON 寄存器的格式如表 4.7 所示。

Table 4.7 PCON

PCON	Address 地址	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value 复位值
	87H	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL	0011 0000

(1) PD: Enable this bit to place the STC15W201S series MCU into Stop Mode/ Power-down Mode (PCON=0x02). Only external interrupts keep on working in such mode, because the internal oscillator has stopped, and the CPU, Timers, UARTs and other function modules have stopped running. The MCU can be waked up by the external interrupt, either positive edge or negative edge, such pins as INT0, INT1, $\overline{\text{INT2}}$, $\overline{\text{INT3}}$ and $\overline{\text{INT4}}$.

（1）PD：置位 PD（PCON= 0x02），单片机将进入停机（掉电）模式。进入停机（掉电）模式后，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。进入停机（掉电）模式的单片机可由外部中断上升沿或下降沿触发唤醒，可将 CPU 从掉电模式唤醒的外部引脚主要有：INT0、INT1、$\overline{\text{INT2}}$、$\overline{\text{INT3}}$、$\overline{\text{INT4}}$。

(2) IDL: Enable this bit to place STC15W201S series MCU into Idle Mode (PCON=0x01). Modules still are active in Idle Mode except CPU. Any of external interrupts, timer interrupts and low-voltage detection interrupt can wake up MCU from Idle Mode.

(3) LVDF: Low-Voltage Detection Flag, and Low-Voltage Detection Interruption Request Flag.

Once low voltage condition is detected (V_{CC} power is lower than LVD voltage), it is set to 1 automatically both in normal mode and idle mode. It can be cleared by software.

If LVDF is disabled before accessing Power-down Mode, LVD circuit will stop working in Power-down Mode in order to reduce power consumption. If LVDF is enabled before accessing Power-down Mode, LVD circuit will stay working in Power-down Mode and it can wake up MCU by LVD interruption when V_{CC} is lower than LVD voltage.

(4) GF1: General-purposed Flag 1, user defined.

(5) GF0: General-purposed Flag 0, user defined.

2. Idle Mode (IDLE)

1）The working state of MCU in Idle Mode

In Idle Mode, only CPU is inactive. But the WDT whether works or not depends on the control bit in IDLE_WDT (WDT_CONTR.3). WDT is active when IDLE_WDT=1, while inactive when IDLE_WDT=0.

The registers, including RAM, Stack Pointer (SP), Program Counter (PC), Program Status Word (PSW), Accumulator (A) and all other registers will maintain their data during Idle Mode. The I/O ports hold the states they had at

（2）IDL：置位 IDL（PCON=0x01;），单片机将进入空闲模式。空闲模式时，除 CPU 不工作外，其余模块仍继续工作，可由外部中断、定时器中断、低压检测中断中的任何一个中断唤醒。

（3）LVDF：低压检测标志位，同时也是低压检测中断请求标志位。

在正常工作和空闲工作状态，如果内部电压 V_{CC} 低于低压检测门槛电压时，该位自动置 1。该位要用软件清零，清零后，如果内部工作电压 V_{CC} 继续低于低压检测门槛电压，该位又会被自动置 1。

在进入掉电工作状态前，如果低压检测中断未被允许，则在进入掉电模式后，低压检测电路不工作以降低功耗；如果低压检测中断被允许，则在进入掉电模式后，低压检测电路继续工作。在内部工作电压低于低压检测门槛电压后，产生低压检测中断，可将 MCU 从掉电状态唤醒。

（4）GF1：通用用户标志 1，用户可以任意使用。

（5）GF0：通用用户标志 0，用户可以任意使用。

2. 空闲模式（IDLE）

1）空闲模式下单片机的工作状态

STC15W201S 单片机在空闲模式下，除 CPU 不工作外，其余模块仍继续工作，但看门狗是否工作取决于 IDLE_WDT（WDT_CONTR.3）控制位。当 IDLE_WDT 为 1 时，看门狗正常工作；当 IDLE_WDT 为 0 时，看门狗停止工作。

在空闲模式下，RAM、堆栈指针（SP）、程序计数器（PC）、程序状态字（PSW）、累加器（A）等寄存器都保持

the time Idle Mode was active. The peripherals are still running in order to wake up the CPU when an interruption occurs.

2）Wake up MCU from Idle Mode

Activation of any interruption will clear IDL (PCON.0) by hardware to terminate the Idle Mode. And following RETI, the next instruction what is going to be executed will be the one following the instruction which drives the device into Idle Mode.

3. Power-down Mode

1）Working state in Power-down Mode

In Power-down mode, internal oscillator has stopped, and the CPU, Timers, UARTs and other function modules have stopped running while the external interrupts still work. If the low-voltage detection interrupt is enabled, LVD circuit works properly.

The I/O ports and specific functional registers hold the states they had at the time Power-down Mode was triggered.

2）Wake up from Power-down Mode

The external interruption INT0, INT1, $\overline{\text{INT2}}$, $\overline{\text{INT3}}$ or $\overline{\text{INT4}}$ can wake up CPU. And following RETI, the next instruction going to be executed will be the one following the instruction which has driven the device into Power-down Mode first, and then the corresponding Interrupt Service Routine (ISR). Thus, executing some more NOP instructions after accessing Power-down Mode is a recommendation.

If the Timer (T0, T2) interrupt is enabled before accessing Power-down Mode, when there is level change from high to low in the external pins of Timer (T0, T2), MCU will be waked up from Power-down Mode. After waked up by

原有数据，I/O 口保持着空闲模式被激活前那一刻的逻辑状态，所有的外围设备都能正常工作。

2）单片机空闲模式的唤醒

在空闲模式下，任何一个中断的产生都会引起 IDL（PCON.0）被硬件清零，从而退出空闲模式。单片机被唤醒后，CPU 将继续执行进入空闲模式语句的下一条指令。

3. 掉电模式

1）掉电模式下的工作状态

STC15W201S 单片机在掉电模式下，单片机所使用的时钟停振，CPU、看门狗、定时器、串行口、A/D 转换等功能模块停止工作，外部中断继续工作，如低压检测中断被允许，低压检测电路正常工作。

在掉电模式下，所有 I/O 口、特殊功能寄存器维持进入掉电模式前那一刻的状态不变。

2）掉电模式的唤醒

在掉电模式下，外部中断（INT0、INT1、$\overline{\text{INT2}}$、$\overline{\text{INT3}}$、$\overline{\text{INT4}}$）可唤醒 CPU。CPU 被唤醒后，首先执行设置单片机进入掉电模式的语句的下一条语句，然后执行相应的中断服务程序。为此，建议在设置单片机进入掉电模式的语句后多加几个 NOP 空指令。

如果定时器（T0、T2）中断在进入掉电模式前被设置为可用，则进入掉电模式后，定时器（T0、T2）外部引脚如发生由高到低的电平变化，就可以将单片机从掉电模式中唤醒。单

interruption, MCU will propagate clock to CPU after waiting 64 clock cycles. The next instruction going to be executed will be the one following the instruction which has driven the device into Power-down Mode when the CPU gained the clock, instead of the corresponding timer interruption program.

If the UARTs (UART1, UART2) interrupt is enabled before accessing Power-down Mode, when there is level change from high to low in the data receiver (RXD, RXD_2) of UART1, MCU will be waked up from Power-down Mode. MCU will propagate clock to CPU after waiting 64 clock cycles after waked up by interruption. The next instruction going to be executed will be the one following the instruction which has driven the device into Power-down Mode when the CPU gained the clock, instead of the corresponding UART interruption program.

If the built-in Power-down wake-up specific timer is enabled in STC15W201S series MCU, this timer starts counting once MCU accesses the Power-down Mode.

片机被唤醒后，在等待 64 个时钟后将时钟供给 CPU 工作。CPU 获得时钟后，程序从设置单片机进入掉电模式的下一条语句开始往下执行，不进入相应定时器的中断程序。

如果串行口（串行口 1、串行口 2）中断在进入掉电模式前被设置为可用，则进入掉电模式后，串行口 1 的数据接收端（RXD、RXD_2）如发生由高到低的电平变化时，就可以将单片机从掉电模式中唤醒。单片机被唤醒后，在等待 64 个时钟后将时钟供给 CPU 工作。CPU 获得时钟后，程序从设置单片机进入掉电模式的下一条语句开始往下执行，不进入相应串行口的中断程序。

如果 STC15W201S 单片机内置掉电唤醒专用定时器被设为可用，则当单片机进入掉电模式后，掉电唤醒专用定时器开始工作。

4．Application of Built-in Power-down Wake-up Specific Timer

Besides the external interruption, the built-in Power-down Wake-up Specific Timer is also used to wake up CPU. This kind of timer is suitable for periodic work scenario.

STC15W201S MCU is managed and controlled by specific functional registers WKTCH and WKTCL, which are defined in Table 4.8.

4．内部掉电唤醒专用定时器的应用

在单片机进入掉电模式后，除了可以通过外部中断进行唤醒外，还可以通过使能内部掉电唤醒专用定时器来唤醒 CPU，使其恢复到正常工作状态。内部掉电唤醒定时器的唤醒功能适合单片机周期性工作的应用场合。

STC15W201S 单片机由特殊功能寄存器 WKTCH 和 WKTCL 进行管理和控制，它们的定义如表 4.8 所示。

Table 4.8 WKTCH & WKTCL

	Address 地址	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value 复位值
WKTCL	AAH									11111111B
WKTCH	ABH	WKTEN								01111111B

Built-in Power-down Wake-up Timer is a timer of 15 bits, counting from 0. A 15 bits register composed of low 7 bits of WKTCH and 8 bits of WKTCL is used to set timer value.

WKTEN: Built-in Power-down Wake-up Timer enable bit. 1: Enable; 0: Disable.

Beside specific functional registers WKTCL and WKTCH, two covert registers WKTCL_ CNT and WKTCH_CNT are designed to control Built-in Power-down Wake-up Timer. WKTCL_ CNT and WKTCL occupy the same address (AAH), and WKTCH_CNT and WKTCH also occupy the same address (ABH). WKTCL_ CNT and WKTCH_CNT are invisible by user. In fact, WKTCL_CNT and WKTCH_CNT are used as counter, while WKTCH and WKTCL are used as comparator. The data will be written into WKTCH and WKTCL while data is read from WKTCH_CNT and WKTCL_ CNT when accessing the registers WKTCH and WKTCL.

Set WKTEN to enable Built-in Power-down Wake-up Timer. Once MCU gets into Power- down Mode, that timer [WKTCH_CNT, WKTCL _CNT] start to count from 7FFFH. When the counter reach the value set by the registers {WKTCH [6:0], WKTCL [7:0]}, system oscillator starts to work. The clock will be propagated to CPU after 64 cycles, and the next instruction going to be executed will be the

内部掉电唤醒定时器是一个 15 位定时器，定时后从 0 开始计数。WKTCH 的低 7 位和 WKTCL 的 8 位构成一个 15 位的数据寄存器，用于设定定时的计数值。

WKTEN：内部掉电唤醒定时器的使能控制位，（WKTEN）=1，使能；（WKTEN）=0，禁止。

STC15W201S 单片机除增加了特殊功能寄存器 WKTCL 和 WKTCH，还设计了两个隐藏的特殊功能寄存器 WKTCL_CNT 和 WKTCH_CNT 来控制内部掉电唤醒专用定时器。WKTCL_CNT 与 WKTCL 共享一个地址（AAH），WKTCH_CNT 与 WKTCH 共享一个地址（ABH）。WKTCL_CNT 和 WKTCH_CNT 是隐藏的，对用户是不可见的。WKTCL_CNT 和 WKTCH_CNT 实际上是作为计数器用，而 WKTCH 和 WKTCL 是作为比较器用。当用户对 WKTCH 和 WKTCL 写入内容时，该内容只写入 WKTCH 和 WKTCL 中；当用户读取 WKTCH 和 WKTCL 的内容时，实际上读的是 WKTCH_CNT 和 WKTCL_CNT 的内容，而不是 WKTCH 和 WKTCL 的内容。

置位 WKTEN，使能内部掉电唤醒定时器，当单片机一旦进入掉电模式，内部掉电唤醒专用定时器[WKTCH_CNT, WKTCL_CNT]就从 7FFFH 开始计数，直到计数到与{WKTCH[6:0], WKTCL[7:0]}寄存器所设定的计数值相等后就启动系统振荡器。单片机在等待 64 个时钟后，就将时钟供给 CPU，CPU 获得时钟后，

one following the instruction which puts the device into Power-down Mode when the CPU gained the clock. The values of WKTCH_CNT and WKTCL_CNT will maintain after wake-up from Power-down Mode, thus, the time spent in Power-down Mode can be obtained from reading the registers WKTCH and WKTCL (WKTCH_CNT and WKTCL_CNT actually).

程序从设置单片机进入掉电模式的下一条语句开始往下执行。掉电唤醒后，WKTCH_CNT 和 WKTCL_CNT 的内容保持不变，因此可通过读取 WKTCH 和 WKTCL 的内容（实际是 WKTCH_CNT 和 WKTCL_CNT 的内容）来读出单片机在掉电模式所等待的时间。

The counting of the timing of Power-down Wake-up Timer: the pulse period is about 488μs, so the timing is:

The value in registers {WKTCH [6:0], WKTCL [7:0]} plus 1 and then multiply by 488μs.

The minimum timing of timer is about 488μs.

The maximum timing of timer is about 488μs×32768 = 15.99s.

内部掉电唤醒定时器定时时间的计算：内部掉电唤醒定时器的计数脉冲周期大约为 488μs，那么定时时间为：

{WKTCH[6:0]，WKTCL[7：0]}寄存器的值加 1 再乘以 488μs。

内部掉电唤醒专用定时器的最短定时时间约为 488μs。

内部掉电唤醒专用定时器的最大定时时间约为 488μs×32768＝15.99s。

4.1.7 Watchdog Timer* 看门狗定时器*

1. Watchdog Timer

There always are some Electro Magnetic Interference (EMI) and program design problems in such high reliability electronic system as industrial control, automotive electronics, aerospace and so on. The system downtime will last long due to such problems. In order to discover and detach from downtime, a personal computer generally have a reset button, which is used to restart computer when the computer is down. In automation control system, it's not desirable to reset manually because of the reliability requirement. A certain circuit is always designed into the system to monitor the system, and reset the system when it is down. Such circuit is called Watchdog or Watchdog Timer. The Watchdog, whose basic role is to

1. 看门狗定时器

在工业控制、汽车电子、航空航天等需要高可靠性的电子系统中，由于电磁干扰的存在或者程序设计的问题，一般计算机系统都可能因程序跑飞或“死机”现象，导致系统长时间无法正常工作。为了及时发现并脱离瘫痪状态，在个人计算机中，一般具有复位按钮。当计算机死机时，可以按下复位按钮，重新启动计算机。在自动控制系统中，要求系统非常可靠稳定地工作，一般不能通过手工方式复位，往往需要在系统中设计一个电路自动地看护。当出现程序跑飞或死机时，迫使系统复位重新进入正常的工作状态。这个电路就称为硬件看门狗或看门狗定时器，简称看门狗。看门狗的基本作用就是监视 CPU 的工作。如果 CPU 在规定的时间内没有按要求访问看门狗，就认为 CPU 处于异常

monitor CPU and will consider if CPU is in an abnormal state or not and reset CPU to force the system to start to run the user program from the beginning by rule, when CPU fails accessing Watchdog as required in specific time. In normal working state, MCU sends pulses to Watchdog input pin via an I/O pin at regular time (the timing is not fixed, but less than the overflow time). The system will fail to send pulse when system is down, and the Watchdog will issue a reset signal after a certain time and reset system.

2. Watchdog Timer

A Watchdog Timer (WDT) is designed in STC15W201S series MCU, which makes system reliability design more convenient and concise. Register WDT_ CONTR is used for setting and controlling WDT, which is defined in Table 4.9.

状态，看门狗就会强迫 CPU 复位，使系统重新从头开始按规则执行用户程序。正常工作时，单片机可以通过一个 I/O 引脚定时向看门狗脉冲输入端输入脉冲（定时时间是不固定的，只要不超出硬件看门狗的溢出时间即可）。当系统出现死机时，单片机就会停止向看门狗脉冲输入端输入脉冲，超过一定时间后，硬件看门狗就会发出复位信号，将系统复位，使系统恢复正常工作。

2. 看门狗定时器

STC15W201S 单片机内部集成了看门狗定时器，使单片机系统的可靠性设计变得更加方便、简洁。通过设置和控制 WDT 控制寄存器（WDT_CONTR）来使用看门狗功能。WDT 控制寄存器的各位定义如表 4.9 所示。

Table 4.9 WDT_CONTR

WDT_CONTR	Address 地址	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value 复位值
	C1H	WDT_FLAG	—	EN_WDT	CLR_WDT	IDLE_WDT	PS2	PS1	PS0	0x00 0000

(1)WDT_FLAG: WDT overflow flag. Set by hardware when overflow, cleared by software.

(2)EN_WDT: WDT enable. 1, enable; 0, disable.

(3)CLR_WDT: Clear WDT when this register is set to “1”, WDT restart counting. Hardware will clear this register after boot.

(4)IDLE_WDT: IDLE mode setting. “1”, WDT will count in IDLE mode; “0”, WDT will not count in IDLE mode.

（1）WDT_FLAG：看门狗溢出标志位，溢出时，该位由硬件置 1，可用软件将其清零。

（2）EN_WDT：看门狗允许位，当设置为“ 1 ”时，看门狗启动。

（3）CLR_WDT：看门狗清零位，当设为“ 1 ”时，看门狗将重新计数。启动后，硬件将自动清零此位。

（4）IDLE_WDT：看门狗“ IDLE ”模式（即空闲模式）位，当设置为“ 1 ”时，WDT 在“ 空闲模式 ”时计数；当清零该位时，WDT 在“ 空闲模式 ”时不计数。

(5)PS2, PS1, PS0: WDT Prescaler factor setting.

WDT overflow time= (12×prescaler factor×32768)/frequency.

The relationship between WDT overflow time and prescaler factor setting when the frequency is 11.0592MHz, 12MHz and 20MHz is shown as Table 4.10.

（5）PS2、PS1、PS0：WDT 预分频系数控制位。

WDT 的溢出时间=(12×预分频系数×32768)/时钟频率。

时钟频率为 11.0592MHz、12MHz 和 20MHz 时，预分频系数设置与 WDT 溢出时间关系如表 4.10 所示。

Table 4.10 WDT Prescaler Factor and Overflow Time WDT 的预分频系数与溢出时间

PS2	PS1	PS0	Prescaler Factor 预分频系数	WDT（ms）		
				11.0592MHz	12MHz	20MHz
0	0	0	2	71.1	65.5	39.3
0	0	1	4	142.2	131.0	78.6
0	1	0	8	284.4	262.1	157.3
0	1	1	16	568.8	524.2	314.6
1	0	0	32	1137.7	1048.5	629.1
1	0	1	64	2275.5	2097.1	1250
1	1	0	128	4551.1	4194.3	2500
1	1	1	256	9.1022	8388.6	5000

4.2 Memory Organization 存储结构

STC15W201S series MCU has separate address space for Program Memory and Data Memory, and there are four individual partition memory banks inside: Program Memory (Program Flash), On-chip Scratch-pad RAM, On-chip Expanded RAM and EEPROM (Data Flash), as shown in Fiure 4.6.

STC15W201S 单片机存储结构的主要特点是程序存储器与数据存储器是分开编址的，STC15W201S 单片机内部在应用上有 4 个相互独立的存储器空间：程序存储器（程序 Flash）、片内基本 RAM、片内扩展 RAM 与 EEPROM（数据 Flash），如图 4.6 所示。

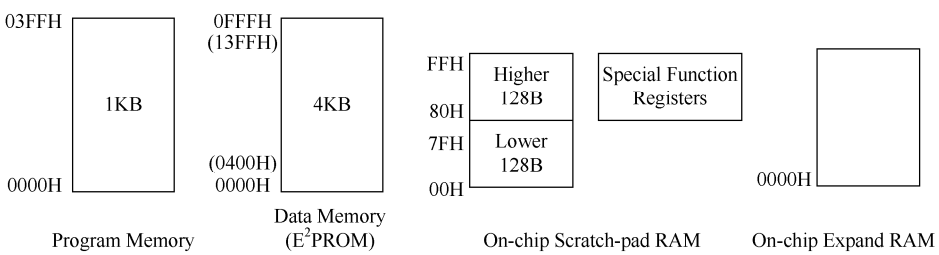


Figure 4.6 STC15W201S Memory Organization STC15W201S 单片机的存储结构

1. Program Memory (Program Flash)

Program Memory is the memory which stores such information as user program, data and table. 1KB flash memory is integrated in MCU, address from 0000H~03FFH.

There are some specific units in Program Memory, which should be taken into special consideration.

(1) 0000H unit. After reset, the value of PC is 0000H, from which CPU begins program execution. Generally, an unconditional transfer instruction is located in the 3 addresses beginning with 0000H, which will allow CPU to execute the program in the location user appoints.

(2) 0003H~00A3H, units use for the access address of 22 interrupt service (Interrupt Vector Address).

0003H: the access address for External Interrupt 0.

000BH: the access address for Timer/Counter0 (T0).

0013H: the access address for External Interrupt 1.

0023H: the access address for Serial Port Interrupt.

0033H: the access address for LVD Interrupt.

0053H: the access address for External Interrupt 2.

005BH: the access address for External Interrupt 3.

0063H: the access address for Timer/Counter2 (T2).

005BH: the access address for External Interrupt 4.

1. 程序存储器（程序 Flash）

程序存储器用于存放用户程序、数据和表格等信息。STC15W201S 单片机片内集成了 1KB 的程序 Flash 存储器，其地址为 0000H～03FFH。

在程序存储器中有些特殊的单元，在应用中应加以注意。

（1）0000H 单元。系统复位后，PC 值为 0000H，单片机从 0000H 单元开始执行程序。一般在 0000H 开始的三个单元中存放一条无条件转移指令，让 CPU 去执行用户指定位置的主程序。

（2）0003H～00A3H，这些单元作为 22 个中断的中断响应的入口地址（或称为中断向量地址）。

0003H：外部中断 0 中断响应的入口地址。

000BH：定时/计数器 0（T0）中断响应的入口地址。

0013H：外部中断 1 中断响应的入口地址。

0023H：串行口中断响应的入口地址。

0033H：LVD 中断响应入口地址。

0053H：外部中断 2 中断响应的入口地址。

005BH：外部中断 3 中断响应的入口地址。

0063H：定时/计数器 2（T2）中断响应的入口地址。

0083H：外部中断 4 中断响应的入口地址。

00ABH: the access address for Comparator Interrupt.

The interrupt service locations are spaced at an interval of 8 bytes. If an interrupt service routine is short enough, it can reside entirely within that 8-byte interval. Longer service routines can use a jump instruction to put in programing, which points to the interrupt service program, to skip over subsequent interrupt locations.

2. On-chip Scratch-pad RAM

There are lower 128 bytes and upper 128 bytes of on-chip scratch-pad RAM data memory plus 128 bytes of Special Function Register (SFR) space available in the scratch-pad RAM.

1）Lower 128 bytes

The lower 128 bytes of data memory may be divided into Register Bank, Bit-Addressable Bank and General Bank, according to the diversity of RAM function, as shown in Figure4.7.

(1) Register Bank (00H~1FH). The lowest 32 bytes of 8051 MCU internal RAM is divided into 4 banks with 8 registers each. When the program is running, only one bank serves as the current active register group which is named from R0 to R7. The RS0 and RS1 bits in PSW register select which register bank is in use. The relationship between status of RS1 and RS0 and current active register group is shown in Table 4.11.

00ABH：比较器中断响应入口地址。

每两个相邻中断向量间相隔 8 个存储单元。编程时，通常在这些入口地址开始处放入一条转移指令，指向真正存放中断服务程序的入口地址。只有在中断服务程序较短时，才可以将中断服务程序直接存放在相应入口地址开始的几个单元中。

2. 片内基本 RAM

片内基本 RAM 分为低 128 字节、高 128 字节和特殊功能寄存器（SFR）。

1）低 128 字节

低 128 字节根据 RAM 作用的差异性，又分为工作寄存器区、位寻址区和通用 RAM 区，如图 4.7 所示。

（1）工作寄存器区（00H～1FH）。8051 单片机片内基本 RAM 低端的 32 个字节分成 4 个工作寄存器组，每组占用 8 个单元。但程序运行时，只能有一个工作寄存器组为当前工作寄存器组。当前工作寄存器组的存储单元可作为寄存器，即用寄存器符号（R0，R1，…，R7）来表示。当前工作寄存器组的选择是通过程序状态字 PSW 中的 RS1、RS0 实现的。RS1、RS0 的状态与当前工作寄存器组的关系如表 4.11 所示。

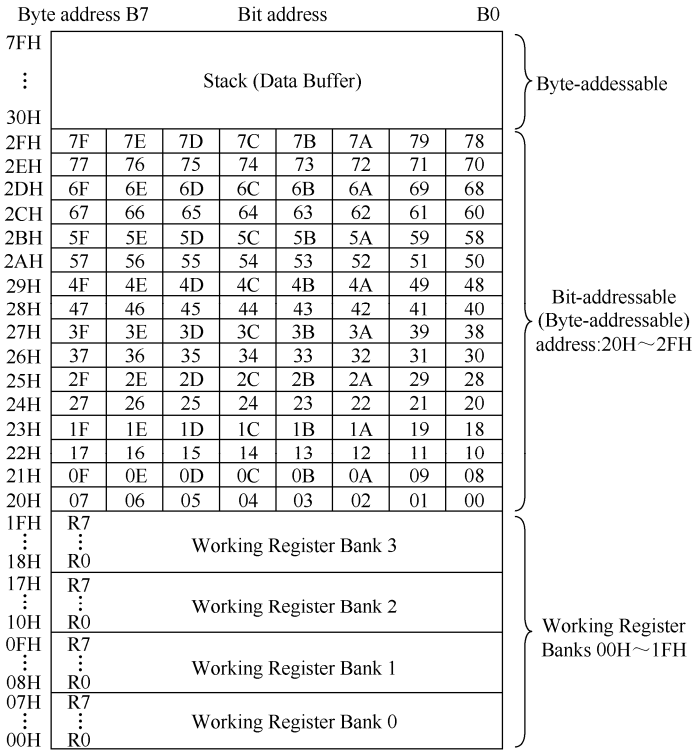


Figure 4.7　Lower 128 Bytes of Internal RAM　低 128 字节的功能分布图

Table 4.11　Address of Work Registers in 8051 MCU　8051 单片机的工作寄存器地址表

Bank	RS1	RS0	R0	R1	R2	R3	R4	R5	R6	R7
0	0	0	00H	01H	02H	03H	04H	05H	06H	07H
1	0	1	08H	09H	0AH	0BH	0CH	0DH	0EH	0FH
2	1	0	10H	11H	12H	13H	14H	15H	16H	17H
3	1	1	18H	19H	1AH	1BH	1CH	1DH	1EH	1FH

The data will be shielded when the current working register bank is switched to other register bank, which can be used for protection operation on site.

(2) Bit-addressable Memory. The next 16 bytes (20H~2FH) of RAM is bit-addressable memory space, which is 128 bits in total. That space is both bit-addressable and byte-addressable. The bit B0 of 20H to bit B7 of 2FH are corresponding to the address from 00H to 7FH of the RAM. The bit address can also consist

当前工作寄存器组从某一工作寄存器组切换到另一个工作寄存器组，原来工作寄存器组的各寄存器的内容将被屏蔽保护起来。利用这一特性可以方便地完成快速现场保护任务。

（2）位寻址区（20H～2FH）。片内基本 RAM 的 20H～2FH 共 16 个字节是位寻址区，每个字节 8 位，共 128 位。该区域不仅可按字节寻址，也可进行位寻址。从 20H 的 B0 位到 2FH 的 B7 位，其对应的位地址依次为 00H～7FH。位地址还可用字节地址加位号表示，如 20H

of byte address plus bit number, for example, the B5 in 20H could both be denoted by 05H and 20H.5.

单元的 B5 位，其位地址可用 05H 表示，也可用 20H.5 表示。

NOTE: The representation byte address plus bit number is always used in programming.
特别提示：编程时，一般用字节地址加位号的方法表示。

(3)General RAM (30H~7FH). The 80 bytes General RAM is a common use RAM space, which is always used for data buffer, such as display buffer, and stack as well.

（3）通用 RAM 区（30H～7FH）。30H～7FH 共 80 个字节为通用 RAM 区，即为一般 RAM 区域，无特殊功能特性。一般作为数据缓冲区用，如显示缓冲区。通常也将堆栈设置在该区域。

2）The Upper 128 bytes

2）高 128 字节

The addresses of upper 128 bytes are 80H~FFH, the same as SFR space. Different access methods, the Upper 128 bytes can only be accessed by indirect addressing while SFR can only be accessed by direct addressing, are used for separating these two different spaces. Besides, the Upper 128 bytes can also be used for stack.

高 128 字节的地址为 80H～FFH，属普通存储区域，但高 128 字节地址与特殊功能寄存器区的地址是相同的。为了区分这两个不同的存储区域，访问时，规定了不同的寻址方式，高 128 字节只能采用寄存器间接寻址方式访问；特殊功能寄存器只能采用直接寻址方式。此外，高 128 字节也可作为堆栈区。

3）Special Function Register (SFR) (80H~FFH)

3）特殊功能寄存器 SFR（80H～FFH）

The addresses of SFR rang from 80H to FFH, but there are only 37 significant addresses. That means there are only 37 SFRs in STC15W201S series MCU. The SFR is related to certain hardware interface circuit, which is either a reflection of a certain hardware interface circuit or determination the working state of a hardware circuit. The management and control of inner I/O interface circuit is accomplished by the operation and management of SFRs. According to their store characteristic, there are 2 types SFRs: bit-addressable and not-bit-addressable SFRs. The bit-addressable SFRs are those whose address ends with 0H or 8H. There is bit address for the bit-addressable space which is

特殊功能寄存器的地址也为 80H～FFH，但 STC15W201S 单片机中只有 37 个地址有实际意义，也就是说实际上只有 37 个特殊功能寄存器。所谓特殊功能寄存器是指该 RAM 单元的状态与某一具体的硬件接口电路相关，要么反映了某个硬件接口电路的工作状态，要么决定着某个硬件电路的工作状态。单片机内部 I/O 接口电路的管理与控制就是通过对其相应特殊功能寄存器进行操作与管理的。特殊功能寄存器根据其存储特性的不同又分为两类：可位寻址特殊功能寄存器与不可位寻址特殊功能寄存器。凡字节地址能够被 8 整除的特殊功能寄存器是可位寻址的，对应可寻址位都有一个位地址，其位地址等于其字节地址加上位号，实际编程时大多

byte address plus bit number. That bit address mostly is denoted by its function symbol such as CY, ACC and so on in PSW. The SFRs can only be accessed by direct addressing. The mapping of SFRs is shown in Table 4.12, as well as the symbol, address and reset values.

是采用其位功能符号表示，如 PSW 中的 CY、ACC 等。特殊功能寄存器与其可寻址位都是按直接地址进行寻址的。特殊功能寄存器的映象如表 4.12 所示，表中给出了各特殊功能寄存器的符号、地址与复位状态值。

NOTE: The address or bit address of SFRs are mostly denoted by the symbols of SFRs or bit address in programing of Assembly Language and C.

特别提示：实际使用汇编语言或 C 语言编程时，用特殊功能寄存器的符号或位地址的符号来表示特殊功能寄存器的地址或位地址。

Table 4.12 STC15W201S series MCU SFRs Byte Address and Bit Address

STC15W201S 单片机特殊功能寄存器字节地址与位地址

SFR	Symbol / Reset Value(B)	Address	Bit Address/Symbol							
P0	P0	80H	87H	86H	85H	84H	83H	82H	81H	80H
	1111 1111		P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
Stack Pointer	SP	81H	Point to stack							
	0000 0111									
Data Pointer Low	DPL	82H								
	0000 0000									
Data Pointer High	DPH	83H								
	0000 0000									
Power Control	PCON	87H	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL
	0011 0000									
Timer Control	TCON	88H	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
	XX00 0000		-	-	TF0	TR0	IE1	IT1	IE0	IT0
Timer Mode	TMOD	89H	-	-	-	-	GATE	C/$\overline{T}$	M1	M0
	XXXX 0000									
Timer Low 0	TL0	8AH								
	0000 0000									
Timer High 0	TH0	8BH								
	0000 0000									
Auxiliary Register	AUXR	8EH	T0x12	-	UART_M0x6	T2R	T2_C/$\overline{T}$	T2x12	-	S1ST2
	0X00 00X0									
Port 1	P1	90H	97H	96H	95H	94H	93H	92H	91H	90H
	XX11 1111		-	-	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
P1 Mode Configuration 1	P1M1	91H	-	-	P1M1.5	P1M1.4	P1M1.3	P1M1.2	P1M1.1	P1M1.0
	XX00 0000									

Continue

SFR	Symbol / Reset Value(B)	Address	Bit Address/Symbol							
P1 Mode Configuration 0	P1M0	92H	-	-	P1M0.5	P1M0.4	P1M0.3	P1M0.2	P1M0.1	P1M0.0
	XX00 0000									
Serial Control	SCON	98H	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H
	0000 0000		SM0	SM1	SM2	REN	TB8	RB8	TI	RI
Serial Buffer	SBUF	99H								
	XXXX XXXX									
Interrupt Enable	IE	A8H	AFH	AEH	ADH	ACH	ABH	AAH	A9H	A8H
	00X0 X000		EA	ELVD	-	ES	-	EX1	ET0	EX0
P3	P3	B0H	B7H	B6H	B5H	B4H	B3H	B2H	B1H	B0H
	11XX 1111		P3.7	P3.6	-	-	P3.3	P3.2	P3.1	P3.0
P3 Mode Configuration 1	P3M1	B1H	P3M1.7	P3M1.6	-	-	P3M1.3	P3M1.2	P3M1.1	P3M1.0
	00XX 0000									
P3 Mode Configuration 0	P3M0	B2H	P3M0.7	P3M0.6	-	-	P3M0.3	P3M0.2	P3M0.1	P3M0.0
	00XX 0000									
Interrupt Priority	IP	B8H	BFH	BEH	BDH	BCH	BBH	BAH	B9H	B8H
	X0X0 X000		-	PLVD	-	PS	-	PX1	PT0	PX0
Watch DogTimer Control	WDT_CONTR	C1H	-	-	EN_WDT	CLR_WDT	IDLE_WDT	PS2	PS1	PS0
	XX00 0000									
ISP/IAP Flash Data Register	IAP_DATA	C2H								
	0000 0000									
ISP/IAP Flash address High	IAP_ADDRH	C3H								
	0000 0000									
ISP/IAP Flash address Low	IAP_ADDRL	C4H								
	0000 0000									
ISP/IAP Flash Command Register	IAP_CMD	C5H	-	-	-	-	-	-	MS1	MS0
	XXXX XX00									
ISP/IAP Flash Command Trigger	IAP_TRIG	C6H								
	5AH/5AH									
ISP/IAP Control Register	IAP_CONTR	C7H	IAPEN	SWBS	SWRST	CMD_FALL	-	WT2	WT1	WT0
	0000 X000									
Port 5	P5	C8H	CFH	CEH	CDH	CCH	CBH	CAH	C9H	C8H
	XX00 XXXX		-	-	P5.5	P5.4	-	-	-	-

Continue

SFR	Symbol / Reset Value(B)	Address	Bit Address/Symbol							
P5 Mode Configuration 1	P5M1 / XX11 XXXX	C9H	-	-	P5M1.5	P5M1.4	-	-	-	-
P5 Mode Configuration 0	P5M0 / XX00 XXXX	CAH	-	-	P5M0.5	P5M0.4	-	-	-	-
Program Status Word	PSW / 0000 0000	D0H	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H
			CY	AC	F0	RS1	RS0	OV	F1	P
Timer 2 High 8-bit Register	T2H (RL_TH2) / 0000 0000	D6H								
Timer 2 Low 8-bit Register	T2L (RL_TL2) / 0000 0000	D7H								
ACC	Accumulator	E0H	E7H	E6H	E5H	E4H	E3H	E2H	E1H	E0H
			ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0
Comparator Control Register 1	CMPCR1 / 0000 0000	E6H	CMPEN	CMPIF	PIE	NIE	PIS	NIS	CMPOE	CMPRES
Comparator Control Register 2	CMPCR2 / 0000 1001	E7H	INVCMPO	DISFLT	LCDTY[5：0]					
			-	-	-	-	P4.3	P4.2	P4.1	P4.0
Register B	B / 0000 0000	F0H	F7H	F6H	F5H	F4H	F3H	F2H	F1H	F0H
			B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0

COMMENT: The shaded SFRs are new from traditional 8051 MCU. A declaration of the SFR address is required when it's in use. For example, the CMPCR2 is a new SFR, the declaration:

说明：带阴影的特殊功能寄存器为在传统 8051 单片机基础上新增的，在使用时须对各特殊寄存器的地址进行声明，例如，CMPCR2 是新增的特殊功能寄存器，声明如下：

C51： sfr CMPCR2 = 0xe7;

The SFRs relevant to common management are introduced here, and the other SFRs relevant to interface circuit will be left for later in the introduction of interface circuit.

(1) ACC, B, PSW: These 3 SFRs are relevant to CPU, which had been introduced in the introduction of CPU structure. Uses A for ACC in programing except PUSH and POP.

在此，介绍与公共管理相关的特殊功能寄存器，其他与各接口电路相关特殊功能寄存器留待各接口电路中学习。

（1）ACC、B、PSW：这 3 个特殊功能寄存器与 CPU 有关，已在 CPU 结构中介绍过。实际编程时，除 PUSH 和 POP 指令外，ACC 用 A 表示。

(2) SP: Stack Pointer, which always points to the top of stack. A stack is a storage area which follows the principle “first In, last out and last in, first out”. When pushing, the SP plus 1 and then the data will be pushed into the unit SP pointing to. When popping, the data will be popped to a specific memory from the unit SP pointing to and then SP minus 1. On reset, the SP is initialized to 07H causing the stack to begin at location 08H, which is also the first register (R0) of register bank 1. Thus, if more than one register bank is to be used, the SP should be initialized to a location in the data memory not being used for data storage. The stack area is used for storing interrupt, breakpoint address when calling the subroutine and field parameter.

(3) DPTR (16 bits): Dual Data Pointer Register (DPTR) consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16-bit address, which is used for accessing the program memory and data memory.

（2）SP：堆栈指针，它始终指向栈顶。堆栈是一种遵循“先进后出，后进先出”原则的存储区域。入栈时，SP 先加 1，数据再压入（存入）SP 指向的存储单元；出栈时，先将 SP 指向单元的数据弹出到指定的存储单元中，SP 再减 1。8051 单片机复位时，SP 为 07H，即默认栈底是 08H 单元。实际应用中，为了避免堆栈区域与工作寄存器组、位寻址区域发生冲突，堆栈区域设置在通用 RAM 区域。堆栈区域主要用于存放中断或调用子程序时的断点地址和现场参数数据。

（3）DPTR（16 位）：数据指针，由 DPL 和 DPH 组成，用于存放 16 位地址，对 16 位地址的程序存储器和数据存储器进行访问。

3. Data Flash Memory (EEPROM)

The data flash memory in STC15W201S series MCU is 4KB, ranging from 0000H~0FFFH. The data flash, EEPROM, is used to store the parameters those are volatile and remains unchanged after Power-down.

The embedded EEPROM consists of 8 sections, 512 bytes each. It’s recommended that the data modified at the same time should be put into the same section, because the erasing operation bases on section. Just filling the target address and data into several SFR, and triggering the built-in ISP automation, user can easily erase, read and program the embedded flash. Please refer to Section 3.4 for details.

STC15W201S series MCU is accessible by MOVC instruction where the beginning

3. 数据 Flash 存储器（EEPROM）

STC15W201S 单片机的数据 Flash 存储器空间为 4KB，地址范围为：0000H～0FFFH。数据 Flash 存储器被当成 EEPROM，用来存放一些应用时需要经常修改，掉电后又能保持不变的参数。

STC15W201S 单片机的数据 Flash 存储器空间分为 8 个扇区，每个扇区 512 字节。数据 Flash 存储器的擦除操作是按扇区进行的，在使用时建议同一次修改的数据放在同一个扇区，不是同一次修改的数据放在不同的扇区。在程序中，用户可以对数据 Flash 存储器实现字节读、字节写与扇区擦除等操作，具体操作方法见 3.4 节。

STC15W201S 单片机的数据 EEPROM 还可以采用 MOVC 指令访问，当采用

section address is 0400H and the tail address of ending section is 13FFH.

MOVC 指令访问时 EEPROM 的起始扇区地址为 0400H，结束扇区尾地址为 13FFH。

4.3 Parallel I/O Ports 并行 I/O 口

1. Parallel I/O Ports and Working Modes

1）I/O Ports feature

STC15W201S series MCU owns 14 I/O ports (P1.0~P1.5, P3.0~P3.3, P3.6, P3.7, P5.4, P5.5).

2）I/O Ports modes

There are 4 modes in STC15W201S series MCU: Quasi-bidirectional (traditional 8051 MCU I/O mode), Push-pull Output, Input-only (Hi-Z) and Open-drain Output. The drive of each port can reach 20mA, but the working current of the entire of MCU must not exceed 100mA. The mode of each port is controlled by corresponding bits in registers PnM1 and PnM0 (n=1, 3, 5). For example, P1M1 and P1M0 are used for setting the mode of port P1, where P1M1.0 and P1M0.0 for P1.0, P1M1.5 and P1M0.5 for P1.5, and so on. All the ports should be working at quasi-bidirectional mode after reboot. The setting is shown in the Table 4.13.

1. 并行 I/O 口与工作模式

1）I/O 口功能

STC15W201S 单片机有 14 个 I/O 口（P1.0～P1.5、P3.0～P3.3、P3.6、P3.7、P5.4、P5.5）。

2）I/O 口的工作模式

STC15W201S 单片机的所有 I/O 口均有 4 种工作模式：准双向口（传统 8051 单片机 I/O 模式）、推挽输出、仅为输入（高阻状态）与开漏模式。每个 I/O 口的驱动能力均可达到 20mA，但单片机整个芯片最大工作电流不要超过 100mA。每个口的工作模式由 PnM1 和 PnM0（n＝1, 3, 5）两个寄存器的相应位来控制。例如，P1M1 和 P1M0 用于设定 P1 口的工作模式，其中 P1M1.0 和 P1M0.0 用于设置 P1.0 的工作模式，P1M1.5 和 P1M0.5 用于设置 P1.5 的工作模式，以此类推。设置关系如表 4.13 所示，STC15W201S 单片机上电复位后所有的 I/O 口均为准双向口模式。

Table 4.13 Setting of I/O Ports Working Modes I/O 口工作模式的设置

Configuration		I/O Ports Mode
PnM1[7:0]	PnM0[7:0]	
0	0	Quasi-bidirectional(traditional 8051 I/O port output); Sink current up to 20mA, pull-up current is 150μA to 270μA 准双向（传统 8051 单片机 I/O 模式）；灌电流可达 20mA，拉电流为 150～230μA
0	1	Push-pull output: strong pull-up output ,current can be up to 20mA; resistors need to be added to restrict current 推挽输出：灌电流、拉电流都可达 20mA，要外接限流电阻
1	0	Input only(high-impedance) 仅为输入（高阻）
1	1	Open drain, internal pull-up resistor should disabled and external pull-up resistor is needed 开漏模式：内部上拉电阻断开，使用时要外接上拉电阻

2. Structure of Parallel I/O Ports

As mentioned above, there are 4 modes in STC15W201S series MCU: Quasi-bidirectional (traditional 8051 MCU I/O mode), Push-pull Output, Input-only (Hi-Z) and Open-drain Output. The modes of port P1, P3, P3 are controlled by corresponding bits in registers PnM1 and PnM0 (n=1, 3, 5). Below mainly introduces the structure and principle in different mode of I/O ports.

1）Quasi-bidirectional Mode

The structure of I/O circuit in Quasi-bidirectional Mode is shown in Figure 4.8.

2．并行 I/O 口的结构

如前所述，STC15W201S 单片机的所有 I/O 口均有 4 种工作模式：准双向口（传统 8051 单片机 I/O 模式）、推挽输出、仅为输入（高阻状态）与开漏模式，由 PnM1 和 PnM0（n＝1，3，5）两个寄存器的相应位来控制 P1、P3、P5 端口的工作模式。下面介绍 STC15W201S 单片机的并行 I/O 口不同模式的结构与工作原理。

1）准双向口工作模式

准双向口工作模式下，I/O 口的电路结构如图 4.8 所示。

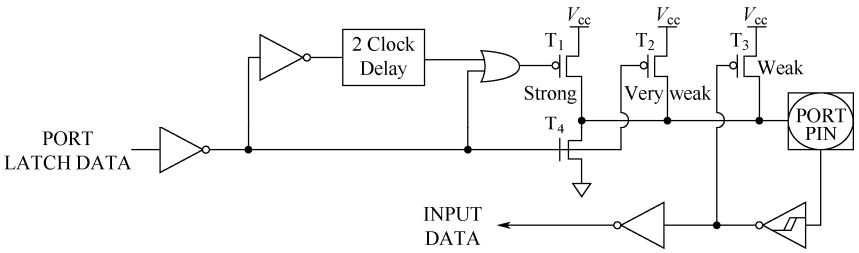


Figure 4.8　Structure of I/O Circuit in Quasi-bidirectional Mode

准双向口工作模式 I/O 口的电路结构

A quasi-bidirectional port can be used as an input or output port without the need to reconfigure the port. This is possible because when the port output logic high, and it is weakly driven, allowing an external device to pull the pin low. When the pin output low, it is driven strongly and able to sink a large current.

Each port contains an 8-bit register, which are SFR P1, P3 and P5. This structure can latch data before new data output to the port. But, the input data will be not latched. Thus the input data cannot change until a read instruction is executed.

There are 3 pull-up Field Effect Transistor T1, T2 and T3 in Quasi-bidirectional port to fit different

准双向口工作模式下，I/O 口可用直接输出而不用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，允许外部装置将其电平拉低。当引脚输出为低电平时，它的驱动能力很强，可吸收相当大的电流。

每个端口都包含一个 8 位锁存器，即特殊功能寄存器 P1、P3、P5。这种结构在数据输出时具有锁存功能，即在重新输出新的数据之前，口线上的数据一直保持不变。但对输入信号是不锁存的，所以外设输入的数据必须保持到读取指令执行为止。

准双向口有三个上拉场效应管 T1、T2、T3，以适应不同的需要。

requirements. T1 is referred to as the "strong" pull-up, whose pull-up current can reach 20mA. T2 is referred to as the "very weak" pull-up, whose pull-up current is 30μA normally. T3 is referred to as the "weak" pull-up, whose pull-up current is around 150 ~270μA, 200μA typically. The sink current can reach 20mA when the output is low-level.

T3 will be turned on when the port register for the pin contains logic "1" and the pin itself is also logic "1" level. T3 provides the primary source current for a quasi-bidirectional pin that is outputting "1". If this pin is pulled low by the external device, T3 will be turned off, and T2 remains on. In order to pull the pin low under these conditions, the external device has to sink enough current to pull the port pin below its input threshold voltage.

T2 will be turned on whenever the port register for the pin contains logic "1". This very weak pull-up sources a very small current that will pull the pin high if it is left floating.

T1 is used to speed up low-to-high transitions on a quasi-bidirectional port when the port register changes from logic "0" to logic "1". When this occurs, T1 will be turned on for 2 CPU cycles, pulling the port high quickly.

The quasi-bidirectional port contains a Schmitt-trigger input and an interference rejection circuit.

When there is data on the input pin, T4 stay in the cut-off state. If there once was "0" latched before data input, T4 was turned on, and the pin was clamped in low, which made high level cannot be inputted. Thus, in order to read data from port pin, the port register must be set to "1" first to make T4 cut-off.

其中，T1 称为“强上拉”，上拉电流可达 20mA；T2 称为“极弱上拉”，上拉电流一般为 30μA；T3 称为“弱上拉”，一般上拉电流为 150～270 μA，典型值为 200μA。输出低电平时，灌电流最大可达 20mA。

当口线寄存器为“1”且引脚本身也为“1”时，T3 导通。T3 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电平时，T3 断开，而 T2 维持导通状态。为了把这个引脚强拉为低电平，外部装置必须有足够的灌电流使引脚上的电压降到门槛电压以下。

当端口锁存器为“1”时，T2 导通。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。

当端口锁存器由“0”到“1”跳变时，T1 用来加快准双向口由逻辑“0”到逻辑“1”的转换。当发生这种情况时，T1 导通约两个时钟以使引脚能够迅速地上拉到高电平。

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。

当从端口引脚上输入数据时，T4 应一直处于截止状态。假定在输入之前曾输出锁存过数据“0”，则 T4 是导通的，这样引脚上的电位就始终被钳位在低电平，使输入高电平无法读入。因此，若要从端口引脚读入数据，必须先向端口锁存器置“1”，使 T4 截止。

COMMENT: Generally, parallel I/O works in quasi-bidirectional mode. Thus, it is recommended that the beginner should study quasi-bidirectional mode first.

说明：一般情况下，并行 I/O 口工作在准双向口工作模式，因此建议初学者只学习准双向口模式，其他模式等后续使用时再学习。

2）Push-pull Output Mode

The structure of I/O circuit in Push-pull Output Mode is shown in Figure 4.9.

2）推挽输出工作模式

推挽输出工作模式下，I/O 口的电路结构如图 4.9 所示。

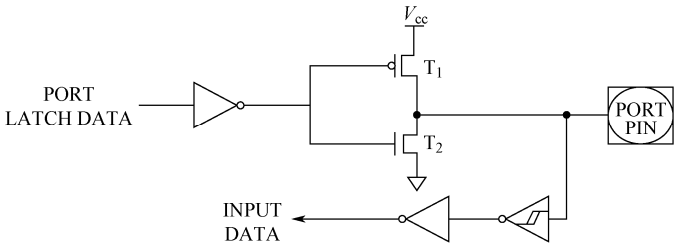


Figure 4.9　Structure of I/O Circuit in Push-pull Output Mode

推挽输出工作模式下 I/O 口的电路结构

The push-pull output has the same pull-down structure and input circuit structure with quasi-bidirectional mode, but provides a continuous strong pull-up when the port register contains logic “1”. The output pull current can be 20mA when output high level, while sink current 20mA when low level.

The port register must be set to “1” first to make T2 cut-off when data is inputted via pin.

推挽输出工作模式下，I/O 口输出的下拉结构、输入电路结构与准双向口模式是一致的，不同的是推挽输出工作模式下 I/O 口的上拉是持续的“强上拉”。若输出高电平，输出拉电流最大可达 20mA；若输出低电平，输出灌电流最大可达 20mA。

当从端口引脚上输入数据时，同准双向口模式一样，必须先向端口锁存器置“1”，使 T2 截止。

3）Input-only (Hi-Z) Mode

The structure of I/O circuit in Input-only Mode is shown in Figure 4.10.

3）仅为输入（高阻）工作模式

仅为输入（高阻状态）工作模式下，I/O 口的电路结构如图 4.10 所示。

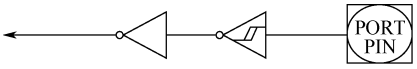


Figure 4.10　Structure of I/O Circuit in Input-only Mode

仅为输入（高阻状态）工作模式下 I/O 口的电路结构

The data can be read from port pin without setting the port register “1” first in this mode.

仅为输入（高阻状态）工作模式下，可直接从端口引脚读入数据，而不需要先对端口锁存器置“1”。

4）Open-drain Output Mode

The structure of I/O circuit in Open-drain Output Mode is shown in Figure 4.11.

The open-drain output mode has the same pull-down structure as both push-pull output mode and quasi-bidirectional mode, and the same input circuit as quasi-bidirectional mode, without any load

4）开漏工作模式

开漏工作模式下，I/O 口的电路结构如图 4.11 所示。

开漏输出工作模式下，I/O 口输出的下拉结构与推挽输出、准双向口一致，输入电路与准双向口一致，但输出驱动无任何负载，即开

in output, that is, open drain. A port pin must have an external pull-up in output application.

漏状态，输出应用时，必须外接上拉电阻。

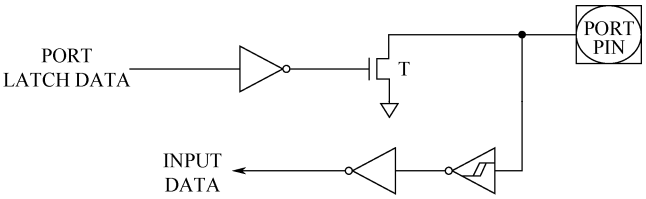


Figure 4.11 Structure of I/O Circuit in Open-drain Output Mode

开漏输出工作模式下 I/O 口的电路结构

3. Parallel I/O Port Application Notes

3. 并行 I/O 口的使用注意事项

1) Typical Triode Control Circuit

It is recommended that adopting MCU I/O port to control transistor to output if a high-power device is driven because of the limited drive of the I/O port. If the I/O is configured as "weak" pull-up, an external pull-up resistor R1 (3.3~10kΩ) should be added. If no pull-up resistor R1, it is recommended to add a resistor R2 at least 15 kΩ or configure I/O as "push-pull" mode, which is shown in Figure 4.12.

1) 典型三极管控制电路

单片机 I/O 引脚本身的驱动能力有限，如果需要驱动较大功率的器件，可以采用单片机 I/O 引脚控制晶体管进行输出的方法。如图 4.12 所示，如果用弱上拉控制，建议加上拉电阻 R1，阻值为：3.3～10 kΩ；如果不加上拉电阻 R1，建议 R2 的取值在 15kΩ 以上，或用强推挽输出。

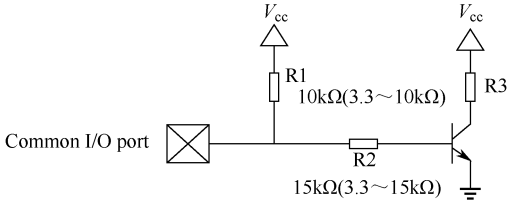


Figure 4.12 Typical Triode Control Circuit 典型三极管控制电路

2) Typical Diode Drive Circuit

For weak pull-up I/O, uses sink current drive LED, which is shown in Figure 4.13(a). For push-pull (strong pull-up) I/O, uses source current drive LED, which is shown in Figure 4.13(b).

2) 典型发光二极管驱动电路

采用弱上拉驱动时，采用灌电流方式驱动发光二极管，如图 4.13（a）所示；采用推挽输出（强上拉）驱动时，采用拉电流方式驱动发光二极管，如图 4.13（b）所示。

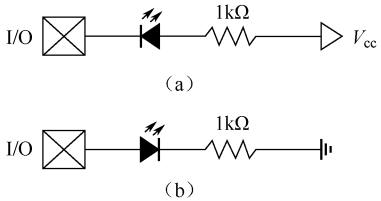


Figure 4.13 Typical Diode Drive Circuit 典型发光二极管驱动电路

In actual use, it is better to use sink current drive instead of source current drive, which can improve the load capability and reliability. In some special need, when the requirement of power supply circuit is simple, it can use source current.

There needs a current limiting resistor in key matrix scanning circuit. Because there will occur low level in 2 I/O ports and the keys shorted when it is pressed, while the output pins of CMOS circuit cannot be shorted. In key scanning circuit, one port must be set high before reading another one port. When the weak pull-up port changes from 0 to 1, there will be a 2-cycle strong push-pull output current outputting to the other low level I/O port, which will damage I/O port. Thus, it's recommended that two 300 limiting resistors are required on both side of the key scanning circuit, or make sure there will not be low level on both side of key I/O ports at the same time in software process.

3）How to Make I/O Port Low after Reset

The general I/O port in STC15W201S series MCU are weak pull-high output, while many practical applications require I/O port remain low level after power-on reset, otherwise the system malfunction would be generated. Here are 2 solutions for this problem.

(1) One of the methods is taking the function of logic reverse.

(2) For there are weak pull-up mode and strong push-pull mode in STC15W201S series MCU, a pull-down resistor (1kΩ/2kΩ/3kΩ) can be added to I/O port, so that when power-on reset, although a weak internal pull-up to make MCU output high, but because of the limit capacity of the internal pull-up, it cannot pull-high the pad, so

在实际使用时，应尽量采用灌电流驱动方式，而不要采用拉电流驱动，这样可以提高系统的负载能力和可靠性。有特别需要时，可以采取拉电流方式，如供电线路要求比较简单时。

设计行列矩阵按键扫描电路时，也需要加限流电阻。因为实际工作时可能出现两个 I/O 口均输出低电平的情况，并且在按键按下时短接在一起，而 CMOS 电路的两个输出脚不能直接短接在一起。在按键扫描电路中，一个口为了读另外一个口的状态，必须先置高。而单片机的弱上拉口在由 0 变为 1 时，会有两个时钟的强推挽输出电流，输出到另外一个输出低电平的 I/O 口，这样就有可能造成 I/O 口损坏。因此，建议在按键扫描电路中的两侧各加 300Ω 的限流电阻，或者在软件处理上，不要出现按键两端的 I/O 口同时为低电平的情况。

3）如何让 I/O 口上电复位时控制输出为低电平

STC15W201S 单片机上电复位时，普通 I/O 口为弱上拉高电平输出，而很多实际应用要求上电时某些 I/O 口控制输出为低电平，否则所控制的系统（如电动机）就会误动作。为了解决这个问题，有两种解决方法如下所示。

（1）通过硬件实现高、低电平的逻辑取反功能。

（2）由于 STC15W201S 单片机既有弱上拉输出模式又有强推挽输出模式，可在单片机 I/O 口上加一个下拉电阻（1kΩ、2kΩ 或 3kΩ），这样上电复位时，虽然单片机内部 I/O 口是弱上拉/高电平输出，但由于内部上拉能力有限，而外部下拉电阻又较小，无

this I/O port is low level after power-on reset. If this I/O port needs to be driven high, the MCU can be set as the push-pull output mode, while the push-pull mode the drive current can be up to 20mA, which can drive this I/O port high. In normal use, this port uses a series connection of 470 current limiting resistor and then connects the pull-down resistor to GND, which is shown in Figure 4.14.

法将其拉为高电平，所以该 I/O 口上电复位时外部输出为低电平。如果要将此 I/O 口驱动为高电平，可将此 I/O 口设置为强推挽输出，此时，I/O 口驱动电流可达 20mA，故可以将该口驱动为高电平输出。实际应用时，先串一个大于 470Ω 的限流电阻，再接下拉电阻到地，如图 4.14 所示。

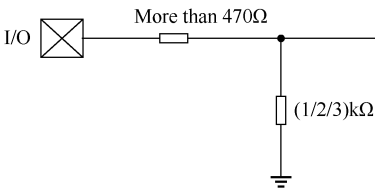


Figure 4.14 The Drive Circuit Making I/O Port Output Low Level After Power-on Reset

让 I/O 口上电复位时控制输出为低电平的驱动电路

Summary 本章小结

This chapter takes the typical STC15 series MCU —— STC15W201S for example to introduce enhanced 8051 MCU: enhanced CPU, memory and I/O port, emphasis on the internal memory and parallel I/O port. The internal memory consists of Program Flash, Data Flash and Scratch-Pad RAM. Program Flash stores program code and parameter. Scratch-Pad RAM is used for storing the parameter that is programmable and there will not be damage when Power-down. There are lower 128 bytes and upper 128 bytes of internal scratch-pad RAM data memory plus 128 bytes of Special Function Register (SFR) space available. The lower 128 bytes of data memory may be divided into Register Bank, Bit-Addressable Bank and General Bank. The upper 128 bytes is common data memory. SFRs have special functions and are always in connection with the internal interface circuit.

STC15W201S series MCU owns 14 I/O ports (P1.1~P1.5, P3.0~P3.3). According to the configurations, these ports P1, P3, P5 can work in the following 4 modes: Quasi-bidirectional, Push-pull Output, Input-only (Hi-Z) and Open-drain Output. The drive of each port can reach 20mA, but the working current of the entire of MCU must not exceed 100mA.

The master clock of STC15W201S series MCU has two clock sources: internal high precise R/C clock and external clock. The speed of system clock can be regulated dynamically by the configuration of clock division register. STC15W201S series MCU output master clock on pin P5.4. However it is MCKO_S1 and MCKO_S0 in CLK_DIV that determine whether output master clock or not, or what the division of clock is.

An internal specific reset circuit is integrated in STC15W201S series MCU, working properly without external reset circuit. There are 4 reset modes in STC15W201S series MCU: On-chip power-on/power-off reset, external RST pin reset, Watchdog Timer reset and software reset.

Exercise 练习题

1. Completions

1) Microcomputer comprises_____, _____, I/O interface and connecting bus.

2) Access and connecting to the CPU of microcomputer are accomplished by address bus, data bus, control bus and peripheral circuit, where, address bus is used for_____, the data size of address bus confines_____; data bus is used for_____, the data size of data bus confines_____; control bus is used for_____.

3) I/O interface is used for_____.

4) According to the storage property, the memory of microcomputer can be divided into _____and data storage.

5) 16 bits CPU means that_____ bus is 16 bits in width.

6) If the address bus is 16 bits in width, the maximum addressable capacity of CPU is_____.

7) The instructions executing of the microcomputer follows the storage order of instructions. The instructions executing comprises accessing, _____ and executing.

8) Microcomputer system comprises microcomputer and_____.

9) The data bus of STC15W201S MCU is_____ in width.

10) The address bus of STC15W201S MCU is_____ in width.

11) The drive capability of STC15W201S MCU I/O interface is_____.

12) Program Counter (PC) in STC15W201S MCU is used for_____, and its performance characteristic is_____.

13) PSW in STC15W201S MCU stands for_____, where, CY for_____, AC for_____, OV for_____, P for_____.

2. Choice Questions

1) That the data bus is 8 bits wide means the data exchange capability is_____.

A. 1 bit B. 4 bits C. 16 bits D. 8 bits

2) That the address bus is 8 bits wide means the maximum addressable capacity of CPU is_____.

A. 8 space B. 16 space C. 256 space D. 64K space

3) The program storage of microcomputer is comprised by_____.

A. ROM B. RAM

4) The data storage of microcomputer is comprised by_____.

A. ROM B. RAM

5) After the adder execution of 25H and 86H, the result in ACC is _____.

A. ABH B. 11H C. OBH D. A7H

6) After the adder execution of A0H and 65H, the value of CY and AC in PSW is _____.

A. 0 and 1 B. 1 and 0 C. 0 and 0 D. 1 and 1

7) After the adder execution of 58H and 38H, the value of OV and P in PSW is _____.

A. 0 and 1 B. 1 and 0 C. 0 and 0 D. 1 and 1

3. True or False

1) Keyboard is a fundamental component of microcomputer. ( )

2) I/O interface is the core part of microcomputer. ( )

3) I/O interface is the connecting bridge between CPU and I/O devices. ( )

4) CPU accesses memory and I/O devices by addressing. ( )

5) MCU is an important branch of microcomputer. ( )

6) 8 bits MCU and 32 bits ARM, they are both embedded microcomputer. ( )

7) The data loss will occur in RAM when it's power off. ( )

8) The data loss will not occur in ROM when it's power off. ( )

9) The Program Counter (PC) in CPU is the Special Function Register (SFR). ( )

10) The PSW in CPU is the Special Function Register (SFR). ( )

11) The Program Counter (PC) in CPU is an 8-bit adder. ( )

12) The maximum load capacity of STC15W201S MCU equal to the number of I/O times the load capacity of I/O. ( )

13) Power On Flag (POF) will be set to 1 after cold reset, and it will be set to 0 after warm reset. ( )

14) In power on reset process, CPU will execute programs from ISP, while from user program in other reset process. ( )

15) Apart from power port and ground port, other ports can be used as I/O port. ( )

4. Question and Answer

1) Please make a brief description of the fundamental structure of STC15W201S MCU.

2) Please make a brief description of the memory structure of STC15W201S MCU.

3) Please make a brief description of the working mode of STC15W201S MCU I/O port and their settings.

4) How to set up clocks in STC15W201S MCU.

5) How many reset modes are there in STC15W201S MCU? And how are they achieved? How is the status when STC15W201S MCU is under resetting?

Unit 5 Memory Application Program in STC15W201S Series MCU

第 5 单元 STC15W201S 单片机存储器的应用编程

As introduced in last chapter, there are individual memory partitions in STC15W201S series MCU: Program Memory (Program Flash), On-chip Scratch-pad RAM and EEPROM (Data Flash). The features of each memory partition and C51 application program will be introduced in this chapter.

前已介绍，STC15W201S 单片机存储器在应用上有以下相互独立的存储器空间：程序存储器（程序 Flash）、基本 RAM 与 EEPROM（数据 Flash），本单元主要学习各存储区域的存储特性与 C51 应用编程。

5.1 C51 Foundation C51 基础

5.1.1 Extended Key Words in Keil C51 Compiler Keil C51 编译器扩展的关键字

There are 32 key words in ANSI C standard definition, and 21 more key words extended in Keil C51 compiler, whose roles are shown in Table 5.1.

ANSI C 标准定义的关键字共有 32 个，Keil C51 编译器扩展了 21 个关键字，其作用如表 5.1 所示。

Table 5.1 Extended Key Words in Keil C51 Compiler

Keil C51 编译器扩展的关键字

Key Word	Type	Description
bit	Bit scalar declaration 位变量声明	Define a function of bit scalar or bit type 声明一个位变量或位类型函数
sbit	Searchable bit 可寻址位声明	Define a bit-searchable address variable 定义一个可位寻址变量的地址
sfr	Special function register 特殊功能寄存器声明	Define a SFR address (8 bits) 定义一个特殊功能寄存器（8 位）的地址
sfr16	Special function register 特殊功能寄存器声明	Define a SFR address (16 bits) 定义一个 16 位特殊功能寄存器的地址

Continue

Key Word	Type	Description
data	Memory type declaration 存储器类型说明	Internal direct addressing memory of 8051 MCU 直接寻址的 8051 单片机内部数据存储器
bdata	Memory type declaration 存储器类型说明	Internal bit addressing memory of 8051 MCU 可位寻址的 8051 单片机内部数据存储器
idata	Memory type declaration 存储器类型说明	Internal indirect addressing memory 间接寻址的 8051 单片机内部数据存储器
pdata	Memory type declaration 存储器类型说明	Internal paging addressing memory 分页寻址的片外数据存储器
xdata	Memory type declaration 存储器类型说明	External data memory 片外数据存储器
code	Memory type declaration 存储器类型说明	Program memory 程序存储器
interrupt	Interruption function declaration 中断函数声明	Define an interruption function 定义一个中断函数
reetrant	Reentrant functions declaration 再入函数声明	Define a reentrant function 定义一个再入函数
using	Register array declaration 寄存器组定义	Define register array in 8051 MCU 定义单片机使用的工作寄存器组
small	Variable storage mode 变量的存储模式	Variables which are not specified storage area are stored in *data* section 未指明存储类型的变量分配在 *data* 区域
large	Variable storage mode 变量的存储模式	Variables which are not specified storage area are stored in *xdata* section 未指明存储类型的变量分配在 *xdata* 区域
compact	Variable storage mode 变量的存储模式	Variables which are not specified storage area are stored in *pdata* section 未指明存储类型的变量分配在 *pdata* 区域
_at_	Address declaration 地址定义	Define the absolute address of variables 定义变量的绝对地址
far	Memory type declaration 存储类型声明	Accessing to extend RAM 某些单片机扩展 RAM 的访问
alicn	Function external declaration 函数外部声明	C functions which call PL/M-51 should be declared by *alicn* first 函数调用 PL/M-51，必须先用 *alicn* 声明
task	Support RTX51 支持 RTX51	Designate a task as a real-time task 指定函数是一个实时任务
priority	Support RTX51 支持 RTX51	Priority of task 指定任务的优先级

5.1.2 Memory Distribution of 8051 Series MCU
8051 系列单片机存储器的分配

The memory distribution of 8051 series MCU is implemented by the storage class of variable, and specific address is not required here.

8051 系列单片机存储器的分配是通过定义变量的存储类型来实现的，一般不需要指定具体的地址。

1. Storage Classes of Variable

The definition format of variable in C51 is shown as below:

[Storage Classes] Data Type [Memory Type] Variable

1．变量的存储类型

在 C51 中对变量的定义格式为：

[存储种类] 数据类型 [存储器类型] 变量名表

1）Storage Classes of C51 Variable

There are four storage classes: auto, extern, static, register. Auto is the default type, which is used generally.

1）C51 变量的存储种类

变量的存储种类有 4 种，分别为：auto（自动）、extern（外部）、static（静态）、register（寄存器）。默认时为 auto，一般取默认状态。

2）Data Types of C51 Variable

For Keil C51 compiler, short type is the same as int type, and double type is the same as float type. The data types supported by Keil C51 compiler are shown in Table 5.2.

2）C51 变量的数据类型

对于 Keil C51 编译器来说，short 型与 int 型相同，double 型与 float 型相同。如表 5.2 所示为 Keil C51 编译器支持的数据类型。

Table 5.2 Data Types Supported by Keil C51 Compiler

Keil C51 编译器支持的数据类型

Data Type	Length	Range
unsigned char	1 byte	0～255
signed char	1 byte	－128～＋127
unsigned int	2 bytes	0～65535
signed int	2 bytes	－32768～＋32767
unsigned long	4 bytes	0～4294967295
signed long	4 bytes	－2147483648～＋2147483647
float	4 bytes	±1.175494E-38～±3.402823E+38
*	1～3 byte	Address of object
bit	1 bit	0 or 1

3）Memory Types of C51 Variable

The hardware structure of 8051 series MCU is supported fully by Keil C51 Compiler. The compiler can access each part of the hardware system and decide in which type of memory the

3）C51 变量的存储器类型

Keil C51 编译器完全支持 8051 系列单片机的硬件结构，可以访问其硬件系统的各个部分，对于各个变量可以准确地赋予其存储器类型，使之能够在单

variable exactly reside, which can locate the variable exactly. The memory types supported by Keil C51 compiler are shown in Table 5.3.

片机内准确定位。Keil C51 编译器支持的存储器类型如表 5.3 所示。

Table 5.3　Register Types Supported by Keil C51 Compiler

Keil C51 编译器支持的存储器类型

Register Type	Description
data	Direct access memory, the fastest access speed 直接访问数据存储器，访问速度最快
bdata	Bit addressing data registers, bits and bytes accessing mixed is available 位寻址区，可进行位和字节混合寻址
idata	Indirect access memory 间接访问数据存储器
pdata	Accessing paged external data memory 分页访问片外数据存储器
xdata	Accessing all external data memory 片外数据存储器
code	Accessing program memory 程序存储器

NOTE: No extended RAM or Off-chip extended RAM function exists in STC15W201S series MCU.

特别提示：STC15W201S 单片机无扩展 RAM 和片外 RAM 扩展功能。

E.g:　　例如：

```
1    auto    int     data    x ;
2            char    code    y=0x22 ;
```

In line 1, the storage class, data type and memory type of variable x are *auto*, *int* and *data* respectively.

第 1 行中，变量 x 的存储种类、数据类型、存储器类型分别为 auto、int、data。

In line 2, only data type and memory type are given to variable y, without storage class. When in practical use, the selection of “storage class” and “memory type” is optional, and the default option of storage class is *auto*. If the memory type selection is omitted, the storage section will be determined by the default memory type which is set by the SMALL, COMPACT or LARGE compilation mode of Keil C51 compiler. Initialization is allowed to go along with the definition of variable in C language, as line 2.

第 2 行中，变量 y 只定义了数据类型和存储器类型，未直接给出存储种类。在实际应用中，“存储种类”和“存储器类型”是可选项，默认的存储种类是 auto（自动）；如果省略存储器类型时，则按 Keil C51 编译器编译模式 SMALL、COMPACT、LARGE 所规定的默认存储器类型确定存储器的存储区域。C 语言允许在定义变量的同时给变量赋初值，如第 2 行中对变量的赋值。

2. Compilation Mode of Keil C51 Compiler and Default Memory Type

1）SMALL

Variable is defined in the Data Memory of 8051 MCU, directly addressable, with a fastest access speed. Besides, all objects, including stack, must be resided in internal data memory.

2）COMPACT

Variable is defined in the External Data Memory (pdata) with up to 256 bytes of external data. Access to variable is implemented by register indirect-addressing (MOVX @Ri). In this kind of compilation mode, the high 8bit of the variable address is determined by port P2. Hence, the parameter in the startup code (STARTUP.A51): PDATASTART and PDATALEN should be regulated properly at the same time. The address of port P2 should be located by control instrument PDATA while connecting with L51, which ensures the right 8bit address the port P2 requires.

3）LARGE

Variable is defined in the External Data Memory (xdata) and accessed by Data Pointer (DPTR). This kind of accessing mode is not so efficient, especially for the variable with two or more bytes, and affects the length of code much. That the symmetry operation of data pointer is not available is another inconvenience.

2. Keil C51 编译器的编译模式与默认存储器类型

1）SMALL

变量被定义在 8051 单片机的内部数据存储器（data）区中，直接寻址访问，因此对这种变量的访问速度最快。另外，所有的对象，包括堆栈，都必须嵌入内部数据存储器。

2）COMPACT

变量被定义在外部数据存储器（pdata）区中，外部数据段长度可达 256 字节。这时对变量的访问是通过寄存器间接寻址（MOVX @Ri）实现的。采用这种模式编译时，变量的高 8 位地址由 P2 口确定。因此，在采用这种模式的同时，必须适当改变启动程序 STARTUP.A51 中的参数：PDATASTART 和 PDATALEN，用 L51 进行连接时还必须采用控制命令 PDATA 来对 P2 口地址进行定位，这样才能确保 P2 口为所需要的高 8 位地址。

3）LARGE

变量被定义在外部数据存储器（xdata）区中，使用数据指针 DPTR 进行访问。这种访问数据的方法效率不高，尤其是对于 2 个或多个字节的变量，用这种数据访问方法对程序的代码长度影响非常大。另外一个不便之处是数据指针不能对称操作。

5.1.3 Operation and Control of Internal Interface in 8051 MCU 8051 系列单片机内部接口的操作与控制

As mentioned before, there are one-to-one correspondences between all the interfaces and the SFRs. The operations to internal interfaces of 8051 MCU are the ones to the corresponding SFRs or specific bits in SFRs. Therefore, an accurate address must be provided, and the address

如前所述，8051 系列单片机所有的接口与特殊功能寄存器有一一对应关系，对 8051 单片机内部接口的操作就是对与其对应的特殊功能寄存器或特殊功能寄存器位的操作。因此，操作时必须给出准确地址，为此，

definition of SFR variable must be defined correctly.

对特殊功能寄存器变量必须进行准确的地址定义。

1. Definition of SFR Variable

1. 特殊功能寄存器变量的定义

There are 21 SFRs in 8051 series MCU (NOTE: Compatible 8051 MCU or enhanced 8051 MCU owns more SFRs). The SFRs are resided in the in-chip RAM from 80H to FFH. In order to access these SFRs or specific bits in SFRs directly, C51 compiler expend the keywords *sfr*, *sfr16* and *sbit*, which can be used to define the "address" of those SFRs or specific bits in SFRs directly in C program.

8051 系列单片机有 21 个特殊功能寄存器（注：兼容型 8051 单片机或增强型 8051 单片机有更多的特殊功能寄存器），它们离散地分布在片内 RAM 的 80H～FFH 地址中。为了能直接访问这些特殊功能寄存器或特殊功能寄存器位，C51 编译器扩充了关键字 sfr、sfr16 和 sbit，利用这些关键字可以在 C 语言源程序中直接对特殊功能寄存器或特殊功能寄存器位进行“地址”定义。

1）The definition of 8bit address SFR

1）8 位地址特殊功能寄存器的定义

Definition format:

定义格式：

sfr SFR name = address constant;

sfr 特殊功能寄存器名＝地址常数；

E.g.

例如：

```
sfr P0=0x80;        /* The address of port P0 SFR is 80H   定义特殊功能寄存器 P0 口的地址为 80H */
```

NOTE：There is different signification between the definition of SFR and normal variable. In the definition of SFR, the assignment is necessary, which is used to define the SFR corresponding address of memory. While in the definition of normal variable, the assignment which is used to assign a value to the memory unit, is optional.

特别提示：特殊功能寄存器的定义与普通变量定义中的赋值，其意义是不一样的。在特殊功能寄存器的定义中，赋值是必须有的，用于定义特殊功能寄存器所对应的内存的地址（即分配存储地址）；而在普通变量的定义中，赋值是可选的，是对变量存储单元赋值。

E.g.

例如：

```
int i=0x22 ;
```

This statement, that x is an integer variable and the assignment provides a value 22H to x, is equal to the two statements below:

此语句为定义 x 为整型变量，同时对 x 进行赋值，即 x 变量的内容为 22H，其效果等同于如下两条语句：

```
int i ;
i = 0x22 ;
```

2）The definition of 16bit address SFR

2）16 位特殊功能寄存器变量的定义

In the latest enhanced 8051 MCU, SFRs is combined into 16 bits. Using keyword *sfr16* is an efficient way to access this kind of 16bit

在新一代的增强型 8051 单片机中，特殊功能寄存器经常组合成 16 位使用。为了有效地访问这种 16 位的特殊功能

SFR in definition. For example, the definition of counter/timer T2 in MCU is as below:

寄存器，可采用关键字 sfr16 进行定义。例如定义单片机的定时器/计数器 T2，就可用如下方法定义：

```
sfr16 T2＝0xCC；   /* Define T2，whose address are T2L＝CCH，T2H＝CDH
                      定义 T2，其地址为 T2L＝CCH，T2H＝CDH */
```

Where the T2 is the name of SFR, the lower byte address is given after the equal sign and the high byte address is the lower byte address plus 1. This kind of SFR definition is only used in those 16bit SFRs whose addresses are consecutive, and the lower byte address must be given in the definition.

这里 T2 为特殊功能寄存器名，等号后面是它的低字节地址，其高字节地址是低字节地址加 1。此定义仅适用于地址相邻的 16 位特殊功能寄存器，而且定义时，给出的一定是低字节地址。

3）The definition of bit variable in SFR

3）特殊功能寄存器中位变量的定义

In 8051 MCU programing, some bits of SFR are accessed so frequently that Keil C51 programmer provides a keyword *sbit*, which is used to define the bit-addressing variable in SFR. There are three types of definitions as below:

在 8051 单片机编程中，要经常访问特殊功能寄存器中的某些位，Keil C51 编译器为此提供了 sbit 关键字，利用 sbit 可以对特殊功能寄存器中的位寻址变量进行定义。定义方法有如下 3 种：

(1)sbit bit variable = bit address. This kind of definition supply bit variable with an absolute address which ranges from 80H to FFH.

（1）sbit 位变量名＝位地址。这种方法将位的绝对地址赋给位变量，位地址必须位于 80H～FFH 之间。

E.g.

例如：

```
sbit OV=0xd2;    /* Define bit variable OV (over-flow flag), its bit address is D7H
                    定义位变量 OV（溢出标志），其位地址为 D2H */
sbit CY=0xd7;    /* Define bit variable CY (carry flag), its bit address is D7H
                    定义位变量 CY（进位位标志），其位地址为 D7H */
```

(2)sbit bit variable = SFR name ^ bit position. Used in the definition of the bit variable of defined SFR. The bit position ranges from 0 to 7.

（2）sbit 位变量名＝特殊功能寄存器名＾位位置。适用已定义的特殊功能寄存器位变量的定义，位位置值为 0～7。

E.g.

例如：

```
sbit OV = PSW ^ 2;    /* Define bit variable OV (over-flow flag), it's the 2nd bit of PSW
                         定义位变量 OV（溢出标志），它是 PSW 的第 2 位*/
sbit CY = PSW ^ 7;    /* Define bit variable CY (carry flag), it's the 7th bit of PSW
                         定义位变量 CY（进位位标志），它是 PSW 的第 7 位*/
```

(3)sbit bit variable name = byte address^ bit position. The kind of definition bases on the address of SFR. Its address ranges from 80H to

（3）sbit 位变量名＝字节地址＾位位置。这种方法是以特殊功能寄存器的地址作为基址，其值位于 80H～

FFH, and its bit position ranges from 0 to 7.

FFH 之间，位位置值为 0～7。

E.g.

例如：

```
sbit OV = 0xd0 ^ 2;          /* Define bit variable OV (over-flow flag), indicating the address of SFR PSW is D0H, and OV is the 2nd bit in the unit of D0H  定义位变量 OV（溢出标志），直接指明了特殊功能寄存器 PSW 的地址为 D0H，OV 是 D0H 地址单元的第 2 位*/
sbit CY = 0xd0 ^ 7;          /* Define bit variable CY (carry flag), indicating the address of SFR PSW is D0H, and CY is the 7th bit in the unit of D0H  定义位变量 CY（进位位），直接指明了特殊功能寄存器 PSW 的地址为 D0H，CY 是 D0H 地址单元的第 7 位*/
```

NOTE：In practical use, sbit is usually used for connecting the function of parallel OUTPUT/INPUT pins and external interface.

特别提示：实际使用中，经常利用 sbit 将并行输入/输出引脚与外部引脚功能联系在一起。

E.g.

例如：

```
sbit OE= 0x80 ^ 0;
```

or

或

```
sbit OE= P0 ^ 0;
```

Where OE is a variable defined according to the function of MCU pin (strobe) and OE is the bit 0 in port P0 that all the operations to OE are to P0.0 actually.

OE 是根据该单片机引脚的作用（选通）而定义的一个变量名称，OE 是 P0 口的第 0 位，对 OE 操作就是对 P0.0 操作。

4）The definition of bit variable in bit addressable locations (20H~2FH)

4）位寻址区（20H～2FH）位变量的定义

The bit object that locates in the internal memory addressable bdata of 8051 MCU is called “Addressable Object”. Keil C51 programmer will put the abject into internal address area of 8051 MCU in programing.

当位对象位于 8051 单片机内部存储器的可寻址区 bdata 时，称之为“可位寻址对象”。Keil C51 编译器编译时会将对象放入 8051 单片机内部可位寻址区。

(1)Bit addressable variable definition.

（1）定义位寻址区变量。

E.g.

例如：

```
int bdata my_y = 0x20;       /* The memory type of variable my_y is defined as bdata. This variable will be put into bit addressable area in memory distribution, and given a value of 20H 定义变量 my_y 的存储器类型为 bdata，分配内存时，自然分配到位寻址区，并赋值为 20H*/
```

(2)Bit addressable bit variable definition.

（2）定义位寻址区位变量。

The keyword *sbit* can define a certain bit of bit addressable object.

sbit 关键字可以定义可位寻址对象中的某一位。

E.g.

例如：

```
sbit   my_ybit0 = my_y ^ 0 ;      /* Define the bit 0 of variable my_y as variable my_ybit0 定义位变量 my_y 的第 0 位地址为变量 my_ybit0 */
sbit   my_ybit15 = my_y ^ 15 ;    /* Define the bit 15 of variable my_y as variable my_ybit15  定义位变量 my_y 的第 15 位地址为变量 my_ybit15 */
```

The maximum of the bit position after the operator depends on the data type of corresponding base address. That is 0~7 for *char*, 0~15 for *int* and 0~31 for *long*.

操作符后面的位置的最大值取决于指定的基地址的数据类型，对于 char 来说是 0～7, 对于 int 来说是 0～15，对于 long 来说是 0～31。

2. Header file REG51.H

The definitions of SFRs and bit variables in SFRs of 8051 series MCU are included in the header file REG51.H in Keil C51 compiler. The symbols of valid SFRs and bit variables in SFR can be used directly when the header file is included in program.

2. 头文件 REG51.H

Keil C51 编译器包含了对 8051 系列单片机各特殊功能寄存器以及特殊功能寄存器位定义的头文件 REG51.H。在程序设计时只要利用包含指令将头文件 REG51.H 包含进来, 8051 单片机中有效的特殊功能寄存器以及特殊功能寄存器位的符号就可以直接使用了。

NOTE：New SFRs in enhanced 8051 MUC must be defined with ***sfr*** and ***sbit***.
特别提示：对于增强型 8051 单片机，新增特殊功能寄存器就需要用 sfr 和 sbit 定义。

REG51.H program detail:

```
/*-----------------------------------------------------------------------
REG51.H
Header file for generic 80C51 and 80C31 microcontroller.
Copyright (c) 1988-2002 Keil Elektronik GmbH and Keil Software, Inc.
All rights reserved.
-----------------------------------------------------------------------*/
#ifndef __REG51_H__
#define __REG51_H__
/* ----------Definition of SFR---------- */
sfr P0          = 0x80;
sfr P1          = 0x90;
sfr P2          = 0xA0;
sfr P3          = 0xB0;
sfr PSW         = 0xD0;
sfr ACC         = 0xE0;
sfr B           = 0xF0;
sfr SP          = 0x81;
sfr DPL         = 0x82;
sfr DPH         = 0x83;
sfr PCON        = 0x87;
sfr TCON        = 0x88;
sfr TMOD        = 0x89;
sfr TL0         = 0x8A;
sfr TL1         = 0x8B;
sfr TH0         = 0x8C;
```

```
sfr TH1         = 0x8D;
sfr IE          = 0xA8;
sfr IP          = 0xB8;
sfr SCON        = 0x98;
sfr SBUF        = 0x99;

/* ----------Definition of bit variables in SFRs ---------- */
/* ------------ PSW----------      */
sbit CY         = 0xD7;
sbit AC         = 0xD6;
sbit F0         = 0xD5;
sbit RS1        = 0xD4;
sbit RS0        = 0xD3;
sbit OV         = 0xD2;
sbit P          = 0xD0;

/* ---------- TCON-------    */
sbit TF1        = 0x8F;
sbit TR1        = 0x8E;
sbit TF0        = 0x8D;
sbit TR0        = 0x8C;
sbit IE1        = 0x8B;
sbit IT1        = 0x8A;
sbit IE0        = 0x89;
sbit IT0        = 0x88;

/* --------- IE------------      */
sbit EA         = 0xAF;
sbit ES         = 0xAC;
sbit ET1        = 0xAB;
sbit EX1        = 0xAA;
sbit ET0        = 0xA9;
sbit EX0        = 0xA8;
/* ---------- IP -----------    */
sbit PS         = 0xBC;
sbit PT1        = 0xBB;
sbit PX1        = 0xBA;
sbit PT0        = 0xB9;
sbit PX0        = 0xB8;
/* --------- P3 ----------- */
sbit RD         = 0xB7;
sbit WR         = 0xB6;
sbit T1         = 0xB5;
sbit T0         = 0xB4;
```

```
sbit INT1       = 0xB3;
sbit INT0       = 0xB2;
sbit TXD        = 0xB1;
sbit RXD        = 0xB0;
/* ------------ SCON--------    */
sbit SM0        = 0x9F;
sbit SM1        = 0x9E;
sbit SM2        = 0x9D;
sbit REN        = 0x9C;
sbit TB8        = 0x9B;
sbit RB8        = 0x9A;
sbit TI         = 0x99;
sbit RI         = 0x98;

#endif
```

5.1.4 C51 Interrupt Function C51 中断函数

Compiling interrupt service function directly in C program is supported by C51 compiler. In order to compiling interrupt service function, an expend keyword *interrupt* is added to the definition of function by C51 compiler. The keyword *interrupt* is an option in function definition. A function will become an interrupt service function with this keyword.

C51 编译器支持在 C 语言源程序中直接编写中断服务函数。为了能够在 C 语言源程序中直接编写中断服务函数，C51 编译器对函数的定义进行了扩展，增加了一个扩展关键字 interrupt。关键字 interrupt 是函数定义时的一个选项，加上这个选项就可以将一个函数定义成中断服务函数。

1. Definition of Interrupt Service Function

1．中断服务函数的定义

The form of interrupt service function definition is just as below:

中断服务函数定义的一般形式如下：

```
Function Type    Function Name (Formal Parameter)    interrupt n    [using m]
函数类型          函数名（形式参数表）
```

Where the *n* after keyword *interrupt* is the interruption number, which ranges from 0 to 31. The compiler will generate interrupt vector at 8*n*+3, whose *n* and interrupt vector depend on different MUC chip. The keyword *using* is optional, and it's used for register bank selection. The *m* indicates the corresponding register bank number. The value of *m* is 0~3, which corresponds register bank 0~3 in 8051 MCU.

其中，关键字 interrupt 后面的 *n* 是中断号，*n* 的取值范围为 0～31。编译器从 8*n*＋3 处产生中断向量，具体的中断号 *n* 和中断向量取决于不同的单片机芯片。关键字 using 是可选项，用于选择工作寄存器组，*m* 为对应的寄存器组号，*m* 取值为 0～3，对应 8051 单片机的 0～3 寄存器组。

2. Interrupt Request Source

The interrupt sources and vector addresses of 8051 MCU are shown in Table 5.4.

2. 中断源

8051 单片机中断源的中断号与中断向量如表 5.4 所示。

Table 5.4 Interrupt Sources and Vector Addresses of 8051 MCU

8051 单片机的中断源与中断向量表

Interrupt Source 中断源	Interrupt Number 中断号	Interrupt Vector 中断向量地址
External Interruption 0 外部中断 0	0	0003H
Timer/Counter Interruption 0 T0 中断	1	000BH
External Interruption 1 外部中断 1	2	0013H
Timer/Counter Interruption 1 T1 中断	3	001BH
Serial Port Interruption 串行口中断	4	0023H

3. Rules of Interrupt Service Function Programing

(1) It's unable to pass parameter in interrupt function. A compile error occurs when there is any parameter definition in interrupt function.

(2) There is no return in interrupt function. An error result will occur when a return is defined in interrupt function. Thus, it's the best way that defines the interrupt function as *void* type with the purpose of indicating no return is required.

(3) Calling interrupt function directly is not allowed in any case, or a compile error will occur. Because 8051 MCU instruction RETI, which implements the return of interrupt function, influences the hardware interrupt system of 8051 MCU.

(4) The state of floating point register must be preserved if floating point arithmetic is executed in interrupt function, while the preservation is not required when there is no other program executing floating point arithmetic.

(5) The function called by interrupt function must use the same register bank with interrupt function. User must ensure the same register bank according to the requirement, or some unexpected error will occur. Compiler will choose one register bank as the absolute register bank to access, when the *using* option is not used in function definition.

3. 中断服务函数的编写规则

(1) 中断函数不能进行参数传递，中断函数中包含任何参数声明都将导致编译出错。

(2) 中断函数没有返回值，如果企图定义一个返回值将得到不正确的结果。因此，定义中断函数时最好将其定义为 void 类型，以明确说明没有返回值。

(3) 在任何情况下都不能直接调用中断函数，否则会产生编译错误。因为中断函数的返回是由 8051 单片机指令 RETI 完成的，RETI 指令影响 8051 单片机的硬件中断系统。

(4) 如果中断函数中用到浮点运算，必须保存浮点寄存器的状态，当没有其他程序执行浮点运算时可以不保存。

(5) 如果在中断函数中调用了其他函数，则被调用函数所使用的寄存器组必须与中断函数相同。用户必须保证按要求使用相同的寄存器组，否则会产生不正确的结果。如果定义中断函数时没有使用 using 选项，则由编译器选择一个寄存器组作为绝对寄存器组访问。

5.2 Program Memory 程序存储器

Program memory is used for storing user programs, and has MCU to run under the rule and flow designated by user program to implement the task. Besides, program memory stores some constants and tables (such as π, digital word and so on) which can be used in the running of user program. Such constants are downloaded to the program memory like programs by ISP. The content in data memory is read-only when the program is running. The constants and tables can be accessed only by such instructions as "MOVC A, @A+DPTR" or "MOVC A, @A+PC". If C language programing is adopted, the storing type of the data storing in program memory should be defined as "CODE". Take the display control of 6 LED light for example to explain the application program of program memory.

程序存储器的主要作用是存放用户程序，使单片机按用户程序指定的流程与规则运行，完成用户指定的任务。除此以外，程序存储器通常还用来存放一些常数或表格数据（如π值、数码显示的字形数据等），供用户程序在运行中使用。这些常数像程序一样通过 ISP 下载并存放在程序存储器区域。在程序运行过程中，程序存储器的内容只能读取，而不能写入。存于程序存储器中的常数或表格数据只能采用"MOVC A，@A+DPTR"或"MOVC A，@A+PC"指令进行访问。若采用 C 语言编程，要把存放在程序存储器中的数据存储类型定义为"CODE"。以 6 只 LED 灯的显示控制为例，说明程序存储器的应用编程。

E.g.

Given that port P1 drives these six LEDs, active low. Export the six groups' data "1EH, 0CH, 00H, 21H, 33H, 3FH" in order circularly.

例

设 P1 口驱动 6 只 LED，低电平有效。从 P1 口顺序输出"1EH、0CH、00H、21H、33H、3FH"6 组数据，周而复始。

Solution: First of all, put all the six groups' data into program memory, and then adopt data array and define the data as "CODE" storage type.

This gives the following code in C51:

解：首先将这 6 组数据存放在程序存储器中，再采用数组并定义为"CODE"存储类型。

C51 参考程序如下：

```
#include <REG51.H>
#define uchar unsigned char
#define uint unsigned int
uchar code date[ ] = {0x1e,0x0c,0x00,0x21,0x33,0x3f};
                                        // Define display data    定义显示数据
/*————————————Delay Subroutine(延时子函数)————————————*/
void delay(uint k)                      // Define delay subroutine    定义延时子函数
{
```

```
        uint i, j;
        for(i = 0; i<k; i++)
        {
                for(j=0; j<1210; j++)
                {;}
        }
}
/*————————————Main Function(主函数)————————————*/
void main(void)
{
        uchar i;
        while(1)                    /* Loop    无限循环*/
        {
              for(i = 0; i<6; i++)  /* Output data in order    顺序输出数据*/
                {
                    P1 = date[i];   /* Fetch the data in program memory  取存在程序存储器
                                       中的数据*/
                delay(50);          /* Set the interval of display. Subject to the frequency of
                                       crystal oscillator    设置显示间隔，晶振频率不同时，时间
                                       可能不一样，自行调整  */
                }
        }
}
```

5.3　Scratch-pad RAM　基本 RAM

There are lower 128 bytes (00H~7FH) and upper 128 bytes (80H~FFH) RAM data memory plus 128 bytes of Special Function Register (SFR) (80H~7FH) space available in the scratch-pad RAM of STC15W201S MCU.

1. The Lower 128 Bytes (00H~7FH)

The lower 128 bytes of data memory, which is considered as the data space “Nearest to CPU”, is the basic data memory in MCU. The lower 128 bytes of data memory may be accessed through

STC15W201S 单片机的基本 RAM 包括低 128 字节 RAM（00H～7FH）、高 128 字节 RAM（80H～FFH）和 128 字节特殊功能寄存器（80H～FFH）。

1．低 128 字节（00H～7FH）

低 128 字节是单片机最基本的数据存储区，可以说是“离单片机 CPU 最近”的数据存储区。整个 128 字节地址既可以直接寻址，也可以寄存器

both direct and indirect addressing. The lowest 32 bytes of data (00H~1FH) of data memory are grouped into register banks. The next 16 bytes (20H~2FH) form a block of bit-addressable memory space.

E.g.

Define variable x, adopting different addressing form, using the lower 128 bytes to store data, and value with 100.

Solution: In C51 programing, the data type of variable should be defined as "data" when accessing the low 128 byte with direct addressing; while as "idata" when with register indirect addressing. So:

```
unsigned char data x=100;
unsigned char idata x=100;
```

间接寻址。其中，00H～1FH 单元可以作为工作寄存器，20H～2FH 单元具有位寻址能力。

例

定义变量 x，采用不同的寻址方式，使用低 128 字节存储单元，并赋值 100。

解：在 C51 编程中，若采用直接寻址访问低 128 字节，则变量的数据类型定义为“ data ”；若采用寄存器间接寻址访问低 128 字节，变量的数据类型定义为“ idata ”，则：

2. The Upper 128 Bytes (80H~FFH) and SFR (80H~FFH)

The upper 126 bytes and SFR have the same address and address conflict occurs here. In practical use, the upper 128 bytes of data memory may only be accessed using indirect address while the SFR can only be accessed through direct addressing, which can distinguish both.

E.g.

Write data 20H into 80H unit in upper 128 bytes and SFR (p0) separately.

Solution: When using upper 128 bytes RAM to store data, the data type of variable should be defined as "idata", while using the register name directly to access SFR. So:

```
unsigned char idata x=0x20 _at_   0x80 ;
P0=0x20;
```

2．高 128 字节（80H～FFH）和特殊功能寄存器（80H～FFH）

高 128 字节（80H～FFH）和特殊功能寄存器（80H～FFH）的地址是相同的，也就是地址“ 冲突 ”了。在实际应用中，高 128 字节 RAM 只能用寄存器间接寻址进行访问，而特殊功能寄存器只能用直接寻址进行访问。

例

分别对高 128 字节 80H 单元和特殊功能寄存器 80H 单元（P0）写入数据 20H。

解：若要在 C51 编程中采用高 128 字节 RAM 存储数据，则在定义变量时，要将变量的存储类型定义为“ idata ”，而对特殊功能寄存器的操作是直接用寄存器名称进行存取即可，则：

5.4 EEPROM（Data Flash） EEPROM（数据闪存）

Internal EEPROM, which could be repeatedly erased more than 100 thousand times, can be used as DATA Flash by In-application Program(IAP) technology. When the program resides in ISP program section, user can impose byte read, byte write and page erase upon user program section and data flash section. When the program resides in user program section, user can only impose byte read, byte write and page erase upon data flash section. The embedded EEPROM consists of several pages. Each page contains 512 bytes.

STC15W201S 单片机的内部 EEPROM 作为数据闪存区是通过 IAP 技术实现的，内部闪存擦写次数达十万次以上。程序在系统 ISP 程序区时可以对用户程序区、数据闪存区进行字节读、字节写和扇区擦除操作；程序在用户程序区时，只可以对数据闪存区进行字节读、字节写和扇区擦除操作。EEPROM 可分为若干个扇区，每个扇区包含 512 字节。

1. Capacity and Address of Internal EEPROM

STC15W201S MCU has integrated a capacity of 4KB of internal EEPROM which is separated from program space with address ranging from 0000H to 0FFFH. There are 8 pages in EEPROM, 512 bytes each. The address of the first page is 0000H~01FFH, second page 0200H~03FFH and so on.

It's not only using IAP to erase pages but also MOVC instruction, but the head of address is 0400H which is the next address of where program memory ends instead of 0000H.

1. 内部 EEPROM 的大小与地址

STC15W201S 单片机共有 4KB EEPROM，且与程序存储空间是分开编址的，地址范围为：0000H～0FFFH，共分为 8 个扇区，每个扇区 512 字节。第一扇区的地址为 0000H～01FFH，第二扇区的地址为 0200H～03FFH，依此类推。

EEPROM 除可以用 IAP 技术擦除外，还可以用 MOVC 指令，但此时 EEPROM 的首地址不再是 0000H，而是程序存储器空间结束地址的下一个地址，即 0400H。

2. SFRs Related to ISP/IAP

STC15W201S MCU is managed and controlled by a set of SFRs. These SFRs related to ISP/IAP are shown in Table 5.5.

2. 与 ISP/IAP 功能有关的特殊功能寄存器

STC15W201S 单片机是通过一组特殊功能寄存器进行管理与控制的，各 ISP/IAP 特殊功能寄存器格式见表 5.5。

Table 5.5 SFRs related to ISP/IAP 与 ISP/IAP 功能有关的特殊功能寄存器

	Address	D7	D6	D5	D4	D3	D2	D1	D0	Reset Value
IAP_DATA	C2H									1111 1111
IAP_ADDRH	C3H									0000 0000
IAP_ADDRL	C4H									0000 0000
IAP_CMD	C5H	—	—	—	—	—	—	MS1	MS0	xxxx xx00
IAP_TRIG	C6H									xxxx xxxx
IAP_CONTR	C7H	IAPEN	SWBS	SWRST	CMD_FAIL	—	WT2	WT1	WT0	0000 x000

(1) IAP_DATA: ISP/IAP Flash Data Register.

It is the buffer register storing the data going to be written into Flash or read out from Flash in ISP/IAP.

(2) IAP_ADDRH, IAP_ADDRL: ISP/IAP Flash Address Register.

IAP_ADDRH is used for storing the high 8bit of the operation address, while IAP_ADDRL storing the low 8 bit.

(3) IAP_CMD: ISP/IAP Flash Command Register.

It's used for setting the operation instruction of ISP/IAP, which will not take effect until a trigger occurs in this register.

MS1/MS0 =0/0: Standby, no ISP/IAP operation.

MS1/MS0 =0/1: Data Flash/EEPROM read.

MS1/MS0 =1/0: Data Flash/EEPROM program.

MS1/MS0 =1/1: Data Flash/EEPROM page erase.

(4) IAP_TRIG: ISP/IAP Flash Command Trigger Register.

IAP_TRIG is the command port for triggering ISP/IAP activity and protected SFRs access. If IAP_TRIG is filled with sequential

（1）IAP_DATA：ISP/IAP Flash 数据寄存器。

它是 ISP/IAP 操作从闪存区中读、写数据的数据缓冲寄存器。

（2）IAP_ADDRH、IAP_ ADDRL：ISP/IAP 闪存地址寄存器。

IAP_ADDRH 用于存放操作地址的高 8 位，IAP_ADDRL 用于存放操作地址的低 8 位。

（3）IAP_CMD：ISP/IAP 闪存命令寄存器。

用于设置 ISP/IAP 的操作命令，但必须在命令触发寄存器实施触发后，方可生效。

MS1/MS0=0/0 时，为待机模式，无 ISP/ IAP 操作。

MS1/MS0=0/1 时，对数据闪存区进行字节读。

MS1/MS0=1/0 时，对数据闪存区进行字节编程。

MS1/MS0=1/1 时，对数据闪存区进行扇区擦除。

（4）IAP_TRIG：ISP/IAP 闪存命令触发寄存器。

IAP_TRIG 是 ISP/IAP 操作及受保护的 SFR 访问动作的命令触发寄存器，在（IAPEN）＝1 时，对 IAP_TRIG 先写入

0x5AH, 0xA5H and if IAPEN = 1, ISP/IAP activity or protected SFRs access will triggered.

(5) IAP_CONTR: ISP/IAP Control Register.

IAPEN: ISP/IAP operation enable.

1: Enable ISP/IAP opeiation function.

0: Global disable all ISP/IAP opeiation function.

SWBS、SWRST: Software boot selection control bit.

CMD_FAIL: Command fail indication for ISP/ IAP operation.

WT2, WT1, WT0: The setting of flash operation hold time for ISP/IAP operation. More details relate to Table 5.6.

0×5AH，再写入 0×A5H，ISP/IAP 命令或受保护的 SFR 访问动作生效。

（5）IAP_CONTR：ISP/IAP 控制寄存器。

IAPEN：ISP/IAP 功能允许位。IAPEN＝1，允许 ISP/IAP 操作改变数据；IAPEN＝0，禁止 ISP/IAP 操作改变数据。

SWBS、SWRST：软件复位控制位。

CMD_FAIL：ISP/IAP 闪存命令触发失败标志。

WT2、WT1、WT0：ISP/IAP 进行闪存操作时 CPU 等待时间的设置位。具体设置情况如表 5.6 所示。

Table 5.6 Setting of Operation Hold Time for ISP/IAP Operation

ISP/IAP 操作 CPU 等待时间的设置

WT2	WT1	WT0	CPU Waiting Time (system clock)			
			Program (55μs)	Read	Section Erase (21ms)	System clock f_{SYS}
1	1	1	55	2	21012	f_{SYS}＜1MHz
1	1	0	110	2	42024	1MHz＜f_{SYS}＜2MHz
1	0	1	165	2	63036	2MHz＜f_{SYS}＜3MHz
1	0	0	330	2	126072	3MHz＜f_{SYS}＜6MHz
0	1	1	660	2	252144	6MHz＜f_{SYS}＜12MHz
0	1	0	1100	2	420240	12MHz＜f_{SYS}＜20MHz
0	0	1	1320	2	504288	20MHz＜f_{SYS}＜24MHz
0	0	0	1760	2	672384	24MHz＜f_{SYS}＜30MHz

3. ISP/IAP Program and Application　3. ISP/IAP 编程与应用

1) Address Definition of ISP/IAP SFR　1）ISP/IAP 特殊功能寄存器地址声明

```
sfr   IAP_DATA    = 0xc2;    //IAP data register    IAP 数据寄存器
sfr   IAP_ADDRH   = 0xc3;    //IAP high address register    IAP 高地址寄存器
sfr   IAP_ADDRL   = 0xc4;    //IAP low address register    IAP 低地址寄存器
sfr   IAP_CMD     = 0xc5;    //IAP command register    IAP 命令寄存器
sfr   IAP_TRIG    = 0xc6;    //IAP trigger register    IAP 触发寄存器
sfr   IAP_CONTR   = 0xc7;    //IAP control register    IAP 控制寄存器
```

2) The Definitions of Instruction, Hold Time and Address of IAP Operation　　2）定义 ISP/IAP 命令、等待时间及 IAP 操作地址

```
#define   CMD_IDLE          0        // Defeat mode    无效模式
#define   CMD_READ          1        // Read instruction    读命令
#define   CMD_PROGRAM       2        // Program instruction    编程命令
#define   CMD_ERASE         3        // Erase instruction    擦除命令
#define   ENABLE_IAP        0x82     /* Enable IAP and set hold time
                                       允许 IAP，并设置等待时间*/
#define   IAP_ADDRESS       0x0000   /*Define the address of IAP operation
                                       定义 IAP 操作地址*/
```

3) Byte Read　　3）字节读

```
BYTE    dat;                        // BYTE 已定义为 8 位无符号数数据类型
IAP_CONTR = ENABLE_IAP;             /* Enable IAP and set hold time
                                       设置等待时间，并允许 IAP 操作*/
IAP_CMD = CMD_READ;                 /* Set the number of bytes read command 0x01
                                       送读字节数据命令 0x01*/
IAP_ADDRL = addr;                   /* Set the address for IAP read, the address is 16bit
                                    IAP address    设置 IAP 读操作地址，addr 为
                                    16 位 IAP 地址*/
IAP_ADDRH = addr>>8;
IAP_TRIG = 0x5A;                    /* Fill IAP_TRIG with sequential 0x5a,
                                    0xa5 to trigger IAP    对 IAP_TRIG 先送
                                    0x5a,再送 0xa5 触发 IAP 启动*/

IAP_TRIG=0xA5;
_nop_();                            /* Waiting    等待操作完成*/
 dat= IAP_DATA;                     /* Return data    取读出数据*/
```

4) Byte Program　　4）字节编程

NOTE：Ensure the program unit is empty, that is FFH, or a page erase must be carried out before byte program.

注意：字节编程前，必须保证编程单元内容为空，即为 FFH；否则须进行扇区擦除。

```
IAP_CONTR = ENABLE_IAP;             /* Enable IAP and set hold time    设置等待时
                                       间，并允许 IAP 操作*/
IAP_CMD = CMD_PROGRAM;              /* Set the program command 0x02    送编程命
                                       令 0x02*/
IAP_ADDRL =addr;                    /* Set the address for IAP read, the address is
                                    16bit IAP address    设置 IAP 编程操作地址，
                                    addr 为 16 位 IAP 地址*/
```

```
    IAP_ADDRH = addr>>8;
    IAP_DATA = dat;                 /*Set the program data; dat is the data going
                                      to be written  设置编程数据，dat 为要写入
                                      的数据*/
    IAP_TRIG = 0x5a;                /* Fill IAP_TRIG with sequential 0x5a,
                                      0xa5 to trigger IAP  对 IAP_TRIG 先送
                                      0x5a，再送 0xa5 触发 IAP 启动*/
    IAP_TRIG = 0xa5;
    _nop_();                        /* Waiting  稍等，待操作完成*/
```

5) Page Erase　　5）扇区擦除

```
    IAP_CONTR = ENABLE_IAP;         /* Enable IAP and set hold time  设置等待时
                                      间 3，并允许 IAP 操作*/
    IAP_CMD = CMD_ERASE;            /* Set the page erase 0x03  送扇区删除命令
                                      0x03*/
    IAP_ADDRL = addr;               /* Set the address for IAP page erase command
                                      设置 IAP 扇区删除操作地址*/
    IAP_ADDRH = addr>>8;
    IAP_TRIG = 0x5a;                /* Fill IAP_TRIG with sequential 0x5a,
                                      0xa5 to trigger IAP  对 IAP_TRIG 先送
                                      0x5a，再送 0xa5 触发 IAP 启动*/
    IAP_TRIG = 0xa5;
    _nop_();                        /* Waiting  稍等，待操作完成*/
```

NOTE： Any address in the page is allowed in page erase operation.
注： 扇区擦除时，输入该扇区的任意地址皆可。

E.g.

EEPROM test. Connecting six LED lights with port P1, active low. After program running, light up the LED controlled by P1.0, and then execute the pages erase instruction. Light up the LED controlled by P1.1 if the erase operation succeeded. Then, data write operation starts from 0000H. Light up the LED controlled by P1.2 after data write, and then data verification is carried out. The verification succeeds when the LED controlled by P1.3 is lighted up. Or, the verification fails when the LED controlled by P1.5 is lighted up.

Here gives the solution:

This is a simple test, in the purpose of how to carry

例　EEPROM 测试。用 P1 口连接 6 只 LED 灯，低电平有效。当程序开始运行时，点亮 P1.0 控制的 LED 灯，接着进行扇区擦除并检验，若擦除成功再点亮 P1.1 控制的 LED 灯，接着从 EEPROM 中 0000H 开始写入数据，写完后再点亮 P1.2 控制的 LED 灯，接着进行数据校验，若校验成功再点亮 P1.3 控制的 LED 灯，测试成功；否则，点亮 P1.5 控制的 LED 灯，表示测试失败。

out such ISP/IAP operations as page erase, byte program and byte read. Given that the frequency of oscillator is 18.432MHz.

解：本测试是一个简单测试，目的是学习如何对 EEPROM 进行扇区删除、字节编程、字节读的 ISP/IAP 操作。设晶振频率为 18.432MHz。

```
#include <reg52.h>
#include <intrins.h>
/*----------------------Definition of SFR----------------------*/
sfr   IAP_DATA    = 0xc2;          //IAP data register    IAP 数据寄存器
sfr   IAP_ADDRH = 0xc3;          //IAP high address register    IAP 高地址寄存器
Sfr   IAP_ADDRL = 0xc4;          //IAP low address register    IAP 低地址寄存器
sfr   IAP_CMD     = 0xc5;          //IAP command register    IAP 命令寄存器
sfr   IAP_TRIG     = 0xc6;          //IAP trigger register    IAP 触发寄存器
sfr   IAP_CONTR  = 0xc7;          //IAP control register    IAP 控制寄存器
/*---------------------Macro definition----------------------*/
typedf   unsigned   char   BYTE        // BYTE 等效于 unsigned    char
typedf   unsigned   int   WORD         // WORD 等效于 unsigned    int
/*---------------------Define IAP operation mode word and test address----------------------*/
#define   CMD_IDLE            0        // Defeat mode    无效模式
#define   CMD_READ            1        // Read instruction    读命令
#define   CMD_PROGRAM    2        // Program instruction    编程命令
#define   CMD_ERASE           3        // Erase instruction    擦除命令
#define   ENABLE_IAP     0x82       /* Enable IAP and set hold time    允许 IAP, 并设置
                                       等待时间*/
#define   IAP_ADDRESS    0x0000     /* Define the address of IAP operation    定义 IAP
                                       操作地址*/
/*---------------------- Delay Subroutine ----------------------*/
void Delay(BYTE n)
{
      WORD    x;
      while( n--)
      {
            x=0;
            while(++x);
      }
}
/*----------------------Disable IAP function----------------------*/
void IapIdle()
```

```
{
    IAP_CONTR = 0;                  // Disable IAP function    关闭 IAP 功能
    IAP_CMD = 0;                    // Clear command register    清除命令寄存器
    IAP_TRIG = 0;                   // Clear trigger register    清除触发寄存器
    IAP_ADDRH = 0x80;               /* Set the address to non-IAP area    将地址设置到
                                    非 IAP 区域*/
    IAP_ADDRL = 0;
}
 /*----------------------Read EEPROM byte subroutine----------------------*/
BYTE    IapReadByte(WORD    addr)       /* Formal parameter consist of high and low
                                        address    形参为高位地址和低位地址*/
{
    BYTE    dat;
    IAP_CONTR = ENABLE_IAP;         /* Enable IAP and set hold time
                                        设置等待时间，并允许 IAP 操作*/
    IAP_CMD = CMD_READ;             /* Set the number of bytes read command 0x01
                                        送读字节数据命令 0x01*/
    IAP_ADDRL = addr;               /* Set the address for IAP read
                                        设置 IAP 读操作地址*/
    IAP_ADDRH = addr>>8;
    IAP_TRIG = 0x5a;    /* Fill IAP_TRIG with sequential 0x5a, 0xa5 to trigger IAP
                            对 IAP_TRIG 先送 0x5a，再送 0xa5 触发 IAP 启动*/
    IAP_TRIG=0xa5;
    _nop_();                        // Waiting    稍等，待操作完成
    da t= IAP_DATA;                 // Return data    返回读出数据
    IapIdle( )；                    // Disable IAP    关闭 IAP
    Return dat;
}
 /*----------------------Write EEPROM byte subroutine----------------------*/
void IapProgramByte(WORD addr,    BYTE dat)
/* Erase the page where byte data locates    对字节地址所在扇区擦除*/
{
    IAP_CONTR = ENABLE_IAP;         /* Enable IAP and set hold time
                                        设置等待时间，并允许 IAP 操作*/
    IAP_CMD = CMD_PROGRAM;          /* Send program command 0x02
                                        送编程命令 0x02*/
    IAP_ADDRL =addr;                /* Set the address for IAP program
                                        设置 IAP 编程操作地址*/
```

```
    IAP_ADDRH= addr>>8;
    IAP_DATA = dat;                          //Set the program data   设置编程数据
    IAP_TRIG = 0x5a;       /* Fill IAP_TRIG with sequential 0x5a, 0xa5 to trigger IAP
                           对 IAP_TRIG 先送 0x5a，再送 0xa5 触发 IAP 启动*/
    IAP_TRIG = 0xa5;
    _nop_();                                 // Waiting   等待操作完成
    IapIdle( )；
}
/*----------------------Section Erase--------------------*/
void IapEraseSector(WORD addr)
{
    IAP_CONTR = ENABLE_IAP;          /* Enable IAP and set hold time
                                        设置等待时间，并允许 IAP 操作*/
    IAP_CMD = CMD_ERASE;             /* Send page erase command 0x03
                                        送扇区删除命令 0x03*/
    IAP_ADDRL = addr;                /* Set the address of IAP page erase operation
                                        设置 IAP 扇区擦除操作地址*/
    IAP_ADDRH = addr>>8;
    IAP_TRIG = 0x5a;      /* Fill IAP_TRIG with sequential 0x5a, 0xa5 to trigger
                             IAP   对 IAP_TRIG 先送 0x5a，再送 0xa5 触发 IAP 启动*/
    IAP_TRIG = 0xa5;
    _nop_();                                 // Waiting   等待操作完成
     IapIdle( )；
}

/*----------------------Main----------------------*/
void main()
{
    WORD i;
    P1=0xfe;                      /* Light LED controlled by P1.0 when program
                                     is running   程序运行时，点亮 P1.0 控制的 LED 灯*/
    Delay(10);
    IapEraseSector(IAP_ADDRESS)          // Page erase   扇区擦除
    for(i=0, i<512, i++)
    {
        if(IapReadByte(IAP_ADDRESS+i)!=0xff)
        Goto Error;
    }
```

```
        P1=0xfc;                /* Page erase success and then light up the LED controlled by
                                   P1.1　扇区擦除成功，再点亮 P1.1 控制的 LED 灯*/
        Delay(10);
        for(i=0, i<512, i++)
        {
            IapProgramByte(IAP_ADDRESS+i, (BYTE)i);
        }
        P1=0xf8;        /* Program finish and then light up the LED controlled by P1.2
                           编程完成，再点亮 P1.2 控制的 LED 灯*/
        Delay(10);
        for(i=0, i<512, i++)
        {
            if(IapReadByte(IAP_ADDRESS+i)!=(BYTE)i)
            Goto　Error;        /* Verification fails; turn to error handling　校验失败，转错
                                   误处理*/
        }
        P1=0xf0;                /* Program verification succeeds and then light up the LED
                                   controlled by P1.3　编程校验成功，再点亮 P1.3 控制
                                   的 LED 灯*/
        while(1);
        Error:                  /* Light up the LED controlled by P1.5 when page erase or
                                   program verification fails　若扇区擦除不成功或编程校验不成
                                   功，点亮 P1.5 控制的 LED 灯*/
        P1&=0xdf;
        while(1);
}
```

4. Notice about the Usage of EEPROM

(1)The requirement of voltage in ISP/IAP operation: Once the V_{cc} of MCU is lower than the low-voltage checking threshold, disable the ISP/IAP operation, which disable the normal operations of EEPROM as well. MCU will not response to any ISP/IAP instruction. In fact, the operation of accessing ISP/IAP register has been carried out, but the erase, program and read instructions to EEPROM are invalid because the ISP/IAP operation is disabled in the case of lower work voltage than reliable threshold voltage.

4. EEPROM 使用注意事项

（1）ISP/IAP 操作的工作电压要求：当单片机 V_{cc} 小于低压检测门槛电压时，禁止 ISP/IAP 操作，即禁止对 EEPROM 的正常操作，此时单片机对相应的 ISP/IAP 指令不响应。实际情况是对 ISP/IAP 寄存器的操作是执行了，但由于此时工作电压低于可靠的门槛电压，单片机内部此时禁止执行 ISP/IAP 操作，即对 EEPROM 的擦除/编程/读命令均无效。

Due to the slow power up of power supply, the program may start running already while the power supply doesn't reach the minimum reliable threshold voltage, which lead to a failure in the corresponding EEPROM instruction, so that it's recommended to choose a high threshold voltage of reset. If a wider range working voltage is required and low threshold voltage of reset is chosen, it's recommended to check the low voltage flag LVDF before operating the EEPROM. That means the power supply was lower than the reliable threshold voltage when the flag is "1", which will be cleared by software. If this flag turns to "0" after several nops, the working voltage is higher than the threshold, and ISP/IAP operations can be executed. Clear it again when the flag is still "1". Don't execute any ISP/IAP operation until the working voltage is higher than the threshold. The LVDF resides in power manage register PCON, whose format is shown as Table 5.7.

如果电源上电缓慢，可能会由于程序已经开始运行，而此时电源电压还达不到 EEPROM 的最低可靠工作电压，导致相应的 EEPROM 指令无效，所以建议用户选择高的复位门槛电压。如用户需要宽的工作电压范围，选择了低的复位门槛电压，建议对 EEPROM 进行操作时，要判断低电压 LVDF 标志位。如果该位为“ 1 ”，则说明电源电压曾经低于有效的门槛电压，应由软件将其清零，加几个空操作延时后再读该位的状态，如果为“ 0 ”，说明工作电压高于有效的门槛电压，则可进行 ISP/IAP、EEPROM 操作。如果为“ 1 ”，则将其再清零，一直等到工作电压高于有效的门槛电压，才能进行 ISP/IAP、EEPROM 操作。LVDF 标志位在电源控制寄存器 PCON 中，PCON 的格式如表 5.7 所示。

Table 5.7　PCON

PCON	Address 地址	D7	D6	D5	D4	D3	D2	D1	D0	Reset Value 复位值
	87H	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL	00110000B

(2) The data modified at the same operation should be put into the same page, otherwise put into different pages, so that the data needn't to be read out and protected in operation. The less bytes used in each page, the more convenient it is. It's a real EEPROM when only one byte is stored in one page. The Data Flash of STC MCU is much faster than external EEPROM, where reading a byte costs 2 clock, programing one byte costs 55 μs, and erasing one costs 21ms.

（2）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，操作时就不用读出来进行保护了。每个扇区使用时，使用的字节数越少越方便。如果一个扇区只放一个字节，那就是真正的 EEPROM 了，STC 单片机的 Data Flash 比外部 EEPROM 要快很多，读一个字节是 2 个时钟，编程一个字节是 55μs，擦除一个扇区是 21ms。

Summary 本章小结

There are three individual memory partitions in STC15W201S series MCU: Program Memory (Program Flash), On-chip Scratch-pad RAM and EEPROM (Data Flash).

Program memory stores not only program used to conduct the MCU running, but also some constants and tables, such as digital word and so on. The designation of storing type in program memory is used to define data in C. The table-lookup instruction is used to fetch data in assembly language while array reference in C.

The scratch-pad RAM consists of lower 128 bytes, upper 128 bytes and SFR, where the addresses of both upper 128 bytes and SFR are overlapping, which could be distinguished by addressing. The upper 128 bytes of data memory may only be accessed using indirect address while the SFR can only be accessed through direct addressing. The lower 128 bytes of data memory may be accessed through both direct and indirect addressing. The lowest 32 bytes of data (00H~1FH) of data memory are grouped into register banks. The next 16 bytes (20H~2FH) form a block of bit-addressable memory space.

Internal EEPROM, which could be repeatedly erased more than 100 thousand times, can be used as Data Flash by In-application Program(IAP) technology. Byte read, byte write and page erase can be operated on Data Flash.

Exercise 练习题

1. Completions

1) The main characteristic of STC15W201S MCU memory structure lies in_____ and data memory are addressing separately.

2) The program memory stores_____, constant data and _____data.

3) The address space pointed by PC in STC15W201S CPU is_____.

4) The executing of user program of STC15W201S begins with _____unit.

5) The memory addresses 0003H~00BBH for program storing stand for the _____ address in STC15W201S.

6) There 3 separate memory spaces in STC15W201S inner memory in physics, they are_____, _____ and Stretch RAM; and it can also divided into 4 separate memory spaces according to application: _____, _____, Scratch-pad RAM and _____.

7) The inner basic RAM of STC15W201S is divided into 3 sections: lower 128 bytes, _____ and_____. According to the application of RAM, the lower 128 byte section can be divided into_____, _____ and general purpose RAM section.

8) The address space of working registers is_____, and the address space for bit addressing is_____.

9) The address space of higher 128 bytes section is the same as SFR. It will access 128 byte section when adopting_____ addressing method; while it will access SFR when adopting_____ addressing method.

10)In SFR, it can be bit addressable when the address can be divisible by_____. The corresponding addressable bit owns a bit address, which is byte address plus_____. But it will be indicated by_____ in practical program, such as the CY and AC in PSW.

11)The EEPROM in PSW series MCU, which is not real EEPROM, is simulated by_____. For STC15 series MCU, both user program section and EEPROM section are_____ addressing, which are Program Flash and Data Flash; For IAP15 series MCU, user program section and EEPROM section are_____ addressing, and the idle user program section can be used as EEPROM.

12)The Scratch-pad RAM in STC15W201S MCU is divided into inner extension RAM and_____ extension RAM, but they cannot be actived simultaneously. When the EXTRAM is_____, off-chip extension RAM is selected; when MCU is resetting, and EXTRAM is_____, _____ is selected.

13)The size of program memory in STC15W201S MCU is_____, and its address range is_____.

14)The size of Scratch-pad RAM in STC15W201S MCU is_____, and its address range is_____.

2. Choice Questions

1) Group____ working registers will be adopted by CPU when RS1RS0=01.

A. 0　　B. 1　　C. 2　　D. 3

2) When CPU needs to adopt 2 groups of working registers, RS1RS0 should be set to_____.

A. 00　　B. 01　　C. 10　　D. 11

3) When RS1RS0=11, the address of R0 in RAM is_____.

A. 00H　　B. 08H　　C. 10H　　D. 18H

4) To define an x variable, 8-bit unsigned, be allocated to program memory section and assigned to 100. The correct statement is _____.

A. unsigned char code x=100;　　B. unsigned char data x= 100;

C. unsigned char xdata x =100;　　D. unsigned char code x; x= 100;

5) To define a 16-bit unsigned y variable, be allocated to bit-addressing section, the correct statement is_____.

A. unsigned int y；　　B. unsigned int data y；

C. unsigned int xdata y；　　D. unsigned int bdata y；

6) When IAP_CMD=01H, the functional operation of ISP/IAP is _____.

A. No ISP/IAP operation; B. Read the Data Flash;

C. Program the Data Flash; D. Erase the Data Flash;

3. True or False

1) STC15W201S MCU retains the extended off-chip program memory and off-chip data memory. ( )

2) The SFR whose byte-address can be divisible by 4 is bit-addressable. ( )

3) EEPROM in STC15W201S MCU is addressing together with user program section, and the idle user program section can be used as EEPROM by IAP. ( )

4) On-chip extension RAM and off-chip extension RAM can be active simultaneously. ( )

5) EEPROM in STC15W201S MCU is real EEPROM, which can be erased in byte and read/write in byte. ( )

6) EEPROM in STC15W201S MCU is erased in section. ( )

7) The trigger code of EEPROM operation in STC15W201S MCU is first A5H and then 5AH. ( )

4. Question and Answer

1) The address of higher 128 byte is conflict with the SFRs, how to distinguish between them?

2) How to describe the bit address of SFR in application when SFRs are bit-addressable?

3) While on-chip extension RAM and off-chip extension RAM can be active simultaneously, how to choose in application?

4) Is there any special implication in the 0000H address storing in program?

5) Is there any special implication in the 000023 address?

6) Please make a description of the working flow of the EEPROM read operation in STC15W201S MCU.

7) Please make a description of the working flow of the EEPROM erase operation in STC15W201S MCU.

8) What is the difference between global variable and local variable? How to define global variable and local variable?

9) What should be noted in program calling when main function and subfunction are in the same program file? What's that when they are in different files?

10)Make interpretations of x/y and x%y. Make a brief description of how to split units digit, tens digit and hundreds digit.

Unit 6　Timer/Counter in STC15W201S MCU

第 6 单元　STC15W201S 单片机的定时/计数器

Timer and counter are always used for timing (or delay) control and external events counting in MCU application system. There are several types of timing as below.

1)　Software timing

When CPU is running a program circularly, it's able to achieve software timing by instructions selection and the number of cycle arrangement. The software timing will occupy CPU fully, add overhead and decrease the CPU's efficiency. Thus, software timing should not last long and it's suitable for the program in which the CPU is less busy.

2)　Hardware timing

The hardware timing, whose timing function is implemented by hardware circuit (such as 555 time-base circuit) and doesn't occupy CPU time, needs to change the circuit parameter to regulate timing, which is not so convenient in use and increase the cost of hardware.

3)　Programmable timer timing

The value and range of programmable timer timing are easily set and modified by software. There are two 16 bit timer/counter (T0 and T2) in STC15W201S MCU, which can be used for timing control or frequency divider and event recorder

在单片机应用系统中，常常需要时钟和计数器，以实现定时（或延时）控制以及对外界事件进行计数。可供选择的定时方法有以下几种。

1）软件定时

让 CPU 循环执行一段程序时，可通过选择指令和安排循环次数以实现软件定时。软件定时要完全占用 CPU，增加 CPU 开销，降低了 CPU 的工作效率，因此软件定时的时间不宜太长。仅适用于 CPU 较空闲的程序。

2）硬件定时

硬件定时的特点是定时功能全部由硬件电路（例如 555 时基电路）完成，不占用 CPU 时间，但需要改变电路的参数调节定时时间，在使用上不够方便，同时增加了硬件成本。

3）可编程定时器定时

可编程定时器的定时值及定时范围很容易通过软件来确定和修改。STC15W201S 单片机内部有 2 个 16 位的定时/计数器（T0 和 T2），通过对系统时钟或外部输入信号进行计

conveniently through the counting and control on the system clock or the input signal for outside.

The key circuit of timer or counter is a counter, shown in Figure 6.1. There are two sources for the counter pulse: one is the external pulse source T0 (P1.2); the other one is the system clock. One of the two sources is accepted for counting, plus 1 for one pulse. The counter will return to 0 after an arrival of new pulse when the counter is full already. The overflow flag TF0 will be set to "1" at the same time and an interrupt request is sent to CPU.

数与控制，可以方便地用于定时控制与事件记录，或作为分频器使用。

定时/计数器的核心电路是一个计数器，如图 6.1 所示。加 1 计数器的脉冲有两个来源：一个是外部脉冲源 T0（P1.2），另一个是系统的时钟信号。计数器对两个脉冲源之一进行输入计数，每输入一个脉冲，计数值加 1。当计数到计数器为全 1 时，再输入一个脉冲就使计数值回零，同时使计数器计满溢出标志位 TF0 置 1，并向 CPU 发出中断请求。

NOTE: The overflow flag TF2 of T2 is implicit and invisible. An interrupt request will be sent to CPU directly.

注：T2 的计满溢出标志位（TF2）是隐含的，不可见。当计满溢出时直接向 CPU 发出中断请求。

Timing: When the source of pulse is system clock (equal interval pulse sequence), the timing time, because the counting pulse is time reference, is the number of pulse times pulse period (system clock or 12 times the clock).

Counting: When the source of pulse is the external input pulse (from pins T0 or T2) which is unequal interval pulse sequence, the counter is for external events counting. The counter will be plus 1 when there is a negative transition in corresponding external input port T0 or T2. The ratio of external input signal won't be limited but that every signal should be sampled before the level changes.

定时功能：当脉冲源为系统时钟（等间隔脉冲序列）时，由于计数脉冲为一时间基准，脉冲数乘以计数脉冲周期（系统周期或 12 倍系统周期）就是定时时间。

计数功能：当脉冲源为间隔不等的外部输入脉冲（由 T0 或 T2 引脚输入）时，就是外部事件的计数器。计数器在其对应的外输入端 T0 或 T2 有一个负跳变时，计数器的状态值加 1。外部输入信号的速率是不受限制的，但必须保证给出的电平在变化前至少被采样一次。

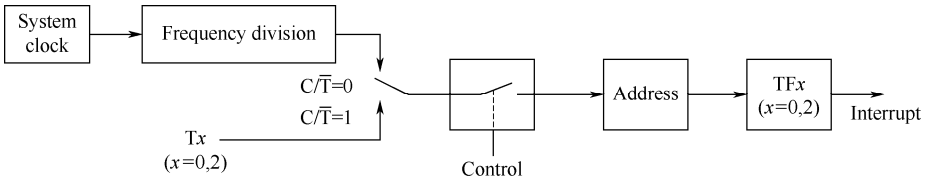


Figure 6.1　Block Diagram of STC15W201S MCU Counter Circuit

STC15W201S 单片机计数器电路框图

6.1 Timer/Counter 0 定时/计数器 T0

6.1.1 Control of Timer/Counter 0 定时/计数器 T0 的控制

The working mode and control are managed by SFR TMOD, TCON and AUXR.

TMOD: Set the working mode and function of Timer/Counter 0.

TCON: Start or stop Timer/Counter 0, including overflow flag of Timer/Counter 0.

AUXR: Set the division factor of timing/counting pulse.

STC15W201S 单片机定时/计数器 T0 的工作方式和控制由 TMOD、TCON 和 AUXR 三个特殊功能寄存器进行管理。

TMOD：设置定时/计数器 T0 的工作方式与功能。

TCON：控制定时/计数器 T0 的启动与停止，并包含定时/计数器 T0 的溢出标志位。

AUXR：设置定时计数脉冲的分频系数。

1. Working Mode Register TMOD

TMOD is the working mode register, whose format is shown in Table 6.1.

1. 工作方式寄存器 TMOD

TMOD 为工作方式寄存器，其格式如表 6.1 所示。

Table 6.1 TMOD

	Address	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value
TMOD	89H	-	-	-	-	GATE	C/$\overline{T}$	M1	M0	xxxx 0000
		← Timer/Counter 0 →								

(1) M0 and M1: Mode selection bit.Shown as Table 6.2.

（1）M1 和 M0：方式选择位。定义如表 6.2 所示。

Table 6.2 M1 & M0

M1 M0	Mode 工作方式	Description 功能说明
0 0	Mode 0	16-bit auto-reload Timer/Counter 自动重装初始值的 16 位定时/计数器
0 1	Mode 1	16-bit Timer/Counter 16 位定时/计数器
1 0	Mode 2	8-bit auto-reload Timer/Counter 自动重装初始值的 8 位定时/计数器
1 1	Mode 3	Divided into 2 8-bit Timer/Counter 分成 2 个 8 位定时/计数器

(2) C / $\overline{T}$: Function selection bit.

0: Timing mode;

1: Counting mode.

（2）C / $\overline{T}$ ：功能选择位。

C / $\overline{T}$ = 0 时，设置为定时工作模式；

C / $\overline{T}$ = 1 时，设置为计数工作模式。

(3)GATE: Gating bit.

0: Start timing/counting once the control bit TR0 is set to 1.

1: Start timing/counting when the software control bit TR0 and INT0 (P3.2) is high level simultaneously, that is, external interrupt INT0 takes part in the start and stop of timer/counter.

TMOD is not bit-addressable and set working mode only by byte instruction. The lower 4 bit of TMOD is defined as T0 while the upper 4 bit is invalid. All bits in TMOD are set to 0 after reset.

（3）GATE：门控位。

当 GATE=0 时，控制位 TR0 置 1 即可启动定时/计数器。当 GATE=1 时，软件控制位 TR0 须置 1，同时还要求 INT0（P3.2）为高电平方可启动定时/计数器，即允许外中断 INT0 输入引脚信号参与控制定时/计数器的启动与停止。

TMOD 不能位寻址，只能用字节指令设置定时器工作方式，高 4 位无效，低 4 位定义 T0。复位时，TMOD 所有位均置 0。

NOTE：In traditional 8051 MCU, the upper 4 bit of TMOD is used for setting working mode of Timer/Counter 1.

注：在传统 8051 单片机中，TMOD 的高 4 位用于选择定时/计数器 T1 的工作方式。

For instance, if the Timer 0 needs to be set to Mode 0, timing working mode has nothing to do with the external interrupt input, then, (M1)=0, (M0)=0, $(C/\overline{T})=0$ and (GATE)=0, thus the lower 4 bit should be 0000 and upper 4 bit 0000. Above all, the instruction should be “TMOD= 0x00;”.

比如，需要设置 T0 工作于方式 0 定时模式，定时工作方式与外部中断输入引脚信号无关，那么 M1=0、M0=0，$C/\overline{T}=0$、GATE=0，因此，低 4 位应为 0000，高 4 位设为 0000。因此，指令形式为：“ TMOD=0x00; ”。

2. Control Register TCON

TCON is used for controlling the start and stop of timer/counter, and recording the overflow flag and external interrupts. The format of TCON is shown in Table 6.3.

2. 控制寄存器 TCON

TCON 的作用是控制定时/计数器的启动与停止、记录溢出标志以及外部中断的控制。TCON 的格式如表 6.3 所示。

Table 6.3　TCON

TCON	Address 地址	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value 复位值
	88H	-	-	TF0	TR0	IE1	IT1	IE0	IT0	xx00 0000

(1) TF0: Overflow flag of Timer/Counter 0. TF0 is set automatically by hardware when there is an overflow in Timer/Counter 0. When the interrupt is enabled, an interrupt request is sent to CPU. TF0 will be cleared by hardware after the interrupt was responded. Inquiry to TF0 can be used for determining the time of overflow. Clear the TF0 by software after inquiry.

（1）TF0：定时/计数器 0 溢出标志位。当 T0 计满产生溢出时，由硬件自动置位 TF0。在中断允许时，向 CPU 发出 T0 的中断请求，中断响应后，由硬件自动清零。也可通过查询 TF0 标志来判断计满溢出时刻，查询结束后，用软件清除 TF0 标志。

(2) TR0: Control bit of Timer/Counter 0. Setting 1 or 0 by software is used for starting or stopping Timer/Counter 0. When (GATE) =0, start Timer/Counter 0 by setting 1 to TR0; when (GATE) =1, start Timer/Counter 0 by setting 1 to TR0 and the input of INT0 (P3.2) is high level at the same time.

The lower 4 bit is used for controlling the external interrupt, nothing to do with timer/counter, which will be introduced in next unit. All bits in TCON will be cleared after reset.

The byte address of TCON is 88H and is byte addressable. TF0 clearance, and start or stop of timer/ counter can be executed by bit manipulation instruction.

3. Auxiliary Register(AUXR)

The T0x12 of AUXR is used for setting the frequency division factor of timing/counting pulse. The format of AUXR is shown in Table 6.4.

（2）TR0：定时/计数器 0 运行控制位。由软件置 1 或清零来启动或关闭 T0。当 GATE=0 时，TR0 置 1 即可启动 T0；当 GATE=1 时，TR0 置 1 且 INT0(P3.2)输入引脚信号为高电平时，方可启动 T0。

TCON 中的低 4 位用于控制外部中断，与定时器/计数器无关，将在下单元中学习。当系统复位时，TCON 的所有位均清零。

TCON 的字节地址为 88H，可以位寻址，清除溢出标志位或启动、停止定时/计数器都可以用位操作指令。

3. 辅助寄存器 AUXR

辅助寄存器 AUXR 的 T0x12 用于设定定时/计数脉冲的分频系数。格式如表 6.4 所示。

Table 6.4 AUXR

AUXR	Address 地址	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value 复位值
	8EH	T0x12	-	UART_M0x6	T2R	T2_C/$\overline{T}$	T2x12	EXTRAM	S1ST2	0x000000

T0x12: Used for setting the division factor of Timer/Counter 0 timing/counting pulse.

0: The pulse period is 12 times the system clock period, that is 1/12 frequency dividing, which is just the same as traditional 8051 MCU.

1: The pulse period is equal to the system clock period. That is no division here.

T0x12：用于设置 T0 定时/计数脉冲的分频系数。

当 T0x12＝0，定时计数脉冲完全与传统 8051 单片机的计数脉冲一样，计数脉冲周期为系统时钟周期的 12 倍，即 12 分频。

当 T0x12＝1，计数脉冲周期等于系统时钟周期，即无分频。

6.1.2　Working Mode of Timer/Counter 0
定时/计数器 T0 的工作方式

Timer/Counter 0 can be configured into four modes, Mode 0, Mode 1, Mode 2 and Mode3, by setting M1 (TMOD.1) and M0 (TMOD.0) in TMOD. Timer/Counter 0 can work in any one of the four modes. Where, Mode 1 and Mode 2 both can be attained by Mode 0, while Mode 3 is rarely used in practical application. Thus, only Mode 0 of Timer/ Counter 0 is introduced here.

Timer/Counter 0 works in Mode 0 when the M1 and M0 in TMOD are both “0”.

In Mode 0, the Timer/Counter 0 is configured in a 16-bit auto-reload timer/counter, which is show as Figure 6.2. There are two hidden registers RL_TH0 and RL_TL0 in Timer/Counter 0, which are used for storing the 16-bit reload value. When the counter formed by TH0 and TL0 is overflow, the value of RL_TH0 and RL_TL0 will be written into TH0 and TL0 automatically. RL_TH0 shares the same address with TH0, so does RL_TL0 and TL0. When (TR0=0), the content written into TH0 and TL0 will be written into RL_TL0 and RL_TH0 too. When (TR0)=1, the content written into TH0 and TL0 actually will not be written into TH0 and TL0, but into RL_TL0 and RL_TH0. Thus, T0 will not be affected in such way.

If $(C/\overline{T})=0$, Timer/Counter 0 would be set for Timer operation, and the multiplexer will connect to the output of division clock. Timer/Counter 0 will count the timing counting pulse. Which kind of clock to be used is determined by T0x12. When (T0x12) = 0, adopts the clock after 12 times division (compatible with traditional 8051 MCU); When (T0x12) = 1,

通过对 TMOD 的 M1、M0 的设置，定时/计数器 T0 有 4 种工作方式，分别为方式 0、方式 1、方式 2 和方式 3。定时/计数器 T0 可以工作在这 4 种工作方式中的任何一种，其中，方式 1、方式 2 的功能可完全由方式 0 实现，而方式 3 在实际应用中极少使用，为此，在本书中只介绍定时/计数器 T0 的工作方式 0。

当 TMOD 的 M1、M0 的值都为 0 时，定时/计数器 T0 工作在方式 0。

方式 0 下，T0 是一个 16 位可自动重装初始值的定时/计数器，其结构如图 6.2 所示，T0 有两个隐含的寄存器 RL_TH0、RL_TL0，用于保存 16 位重装初始值，当 TH0、TL0 构成的 16 位计数器计满溢出时，RL_TH0、RL_TL0 的值自动装入 TH0、TL0 中。RL_TH0 与 TH0 共享同一个地址，RL_TL0 与 TL0 共享同一个地址。当 TR0＝0 时，对 TH0、TL0 寄存器写入数据时，也会同时写入 RL_TH0、RL_TL0 寄存器中；当 TR0＝1 时，对 TH0、TL0 写入数据时，只写入 RL_TH0、RL_TL0 寄存器中，而不会写入 TH0、TL0 寄存器中，这样不会影响 T0 的正常计数。

当 $C/\overline{T}=0$ 时，多路开关连接系统时钟的分频输出，定时/计数器 T0 对定时计数脉冲计数，即工作于定时工作方式。由 T0x12 决定如何对系统时钟进行分频，当 T0x12＝0 时，使用 12 分频（与传统 8051 单片机兼容）；当 T0x12＝1 时，直接使用系统时钟（即不分频）。

adopts the system clock directly without division.

If $(C/\overline{T})=1$, Timer/Counter 0 would be set for Counter operation. Timer/Counter 0 will count the pulse input from pin T0.

GATE function: In general, GATE should be set to "0" so that Timer/Counter 0 works under the control of TR0 (1: start; 0: stop). Set GATE to "1" only when the start of counting is controlled by external input INT0. According to Figure 6.2, if (GATE)=1, Timer/Counter 0 starts to count only when TR0 is 1 and the input from INT0 pin is high, which can facilitate pulse width measurement.

当 $C/\overline{T}=1$ 时，定时/计数器 T0 对 T0 引脚输入脉冲计数，即工作于计数工作方式。

门控位 GATE 的作用：一般情况下，应使 GATE 为 0，这样，T0 的运行控制仅由 TR0 位的状态确定（TR0 为 1 时启动，TR0 为 0 时停止）。只有在启动计数要由外部输入 INT0 控制时，才使 GATE 为 1。由图 6.2 可知，当 GATE＝1 时，TR0 为 1 且 INT0 引脚输入高电平时，定时/计数器 T0 才能启动计数。利用 GATE 的这一功能，可以很方便地测量脉冲宽度。

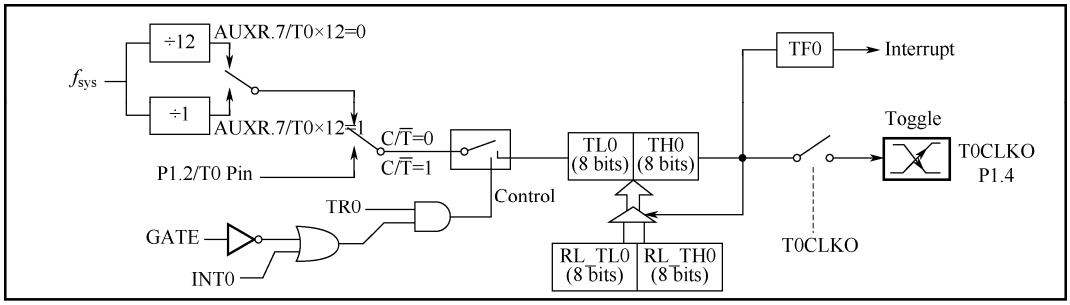


Figure 6.2 Timer/Counter Mode 0

定时/计数器的工作方式 0

$$\text{Timing time} = (M - \text{Initial value of timer}) \times \text{Clock cycle} \times 12^{(1-T0\times12)}$$

$$M = 2^{16} = 65536$$

E.g. Implement timing using T0 Mode 0, output a square wave with a clock period of 10ms in pin P1.0.

Here gives the solution:

Using the T0 Mode 0, thus, (TMOD) = 00H.

Because the clock period of square wave is 10ms, the timing time of T0 should be 5ms. Reverse the P1.0 every 5ms, which can attain such kind of square wave output in pin P1.0. The system adopts a 12MHz oscillator, the division factor is 12 and the period of pulse clock is 1μs. Thus, the initial value of T1 is:

例 用 T0 方式 0 实现定时，在 P1.0 引脚输出周期为 10ms 的方波。

解：根据题意，采用 T0 方式 0 进行定时，因此，TMOD=00H。

因为方波周期是 10ms，因此 T0 的定时时间应为 5ms，每 5ms 时间到就对 P1.0 取反，就可实现在 P1.0 引脚输出周期为 10ms 的方波。系统采用 12MHz 晶振，分频系数为 12，即定时脉钟周期为 1μs，则 T1 的初值为：

$X = M$ – Counting value = 65536 – 5000 = EC78H

That is: (TH0) = ECH; (TL0) = 78H.　即：TH0= ECH，TL0=78H。

The reference C51 code is shown as below:　C51 参考程序如下：

```
#include<reg51.h>
#define uchar unsigned char
#define uint unsigned int
sbit P10=P1^0;
void T0_init()
{
    TMOD=0x00;
    TH0=0xec;          // (65536-5000)/256
    TL0=0x78;          // (65536-5000)%256
    TR0=1;             // Start T0    启动定时器 T0
}
void   main()
{
    T0_init();
    while(1)
    {
        if(TF0==1)
        {
            TF0=0;
            P10=!P10;
        }
    }

}
```

6.2　Timer/Counter 2　定时器 T2

6.2.1　Structure of T2　定时器 T2 的电路结构

The structure of Timer/Counter 2 in STC15W201S MCU is shown as Figure 6.3. Such structure is almost the same as T0 in general, but the working mode of T2 is fixed at 16-bit initial value auto-reloadable mode.

STC15W201S 单片机定时/计数器 T2 的电路结构如图 6.3 所示。T2 的电路结构与 T0 基本一致，但 T2 的工作模式固定为 16 位自动重装初始值模式。

There are two hidden registers RL_TH2 and RL_TL2 in Timer/Counter 2, which are

T2 有两个隐含的寄存器 RL_TH2、RL_TL2，用于保存 16 位定时/计数器的

used for storing the 16-bit reload value. When the counter formed by TH2 and TL2 is overflow, the value of RL_TH2 and RL_TL2 will be written into TH2 and TL2 automatically. RL_TH2 shares the same address with TH2, so does RL_TL2 and TL2. When (TR2)=0, the content written into TH2 and TL2 will be written into RL_TL2 and RL_TH2 too. When (TR2)=1, the content written into TH2 and TL2 actually will not be written into TH2 and TL2, but into RL_TL2 and RL_TH2. Thus, T2 will not be affected in such way.

T2 is not only used for timer and counter but also Baud Rate Generator (BRG) for serial port and programmable clock source.

重装初始值，当 TH2、TL2 构成的 16 位计数器计满溢出时，RL_TH2，RL_TL2 的值自动装入 TH2、TL2 中。RL_TH2 与 TH2 共享同一个地址，RL_TL2 与 TL2 共享同一个地址。当 TR2＝0 时，对 TH2、TL2 寄存器写入数据时，也会同时写入 RL_TH2、RL_TL2 寄存器中；当 TR2＝1 时，对 TH2、TL2 写入数据时，只写入 RL_TH2、RL_TL2 寄存器中，而不会写入 TH2、TL2 寄存器中，这样不会影响 T2 的正常计数。

T2 可以当定时器用和计数器用，也可作为串口的波特率发生器和可编程时钟输出源。

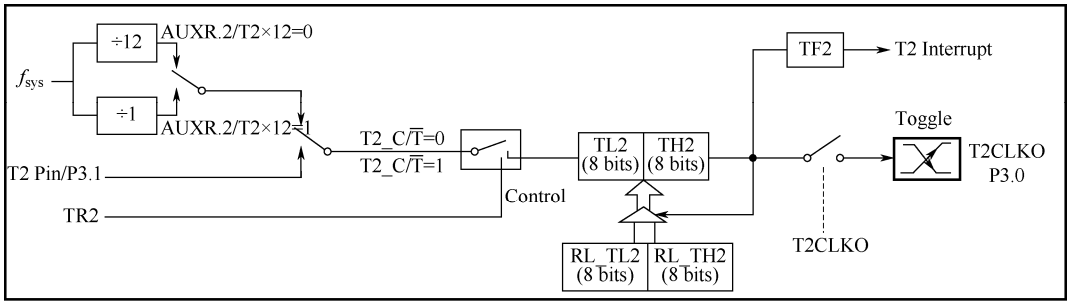


Figure 6.3 Functional Block Diagram of Timer T2

定时器 T2 的原理框图

6.2.2 Control Register for Timer/Counter 2
定时/计数器 T2 的控制寄存器

The status registers of internal Timer/Counter 2 in STC15W201S MCU are T2H and T2L. The control and manage of T2 is undertaken by SFRs AUXR, INT_CLKO (AUXR2) and IE2. The SFRs related to T2 are shown in Table 6.5.

STC15W201S 单片机内部定时/计数器 T2 状态寄存器是 T2H、T2L，T2 的控制与管理由特殊功能寄存器 AUXR、INT_CLKO（AUXR2）、IE2 承担。与定时/计数器 T2 有关的特殊功能寄存器如表 6.5 所示。

Table 6.5　SFRs related to T2

	Address	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value
T2H	D6H	T2 high 8-bit								0000.0000
T2L	D7H	T2 low 8-bit								0000.0000
AUXR	8EH	T0x12	-	UART_M0x6	T2R	T2_C/$\overline{T}$	T2x12	-	S1ST2	0x00.00x0
INT_CLKO	8FH	-	EX4	EX3	EX2	-	T2CLKO	-	T0CLKO	x000.x0x0
IE2	AFH	-	-	-	-	-	ET2	-	-	xxxx.x0xx

(1) T2R: Timer 2 run control bit.

0: Not run Timer 2.

1: Run Timer 2.

(2) T2_C / $\overline{T}$: Counter or timer selector.

0: As Timer (namely count on the internal system clock or the clock 1/12 frequency dividing).

1: As Counter (namely count on the external pulse input from P3.1).

(3) T2x12: Timer 2 clock source bit.

0: The clock source of Timer 2 is the system clock 1/12 frequency dividing.

1: The clock source of Timer 2 is system clock.

(4) T2CLKO: The output control bit of Timer/Counter 2.

0: P3.0 is not configured for Timer 2 programmable clock output port.

1: P3.0 is configured for Timer 2 programmable clock output port.

(5) ET2: Timer 2 interrupt enable bit.

0: Timer 2 interrupt would be disabled.

1: Timer 2 interrupt would be enabled.

（1）T2R：定时/计数器 T2 运行控制位。

0：T2 停止运行。

1：T2 运行。

（2）T2_C / $\overline{T}$ ：定时、计数选择控制位。

0：定时/计数器 T2 为定时状态，计数脉冲为系统时钟或系统时钟的 12 分频信号。

1：定时/计数器 T2 为计数状态，计数脉冲为 P3.1 输入引脚的脉冲信号。

（3）T2x12：定时脉冲的选择控制位。

0：定时脉冲为系统时钟的 12 分频信号。

1：定时脉冲为系统时钟信号。

（4）T2CLKO：定时/计数器 T2 时钟输出控制位。

0：不允许 P3.0 配置为定时/计数器 T2 的时钟输出口。

1：P3.0 配置为定时/计数器 T2 的时钟输出口。

（5）ET2：定时/计数器 T2 的中断允许位。

0：禁止定时/计数器 T2 中断。

1：允许定时/计数器 T2 中断。

6.3　Application Notes for Timer in Practice
定时/计数器的应用举例

The timer/counter of STC15W201S MCU is programmable. Thus, it should be initialized by software before timing or counting.

STC15W201S 单片机的定时/计数器是可编程的。因此，在利用定时/计数器进行定时或计数之前，先要通过软件对

The flow of application program is shown as below:

(1) Assign a value to TMOD to set working mode of T0.

(2) Assign a value to AUXR to set timing pulse division factor (default 1/12 frequency dividing, the same as traditional 8051).

(3) Calculate the initial values, and write them into TH0, TL0 or TH2, TL2.

(4) In interrupt mode, assign a value to IE or IE2, enable the interrupt, and manipulate IP if necessary to determine the priority of all interruption.

(5) Set TR0 or TR2, start T0 or T2 timing or counting.

(6) Detect the overflow flag of timer/counter; when the timer is timeout, arrange tasks in schedule.

它进行初始化。

定时/计数器应用编程流程如下：

（1）对 TMOD 赋值，以确定 T0 的工作方式。

（2）对 AUXR 赋值，确定定时脉冲的分频系数，默认为 12 分频，与传统 8051 单片机兼容。

（3）计算初值，并将其写入 TH0、TL0 或 TH2、TL2。

（4）为中断方式时，则对 IE 或 IE2 赋值，开放中断，必要时，还须对 IP 操作，确定各中断源的优先等级。

（5）置位 TR0 或 TR2，启动 T0 或 T2 开始定时或计数。

（6）检测定时/计数器的溢出标志（查询或中断），定时到了，安排之后的事务。

6.3.1 Timing Application of Timer/Counter (T0/T2)
定时/计数器（T0/T2）的定时应用

E.g. Light up LEDs circularly. Firstly, light them up from left to right, and then from right to left with an interval of 1 second. Setting the interval by timer/counter of MCU is required.

Here gives the solution:

The hardware circuit, using port P1 to output drive, active low, is simple, as shown in Figure 6.4.

System adopts a 12MHz oscillator, and the factor of frequency dividing is 12, that is, the period of clock is 1μs. By using Timer T0 Mode 0, it can archieve 65.536ms, which is far from 1s. Therefore, an approach using timing plus accumulation is required to make T0 implement 50ms timing. 1s is obtained after twenty times' accumulations.

例　使信号灯循环点亮，首先按从左至右轮流点亮，再按从右至左轮流点亮，每个信号灯点亮的时间间隔为 1 秒。时间间隔要求用单片机定时/计数器定时实现。

解：硬件电路比较简单，采用 P1 口输出驱动电平，低电平有效。电路如图 6.4 所示。

系统采用 12MHz 晶振，分频系数为 12，即定时时钟周期为 1μs，采用定时器 T0 方式 0 定时最多可实现 65.536ms，离 1s 还差很远。因此，需要采用定时加累计的方法实现，让定时器 T0 实现 50ms 定时，累计 20 次时即为 1s。

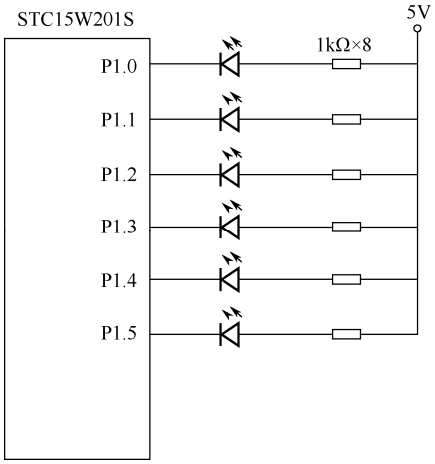


Figure 6.4　LED Display Circuit　流水灯显示电路

The reference C51 program is shown as below:　C51 参考程序（time.c）如下：

```
#include <REG51.H>                  // Include standard 51 head file   包含 51 标准文件头
#include <intrins.h>                /* Include subroutines of rotate left and rotate right
                                       包含循环左移、右移子函数*/
#define  uchar  unsigned  char
#define  uint  unsigned  int
uchar   LED = 0xfe;
/*---------------------- Timing Subroutines Using T0------------------*/
void DELAY(void)                    /* Delay Function 1s   延时函数  1s*/
{
    uint   i = 0;                   /* Define 50ms accumulation variable
                                       定义 50ms 累计变量*/
    TMOD = 0x00;                    // T0 working mode 0   T0 工作模式 0
    TH0 = 0x3c;                     // (65536-50000)/256
    TL0 = 0xb0;                     // (65536-50000)%256
    TR0 = 1;                        // Start Timer T0   启动定时器 T0
    while(i<20)                     /* Timeout after twenty times' accumulations
                                       累计 20 次后，定时结束*/
    {
        while(TF0==0);              // Check the overflow flag of T0   查询 T0 溢出标志
        TF0 = 0;                    // Clear the overflow flag of T0   清零  T0 溢出标志
        i++;                        /* Got 1s after twenty times' accumulations
                                       50ms 累计变量加 1*/
    }
    TR0 = 0;                        // Stop timer T0   关闭定时器 T0
```

```
}
/*--------------------Rotate Left Subroutine-----------------*/
void Left_Shift(void)
{
    P1 = LED;
    DELAY();                    // Delay 1s   延时一秒
    LED = _crol_(LED,1);        // Rotate left for 1bit   循环左移一位
}
/*-------------------- Rotate Right Subroutine ----------------*/
void Right_Shift(void)
{
    P1 = LED;
    LED = _cror_(LED,1);        // Rotate right for 1bit   循环右移一位
    DELAY();                    // Delay 1s   延时一秒
}

/*--------------------Main Function-----------------*/
void main(void)
{
    uchar   j;
    while(1)
    {
        for(j=0;j<5;i++)
        {
            Left_Shift ();
        }
        for(j=0;j<5;i++)
        {
            Right_Shift ();
        }
    }
}
```

6.3.2 Counting Application of Timer/Counter (T0/T2)
定时/计数器（T0/T2）的计数应用

E.g. Flip the status of LED under the control of MCU by five signal pulses input continuously. Implemented by timer/counter of MCU is required.

例　连续输入 5 个单次脉冲使单片机控制的 LED 灯状态翻转一次。要求用单片机定时/计数器计数功能

Here gives the solution:

The hardware circuit using T2 is shown as Figure 6.5.

实现。

解：采用 T2 实现，硬件如图 6.5 所示。

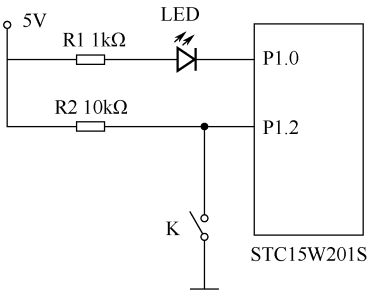


Figure 6.5　Counter Control in Signal Light　信号灯的计数控制

Adopting the counting mode of T0 Mode 0, assign an initial value of FFFBH. Check the overflow flag after five input pulses and then control the P1.0 LED.

The reference C51 program is shown as below:

采用 T0 的方式 0 的计数方式，初始值设置为 FFFBH，当输入 5 个脉冲时，通过查询 TF0 标志，进而对 P1.0 LED 灯进行控制。

C51 参考程序（count.c）如下：

```
#include<stc15w201s.h>      //须预先对 Keil 添加 stc15w201s.h 头文件
sbit    led = P1^0;
void    T0_init (void)
{
      TH0 = 0xff;           // Overflow after five pulses    5 个脉冲以后溢出
      TL0 = 0xfb;
      TR0=1;                // Start T2 counting    启动 T2 计数
   }
void main(void)
{
      T0_init();
      while(1)
      {
            while(TF0==0);  /* Check overflow flag. Wait until overflow    不断查询是否
                            溢出，没有溢出就等待，直到溢出*/
             TF0 = 0;
            led = ~led;     /* Clear the flag after overflow. Reverse led
                            清空溢出标志，led 取反*/
      }
}
```

6.4 Programmable Clock Output 可编程时钟输出功能

6.4.1 Programmable Clock 可编程时钟

In most of occasions, clock provided to peripheral is required in application system. It can reduce not only the cost of system but also the space of PCB when a programmable clock is provided by MCU. The clock can be shut down when clock is not required, which can reduce the power consumption and the electromagnetic radiation from clock as well. Two programmable clock output pins T0CLKO (P1.4) and T2CLKO (P3.0) are added in STC15W201S, where the clock frequency output from T0CLKO is controlled by the Timer/Counter 0 which should work in Mode 0 or Mode 2 (data auto-reload mode), and the clock frequency output from T2CLKO is controlled by the Timer/Counter 2.

很多实际应用系统需要给外围器件提供时钟，如果单片机能提供可编程时钟输出功能，既可以降低系统成本，也可缩小 PCB 的面积；当不需要时钟输出时，还可关闭时钟输出，这样不但降低了系统的功耗，而且减轻时钟对外的电磁辐射。STC15W201S 单片机增加了 T0CLKO（P1.4）和 T2CLKO（P3.0）2 个可编程时钟输出引脚。T0CLKO 的输出时钟频率由定时/计数器 T0 控制，T0 需要工作在方式 0 或方式 2（自动重装数据模式）。T2CLKO 的输出时钟频率由定时/计数器 T2 控制。

1. Control of Programmable Clock Output 1. 可编程时钟输出的控制

The output of two programmable clocks is controlled by SFR INT_CLKO, whose definition is shown in Table 6.6.

2 个可编程时钟输出由 INT_CLKO 特殊功能寄存器进控制，INT_CLKO 特殊功能寄存器的定义如表 6.6 所示。

Table 6.6 INT_CLKO

INT_CLKO	Address	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value
	8FH	-	EX4	EX3	EX2	-	T2CLKO	-	T0CLKO	x000.x0x0

1) T0CLKO: Control Bit for Timer/Counter 0 Clock Output

0: P1.4 (T0CLKO) is not configured as Timer/Counter 0 clock output port.

1: P1.4 (T0CLKO) is configured as Timer/Counter 0 clock output port.

1）T0CLKO：定时/计数器 T0 时钟输出控制位

0：不允许 P1.4 (T0CLKO)配置为定时/计数器 T0 的时钟输出口。

1：P1.4 (T0CLKO)配置为定时/计数器 T0 的时钟输出口。

2) T2CLKO: Control Bit for Timer/Counter 2 Clock Output

0: P3.0 (T2CLKO) is not configured as

2）T2CLKO：定时/计数器 T2 时钟输出控制位

0：不允许 P3.0 (T2CLKO)配置为

Timer/Counter 2 clock output port.

1: P3.0 (T2CLKO) is configured as Timer/Counter 2 clock output port.

定时/计数器 T2 的时钟输出口。

1：P3.0 (T2CLKO)配置为定时/计数器 T2 的时钟输出口。

2. Calculation of Programmable Clock's Frequency

2. 可编程时钟输出频率的计算

The frequency of programmable clock is the divided-by-2 frequency of timer/counter's overflow rate.

可编程时钟输出频率为定时/计数器溢出率的二分频信号。

1) T0 Output Clock

1）T0 输出时钟

Frequency of T0 output clock (T0CLKO) = T0 overflow rate /2.

T0 输出时钟（T0CLKO）频率＝（1/2）T0 溢出率。

If T0 works in Mode 0 timing,

若 T0 工作在方式 0 定时状态，则

设[RL_TH0, RL_TL0]=x，则

(T0x12)=0:

$$\text{T0CLKO frequency} = (f_{SYS}/12)/(65536 - x)/2 = f_{SYS}/[24\times(65536 - x)]$$

(T0x12)=1:

$$\text{T0CLKO frequency} = f_{SYS}/(65536 - x)/2 = f_{SYS}/[2\times(65536 - x)]$$

If T0 works in Mode 0 counting,

若 T0 工作在方式 0 计数状态，则

设[RL_TH0, RL_TL0]=x，则

$$\text{T0CLKO frequency} = (\text{T0\_PIN\_CLK})/(65536 - x)/2 = (\text{T0\_PIN\_CLK})/[2\times(65536 - x)]$$

NOTE：T0_PIN_CLK is the frequency of pulse on input pin T0 (P1.2) which is used for Timer/Counter 0.

注：T0_PIN_CLK 为定时/计数器 T0 的计数输入引脚 T0（P1.2）输入脉冲的频率。

2) T2 Output Clock

2）T2 输出时钟

Frequency of T2 output clock (T2CLKO) = T2 overflow rate 2.

T2 输出时钟（T2CLKO）频率＝（1/2）T2 溢出率。

The calculation related refers to T0.

相关计算参见 T0。

NOTE：T2_PIN_CLK is the frequency of pulse on input pin T2 (P3.1) which is used for Timer/Counter 2.

注：T2_PIN_CLK 为定时/计数器 T2 的计数输入引脚 T2（P3.1）输入脉冲的频率。

6.4.2 Application Instance for Programmable Clock 可编程时钟的应用举例

E.g. Output 115.2kHz, 51.2kHz clock from P3.0, P1.4 respectively.

例　编程在 P3.0、P1.4 引脚上分别输出 115.2kHz、51.2kHz 的时钟信号。

Here gives the solution:

Let the clock frequency be 12MHz, T0 works in Mode 0 timing, and no frequency division. Timing pulse frequency for timers is equal to system clock frequency, that is (T0x12) = (T2x12) = 1. Calculate the initial value of timers according to the calculation of programmable clock output above:

解：设系统时钟频率为 12MHz，T0 工作在方式 0 定时状态，且工作在无分频模式，即各定时器的定时脉冲频率等于系统时钟频率，即 T0x12＝T2x12＝1 ;根据前面可编程时钟输出频率的计算公式，计算各定时器的定时初始值：

（T2H）=FFH，（T2L）=CCH，
（TH0）=FFH，（TL0）=8BH

The reference C51 program is shown as below:

C51 参考程序如下：

```
#include<reg51.h>
sfr   INT_CLKO = 0x8F;   /*Generate head file "stc15w201s.h"directly by STC-ISP online
                           program software    可从 STC-ISP 在线编程软件中直接生
                           成 STC15W201S 单片机的头文件 stc15w201s.h*/
sfr   AUXR = 0x8E;      /* Change "reg51.h" to "stc15w201s.h", and needn't any definition
                           for new SFR   把 "reg51.h" 改为 "stc15w201s.h"，此时，不
                           需要对新增特殊功能寄存器进行定义了*/
sfr   T2H = 0XD6;
sfr   T2L = 0Xd7;
main()
{
      TMOD = 0x00;
      AUXR = (AUXR|0x80);              /* T0 works in no frequency division mode
                                          T0 工作在无分频模式*/
      AUXR = (AUXR|0x04) ;             /* T2 works in no frequency division mode
                                          T2 工作在无分频模式*/
      T2H = 0xFF;                      /* Assign initial values to T2, T0
                                          给 T2、T0 定时器设置初值*/
      T2L = 0xCC;
      TH0 = 0xFF;
      TL0 = 0x8B;
      INT_CLKO  =  (WAKE_CLKO|0x05);     /*Enable T0, T2 output clock
                                           允许 T0、T2 输出时钟信号*/
      TR0 = 1;                           //Start T0   启动 T0
      AUXR = (AUXR|0x10) ;               // Start T2   启动 T2
      while(1);                          // Infinite loop   无限循环
}
```

Summary 本章小结

There are two universal programmable timer/counter T0 and T2 in STC15W201S. The core circuit of T0 or T2 is 16-bit accumulator, corresponding to the 16-bit register [TH0, TL0] and register [T2H, T2L] in SFR. $C/\overline{T}$ in TMOD is used for setting the timing mode or counting mode of the Timer/Counter 0. The difference between timing mode and counting mode lies in the counting pulse. The counting pulse for timer is internal system clock or its divided-by-12 clock, while the counting pulse for counter comes from input pulse from external counting pin. There are four working modes for T0, no matter it is working as timer or counter, which is determined by M1 and M0. That is

Ml/M0=0/0: Mode 0. A 16bit initial value reloadable timer/counter;

Ml/M0=0/1: Mode 1. A 16bit initial timer/counter;

Ml/M0=1/0: Mode 2. An 8bit initial value auto-reloadable timer/counter;

Ml/M0=1/1: Mode 3. T0 is divided into two separate 8bit timer/counter.

From the perspective of function, Mode 0 contains all the functions of Mode 1 and Mode 2. Mode 3 is rarely used in practice, thus, Mode 0 is mostly used.

The start and stop of Timer/Counter 0 is controlled by the GATE in TMOD and TR0 of TCON. When GATE is "0", the start and stop of Timer/Counter 0 is controlled by TR0 only, while when GATE is "1", the start and stop of Timer/Counter 0 is controlled by TR0 **and an input from** $\overline{\text{INT1}}$ together.

Timer/Counter 0 can provide a programmable clock, whose frequency is the divided-by-2 frequency of timer/counter's overflow rate.

The circuit structure and management of Timer/Counter 2 in STC15W201S MCU are the same as T0 in general, but the working mode of T2 is fixed at 16-bit initial value auto-reloadable mode.

Two programmable clock output pins T0CLKO (P1.4) and T2CLKO (P3.0) are added in STC15W201S, where the clock frequency output from T0CLKO is controlled by the Timer/Counter 0, and the clock frequency output from T2CLKO is controlled by the Timer/Counter 2. T2 is not only used for timer and counter but also Baud Rate Generator (BRG) for serial port and programmable clock source.

In general, there are Watchdog Timer and Power-down Wake-up Special Timer in STC15W201S. These timers will be introduced in corresponding chapters.

Exercise 练习题

1. Completions

1) There are_____16-bit timer/counter.

2) The external pulse count input pin of T0 timer/counter is_____, and programmable clock output pin is_____.

3) The external pulse count input pin of T2 timer/counter is_____, and programmable clock output pin is_____.

4) The core circuit of timer/counter of STC15W201S MCU is_____. When T0 is working at timer mode, the pulse count of counting circuit is_____; when T0 is working at counter mode, the pulse count of counting circuit is_____.

5) The overflow flag of T0 timer/counter is_____, and its control bit of start-stop is_____.

6) The overflow flag of T2 timer/counter is_____, and its control bit of start-stop is_____.

7) There are_____ working mode in T0, while _____ in T2, and the working mode options are_____ and _____.

2. Choice Questions

1) T0 is working at MODE_____ and_____status.

A. 2, timing　　B. 1, timing

C. 1, counting　　D. 0, timing

2) The pulse count of T0 is_____ when TMOD= 00H and T0x12= 1.

A. System clock

B. System clock after 12-frequency-division

C. Signal from P3.4 pin

D. Signal from P3.5 pin

3) T0 will be active when TMOD=08H and _____.

A. TR0=1

B. TR1=1

C. TR0=1 and INT0 pin(P3.2) is active high

D. TR0=1 and INT1 pin(P3.3) is active high

4) The values of TH0, TL0, RL_TH0, RL_TL0 will be_____after the execution of "*TH0=0x3c;TL0= 0xb0*;" when TH0=01H, TL0=22H, TR0=1.

A. 3CH, B0H, 3CH, B0H

B. 01H, 22H, 3CH, B0H

C. 3CH, B0H, unchanged, unchanged

D. 01H, 22H, unchanged, unchanged

5) The values of TH0, TL0, RL_TH0, RL_TL0 will be_____ after the execution of "*TH0=0x3c;TL0= 0xb0*; " when TH0=01H, TL0=22H, TR0=0.

A. 3CH, B0H, 3CH, B0H

B. 01H, 22H, 3CH, B0H

C. 3CH, B0H, unchanged, unchanged

D. 01H, 22H, unchanged, unchanged

3. True or False

1) The core circuit of STC15W201S MCU timer/counter is counting circuit. ( )

2) The pulse count is system clock when the timer/counter in STC15W201S MCU is working at timing mode. ( )

3) The interrupt request flag of STC15W201S MCU timer/counter T0 is TF0. ( )

4) The overflow flag and the interrupt request flag of STC15W201S MCU timer/counter are different. ( )

5) The overflow flag of STC15W201S MCU timer/counter T2 is implicit. ( )

6) The start-stop of STC15W201S MCU timer/counter T0 is controlled by TR0 only. ( )

7) The STC15W201S MCU timer/counter T2 is fixed to a 16-bit initial value reloadable timer/counter. ( )

4. Question and Answer

1) What are the differences and similarities between timing and counting mode in STC15W201S MCU timer/counter?

2) What's the control principle of the start-stop of STC15W201S MCU timer/counter?

3) What's the computing formula when the T0 is working at Mode 0?

4) What should the initial value of T0 be when TMOD=00H, T0x12=1 and T0 timing 10ms?

5) What's the difference of the assignment between TH0 and TL when TR0=1 and TR0=0?

Unit 7　Interrupt System of STC15W201S

第 7 单元　STC15W201S 单片机的中断系统

Interrupt which was issued in the 1950s, is an important technology in computer. It is concerned with both hardware and software. With the help of interrupt, tasks in computer became more flexible and more efficient. The physical base for the management schedule of modern computer operating system is the plentiful interrupt functions and perfect interrupt system. Conflict occurs when CPU faces multi-tasking, while interrupt is a resource sharing technology essentially. The appearance of interrupt technology improves the development and application of computer greatly. Therefore, the competence of interrupt function is the key indicator which measures whether the computer is faultless or not.

中断的概念是在 20 世纪 50 年代中期提出的，是计算机中一个很重要的技术，它既和硬件有关，也和软件有关。正是因为有了中断技术，才使得计算机的工作更加灵活、效率更高。现代计算机中操作系统实现的管理调度，其物质基础就是丰富的中断功能和完善的中断系统。一个 CPU 资源要面向多个任务，导致资源竞争，而中断技术实质上是一种资源共享技术。中断技术的出现使得计算机的发展和应用大大地推进了一步。所以，中断功能的强弱已成为衡量一台计算机功能完善与否的重要指标。

7.1　Overview of Interrupt System　中断系统概述

7.1.1　Some Concepts of Interrupt System　中断系统的几个概念

1．Interrupt

In systems programming, an interrupt is a signal to the processor emitted by hardware indicating an external

1．中断

所谓中断是指程序执行过程中，允许外部或内部事件通

or internal event that needs immediate attention. An interrupt alerts the processor to a high-priority condition requiring the interruption of the current code the processor is executing (the current thread). The CPU responds by suspending its current activities, saving its state, and executing an interrupt handling also known as an Interrupt Service Routine (ISR) to deal with the event. This interruption is temporary, and after the interrupt handling finishes, the CPU resumes execution of the previous thread. The flow of interrupt response is shown in Figure7.1(a), where the multi-interrupt is shown in Figure 7.1(b).

A full interrupt process consists of 4 steps: interrupt request, interrupt response, interrupt service and return from interrupt (RETI).

Take the following case for example: when a manager is handling files, a phone comes (interrupt request), he has to mark the current state (breakpoint address, address for return as well), pause the current work to answer the call (interrupt response), handle the request from call (interrupt service), and then, calm down (return to the previous state), and go on handling the files (interrupt return).

过硬件打断程序的执行，使其转向到处理外部或内部事件的中断服务程序中去，完成中断服务程序后，CPU 返回继续执行被打断的程序。如图 7.1 所示为中断响应过程的示意图，其中，图 7.1（b）为多重中断的中断响应过程。

一个完整的中断过程包括 4 个步骤 :中断请求、中断响应、中断服务与中断返回。

打个比方，当一位经理正处理文件时，电话铃响了（中断请求），不得不在文件上画一个记号（断点地址，即返回地址），暂停工作，去接电话（响应中断），并处理“电话请求”（中断服务），然后，再静下心来（恢复中断前状态），接着处理文件（中断返回）……

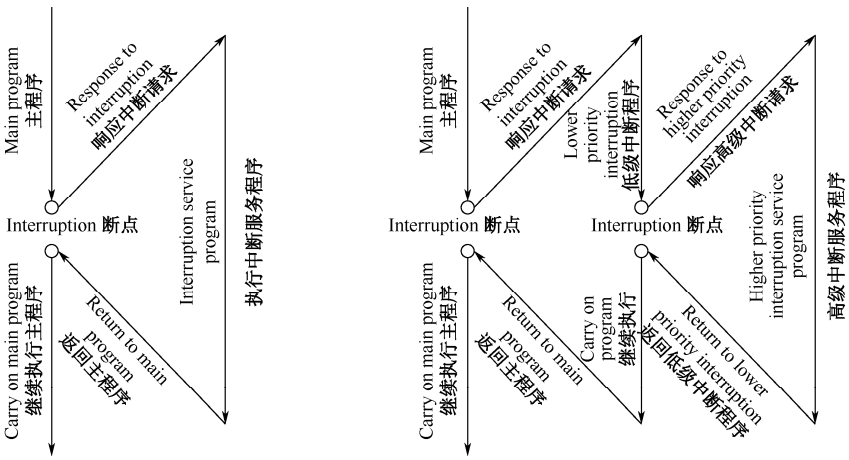


Figure 7.1　Diagram of Interrupt Response Process

中断响应过程示意图

2. Interrupt Source

The source causes interruption of CPU. The handling request sent to CPU from interrupt source is called Interrupt Request.

3. Interrupt Priority

If there are some interrupt requests at the same time, there occurs a problem that to which interrupt the CPU should response first. Thus, priority for interrupt is established. The high priority interrupt gets response preferentially.

4. Interrupt Nesting

A high priority interrupt could break the low priority interrupt service routine. The CPU suspends the current low priority interrupt response activities to deal with high priority interrupt. After the interrupt handling finishes, the CPU resumes execution of the previous low priority interrupt ISR. That is interrupt nesting, which is shown in Figure 7.1 (b).

2. 中断源

引起 CPU 中断的根源或原因，称为中断源。中断源向 CPU 提出的处理请求，称为中断请求或中断申请。

3. 中断优先级（权）

当有几个中断源同时申请中断时，那么就存在 CPU 先响应哪个中断请求的问题，为此，CPU 要对各中断源确定一个优先等级，称为中断优先权。中断优先权高的中断请求被优先响应。

4. 中断嵌套

中断优先级高的中断请求可以中断 CPU 正在处理的优先级更低的中断服务程序，待完成了中断优先权高的中断服务程序之后，再继续执行被打断的优先级低的中断服务程序，这就是中断嵌套，如图 7.1（b）所示。

7.1.2 Superiority of Interrupt Technology 中断技术的优势

(1)Solve the contradiction between fast CPU and slow peripheral; make the CPU and peripheral work in parallel.

Because many peripherals work slowly in application system, interrupt is used to coordinate the work between fast CPU and slow peripherals.

(2)Handle a number of random parameters and information in control system.

Real-time control can be archieved by interrupt. The analysis and calculation task issued randomly by controlled object is required to be handled in time in real-time control. In automatic control system, the request of any controlled object is able to be sent to CPU in any

（1）解决了快速 CPU 和慢速外设之间的矛盾，可使 CPU 和外设并行工作。

由于应用系统的许多外部设备速度较慢，可以通过中断的方法来协调快速 CPU 与慢速外部设备之间的工作。

（2）可及时处理控制系统中许多随机参数和信息。

依靠中断技术能实现实时控制。实时控制要求计算机能及时完成被控对象随机提出的分析和计算任务。在自动控制系统中，要求各控制参量随机地在任何时刻可向计算机发出请求，CPU 必须快速响应、及时处理。

time, and CPU should make a response rapidly and handle in time.

(3)Have the ability to handle failure and improve the reliability.

Such factors as the external disturb and the design deficiency in hardware or software, hardware problem, calculating error and software failure would occur in practice. Computer can detect the errors in time and fix them automatically due to interrupt technology.

(4)Human-Computer Interaction.

Interrupt can be issued to MCU by key -board to interfere MCU in real time.

（3）具备了处理故障的能力，提高了机器自身的可靠性。

由于外界的干扰、硬件或软件设计中存在问题等因素，在实际运行中会出现硬件故障、运算错误、程序运行故障等，有了中断技术，计算机就能及时发现故障并自动处理。

（4）实现人机联系。

比如通过键盘向单片机发出中断请求，可以实时干预计算机的工作。

7.1.3 Problems Need to be Solved in Interrupt System 中断系统需要解决的问题

The implementation of interrupt technology relies on a perfect interrupt system. The problems need to be solved in interrupt system is such as below.

(1) A register is needed to record the interrupt request from interrupt source. That is interrupt request flag.

(2) The interrupt request signal can be masked. Mask and unmask to interrupt request signal can be managed flexibly.

(3) When an interrupt request comes, CPU can response rapidly by suspending the current task and turn to Interrupt Service Routine (ISR). CPU can return to previous task after ISR finishes.

(4) If there are some interrupt requests at the same time, interrupt priority is managed to ensure that priority interrupt is able to receive response preferentially.

(5) A high priority interrupt could break the low priority ISR. The CPU suspends the current low priority interrupt response activities to deal

中断技术的实现依赖于一个完善的中断系统，一个中断系统需要解决的问题主要如下。

（1）当有中断请求时，需要有一个寄存器把中断源的中断请求记录下来，即中断请求标志。

（2）能够对中断请求信号进行屏蔽，灵活地对中断请求信号实现屏蔽与允许的管理。

（3）当有中断请求时，CPU 能及时响应中断，停下正在执行的任务，自动转去处理中断服务子程序，中断服务处理后能返回到断点处继续处理原先的任务。

（4）当有多个中断源同时申请中断时，应能优先响应优先权高的中断源，实现中断优先权的管理。

（5）当 CPU 正在执行低优先级中断源中断服务程序时，若优先级比它高的中断源也提出中断请求，要求能暂停

with high priority interrupt. After the interrupt handling finishes, the CPU resumes execution of the previous low priority interrupt ISR. That is interrupt nesting.

执行低优先级中断源的中断服务程序，转去执行更高优先级中断源的中断服务程序，实现中断嵌套，并能逐级正确返回原断点处。

7.2 Interrupt System 中断系统

A full interrupt process consists of 4 steps: interrupt request, interrupt response, interrupt service and return from interrupt (RETI). The interrupt system of STC15W201S is introduced below according to the interrupt process.

一个中断的工作过程包括中断请求、中断响应、中断服务与中断返回共四个阶段，下面按照中断系统工作过程介绍 STC15W201S 单片机的中断系统。

7.2.1 Interrupt Request 中断请求

As is shown in Figure 7.2, there are ten interrupt request sources and two priority levels in STC15W201S MCU, which can implement two-level nested interrupt service. The interrupt enabling registers IE, IE2, INT_CLKO and CMPCR1 residing in internal SFRs are used for controlling CPU whether to response interrupt request or not. The interrupt priority register IP is used for arranging priorities of all interrupt sources. The response order of interrupts in the same priority level is determined by the internal query logic.

如图 7.2 所示，STC15W201S 单片机的中断系统有 10 个中断源，2 个优先级，可实现二级中断服务嵌套。由片内特殊功能寄存器中的中断允许寄存器 IE、IE2、INT_CLKO、CMPCR1 控制 CPU 是否响应中断请求；由中断优先级寄存器 IP 安排各中断源的优先级；同一优先级内 2 个以上中断源同时提出中断请求时，由内部的查询逻辑确定其响应次序。

1. Interrupt Request Source

There are ten interrupt sources in STC15W201S, they are:

(1) External interrupt 0 (INT0): Input from pin P3.2. IT0 is used for setting the triggering method of interrupt request. When IT0 is “1”, INT0 is triggered Negative Edge; when IT0 is “0”, INT0 is triggered both Negative Edge and Positive Edge. Set IE0 flag and send request to

1. 中断源

STC15W201S 单片机有 10 个中断源，详述如下：

（1）外部中断 0（INT0）：中断请求信号由 P3.2 引脚输入。通过 IT0 来设置中断请求的触发方式。当 IT0 为“1”时，外部中断 0 为下降沿触发；当 IT0 为“0”时，无论是上升沿还是下降沿，都会引发外部中断 0。一旦输入

CPU once the input signal is valid.

信号有效，则置位 IE0 标志，向 CPU 申请中断。

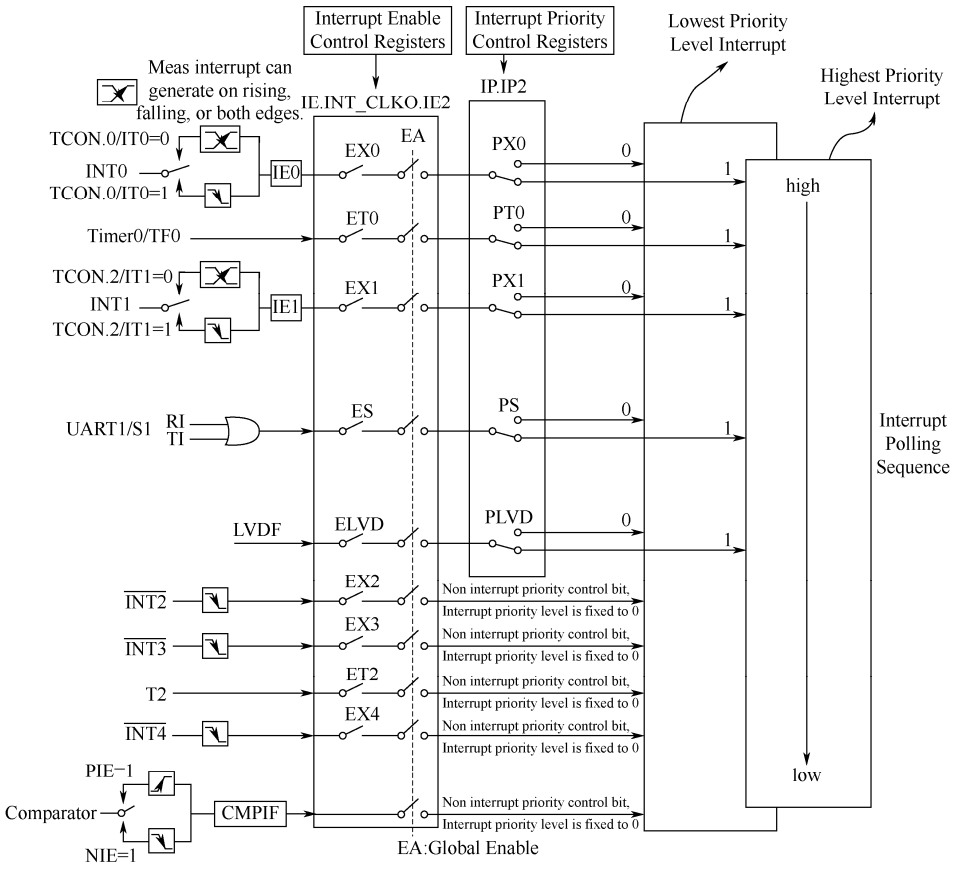


Figure 7.2 Interrupt Structure of STC15W210S

STC15W201S 单片机的中断系统结构图

(2) External interrupt 1 (INT1): Input from pin P3.3. IT1 is used for setting the triggering method of interrupt request. When IT1 is "1", INT1 is triggered by Negative Edge; when IT1 is "0", INT1 is triggered both by Negative Edge and Positive Edge. Set IE1 flag and send request to CPU once the input signal is valid.

(3) Timer/Counter 0 overflow interrupt: When the Timer/Counter 0 is overflow, the Timer/Counter 0 overflow interrupt request flag TF0 is set and an interrupt request is sent to CPU.

(4) Serial port interrupt: RI is set after a serial line frame is received or TI is set after a

（2）外部中断 1（INT1）：中断请求信号由 P3.3 引脚输入。通过 IT1 来设置中断请求的触发方式。当 IT1 为“ 1 ”时，外部中断 1 为下降沿触发；当 IT1 为“ 0 ”时，无论是上升沿还是下降沿，都会引发外部中断 1。一旦输入信号有效，则置位 IE1 标志，向 CPU 申请中断。

（3）定时/计数器 T0 溢出中断：当定时/计数器 T0 计数产生溢出时，定时/计数器 T0 中断请求标志位 TF0 置位，向 CPU 申请中断。

（4）串行口中断：当串行口接收完一串行帧时置位 RI 或发送完一串行帧

serial line frame is transmitted, and an interrupt request is sent to CPU.

(5) On-chip Low Voltage Detection (LVD) interrupt: LVDF is set when low voltage is detected. There is process of voltage rising in power-on reset, and LVDF is set when low voltage is detected in this stage, and an interrupt request is sent to CPU. After power-on reset, LVDF will be "1". It should be cleared first and detected again several system clocks later.

(6) External interrupt 2 ($\overline{\text{INT2}}$): Negative Edge active. An interrupt request is sent to CPU once the input signal is valid. The priority is fixed in low priority.

(7) External interrupt 3 ($\overline{\text{INT3}}$): Negative Edge active. An interrupt request is sent to CPU once the input signal is valid. The priority is fixed in low priority.

(8) Timer 2 (T2) interrupt: When the T2 is overflow, an interrupt request is sent to CPU. The priority is fixed in low priority.

(9) External interrupt 4 ($\overline{\text{INT4}}$): Negative Edge active. An interrupt request is sent to CPU once the input signal is valid. The priority is fixed in low priority.

(10) Comparator interrupt: When the comparison result is from low to high and PIE is set to "1", or, when the comparison result is from high to low and comparator interrupt flag is set to "1", an interrupt request is sent to CPU. The priority is fixed in low priority.

2. Interrupt Request Flag

The interrupt request flags of all the ten interrupts are stored in registers TCON, SCON, PCON and CMPCR1, which can be seen in Table 7.1. Where the interrupt request flags of external interrupt 2 ($\overline{\text{INT2}}$), external interrupt 3 ($\overline{\text{INT3}}$) and external interrupt 4 ($\overline{\text{INT4}}$), which are hidden

时置位 TI，向 CPU 申请中断。

（5）片内电源低电压检测中断：当检测到电源电压低于一定值，则置位 LVDF；上电复位时，由于电源电压上升有一个过程，低压检测电路会检测到低电压，置位 LVDF，向 CPU 申请中断。单片机上电复位后，LVDF=1，若要应用 LVDF，则先对 LVDF 清零，若干个系统时钟后，再检测 LVDF。

（6）外部中断 2（$\overline{\text{INT2}}$）：下降沿触发，一旦输入信号有效，则向 CPU 申请中断。中断优先级固定为低级。

（7）外部中断 3（$\overline{\text{INT3}}$）：下降沿触发，一旦输入信号有效，则向 CPU 申请中断。中断优先级固定为低级。

（8）定时器 T2 中断：当定时/计数器 T2 计数产生溢出时，即向 CPU 申请中断。中断优先级固定为低级。

（9）外部中断 4（$\overline{\text{INT4}}$）：下降沿触发，一旦输入信号有效，则向 CPU 申请中断。中断优先级固定为低级。

（10）比较器中断：当比较器的比较结果由低到高且 PIE 被设置为 1，或当比较器的比较结果由高到低且 NIE 被设置为 1，比较器中断标志位置 1，向 CPU 发出中断请求。中断优先级固定为低级。

2. 中断请求标志

STC15W201S 单片机的 10 个中断源的中断请求标志分别寄存在 TCON、SCON、PCON、CMPCR1 中，详见表 7.1。其中，外部中断 2（$\overline{\text{INT2}}$）、外部中断 3（$\overline{\text{INT3}}$）和外部中断 4（$\overline{\text{INT4}}$）的中断请求标志位被隐藏起

and invisible to user, will be cleared when the corresponding interrupt is served or EXn = 0 (n = 2, 3, 4). The interrupt request flag of T2 is hidden and invisible as well. When the interrupt of T2 is served or (ET2) = 0, this flag will be cleared automatically.

来了，对用户是不可见的，当相应的中断被响应后或 EXn=0（n=2、3、4）时，这些中断请求标志位会自动被清零。定时器 T2 的中断请求标志位也被隐藏起来了，对用户是不可见的，当 T2 的中断被响应后或 ET2=0 时，这些中断请求标志位会自动被清零。

Table 7.1　Interrupt Request Flags of The Ten Interrupt Sources in STC15W201S MCU

STC15W201S 单片机的 10 个中断源的中断请求标志位

	Address	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value
TCON	88H	-	-	TF0	TR0	IE1	IT1	IE0	IT0	0000 0000
SCON	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	0000 0000
PCON	87H	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL	0011 0000
CMPCR1	E6H	CMPEN	CMPIF	PIE	NIE	PIS	NIS	CMPOE	CMPRE	0000 1001

1)　Interrupt Request Flag in TCON

TCON is the Timer/Counter 0 and 1 control register, and lock the overflow interrupt request flag of T0 and T1 and the interrupt request of INT0 and INT1. The control bit related to interrupt is shown in Table 7.2.

1）TCON 寄存器中的中断请求标志

TCON 为定时器 T0 和 T1 的控制寄存器，同时也锁存 T0 和 T1 的溢出中断请求标志及外部中断 0 和 1 的中断请求标志等。与中断有关控制位如表 7.2 所示。

Table 7.2　TCON

TCON	Address	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value
	88H	-	-	TF0	TR0	IE1	IT1	IE0	IT0	0000 0000

(1)TF0: T0 overflow flag. Set by hardware on T0 overflow and send interrupt request to CPU. This flag can be cleared by software but is automatically by hardware when the corresponding interrupt service routine has been processed.

(2)IE1: External interrupt 1 request flag. Set by hardware when the input signal from pin INT1 meets the requirement of trigger. This flag is automatically cleared when the corresponding interrupt service routine has been processed.

（1）TF0：T0 的溢出中断请求标志。T0 被启动计数后，从初值进行加 1 计数，计满溢出后由硬件置位 TF0，同时向 CPU 发出中断请求，此标志一直保持到 CPU 响应中断后才由硬件自动清零。也可由软件查询该标志，并由软件清零。

（2）IE1：外部中断 1 的中断请求标志。当 INT1（P3.3）引脚的输入信号满足中断触发要求时，由硬件置位 IE1。中断响应后中断请求标志自动清零。

(3)IT1: External interrupt 1 type select bit.

0: INT1 is both rising and falling edges triggered. Set IE1 when rising or falling signal is detected in INT1 input by CPU. This flag is automatically cleared when the corresponding interrupt service routine has been processed.

1: INT1 is falling edge triggered only. Set IE1 when falling signal is detected in INT1 input by CPU. This flag is automatically cleared when the corresponding interrupt service routine has been processed.

(4)IE0: External interrupt 0 request flag. The operation is the same as IE1.

(5)IT0: External interrupt 0 type select bit. The operation is the same as IT1.

2) Interrupt Request Flag in SCON

The lower two bits, TI and RI, are used for latching the tramsmit interrupt request flag and receive interrupt request flag respectively, which are shown in Table 7.3.

（3）IT1：外部中断 1（INT1）中断触发方式控制位。

当 IT1= 0 时，外部中断 1 为上升沿触发和下降沿触发两种方式。在这种方式下，无论 CPU 检测到外部中断 1（INT1）输入端出现下降沿信号还是上升沿信号，都认为有中断申请，随即使 IE1 标志置位。中断响应后中断请求标志会自动清零，无须进行其他处理。

当 IT1= 1 时，外部中断 1 为下降沿触发方式。在这种方式下，若 CPU 检测到外部中断 1（INT1）输入端出现下降沿信号，则认为有中断申请，随即使 IE1 标志置位。中断响应后中断请求标志会自动清零，无须进行其他处理。

（4）IE0：外部中断 0 的中断请求标志。其操作功能与 IE1 相同。

（5）IT0：外部中断 0 的中断触发方式控制位。其操作功能与 IT1 相同。

2）SCON 寄存器中的中断请求标志

SCON 低 2 位 RI 和 TI 用于锁存串行口的接收中断请求标志和发送中断请求标志，见表 7.3。

Table 7.3 SCON

SCON	Address 地址	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value 复位值
	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	0000 0000

(1)TI: Transmit interrupt request flag. Start transmitting when a byte of data has been written into transmit buffer SBUF. Set by hardware once a serial line frame is sent. This flag will not be cleared when the corresponding interrupt service routine has been processed and must be cleared by software.

(2)RI: Receive interrupt request flag. Set by hardware once a serial line frame is sent when the serial port is enabling receiving. This flag will not be

（1）TI：串行口发送中断请求标志。CPU 将数据写入发送缓冲器 SBUF 时，就启动发送，每发送完一个串行帧，硬件将使 TI 置位。但 CPU 响应中断时并不清除 TI，必须由软件清除。

（2）RI：串行口接收中断请求标志。在串行口允许接收时，每接收完一个串行帧，硬件将使

cleared when the corresponding interrupt service routine has been processed and must be cleared by software.

RI 置位。同样，CPU 在响应中断时不会清除 RI，必须由软件清除。

3)　Interrupt Request Flag in PCON

3）PCON 寄存器中的中断请求标志

NOTE：PCON is power control register, where bit B5 is the interrupt request of LVD interrupt source, which is shown in Table 7.4.

注：PCON 是电源控制寄存器，其中 B5 位为 LVD 中断源的中断请求标志，见表 7.4。

LVDF: On-chip Low-voltage Detection Flag. Once the low voltage condition is detected, set LVDF. This flag must be cleared by software.

LVDF：片内电源低电压检测中断请求标志，当检测到低电压，则 LVDF=1；LVDF 中断请求标志应由软件清零。

Table 7.4　PCON

PCON	Address 地址	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value 复位值
	87H	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL	00110000B

4)　Interrupt request flag in CMPCR1

4）CMPCR1 中的中断请求标志

CMPIF: Comparator Interrupt Flag. If the comparing result has changed from low to high and PIE has been set to“1” simultaneously, or, if the comparing result has changed from high to low and NIE has been set to “1”, set CMPIF. This flag will not be cleared when the corresponding interrupt service routine has been processed and must be cleared by software, which is shown in Table 7.5.

CMPIF：比较器中断标志位。当比较器的比较结果由低到高且 PIE 被设置为 1，或当比较器的比较结果由高到低且 NIE 被设置为 1，比较器中断标志位置 1，但 CPU 响应中断时并不清零 CMPIF，必须由软件清零，见表 7.5。

Table 7.5　CMPCR1

CMPCR1	Address 地址	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value 复位值
	E6H	CMPEN	CMPIF	PIE	NIE	PIS	NIS	CMPOE	CMPRE	0000 1001

3．Interrupt Enable Control

3．中断允许的控制

There are two kinds of interrupts in computer interrupt system: one is unmaskable interrupt, the other is maskable interrupt. CPU must response to the unmaskable interrupt and

计算机中断系统有两种不同类型的中断：一类称为非屏蔽中断，另一类称为可屏蔽中断。对非屏蔽中断，用户不能用软件禁止，一旦有中断申请，

unmaskable interrupt can't be disabled by software. While the maskable interrupt can be disabled and enabled by software, which is called interrupt mask and interrupt open. The ten interrupt sources in STC15W201S are all maskable. There are four specific registers IE, IE2, INT_CLKO and CMPCR1 used for controlling the open or mask of all interrupt source. The detail can be seen in Table 7.6.

CPU 必须予以响应。对可屏蔽中断，用户则可以通过软件来控制是否允许某中断源的中断请求被响应，允许称中断开放，不允许称中断屏蔽。STC15W201S 单片机的 10 个中断源都是可屏蔽中断，其中断系统内部设有 4 个专用寄存器（IE、IE2、INT_CLKO、CMPCR1）用于控制 CPU 对各中断源的开放或屏蔽，详见表 7.6。

Table 7.6 Interrupt Enable Control Bits 中断开放控制位

	Addr	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value
IE	A8H	EA	ELVD	-	ES	-	EX1	ET0	EX0	00x0 0000
IE2	AFH	-	-	-	-	-	ET2	-	-	xxxx x000
INT_CLKO	8FH	-	EX4	EX3	EX2	-	T2CLKO	-	T0CLKO	x000 0000
CMPCR1	E6H	CMPEN	CMPIF	PIE	NIE	PIS	NIS	CMPOE	CMPRE	0000 1001

(1) EA: Control bit for all interrupt.

0: Disable all interrupt;

1: Each interrupt source would be individually enabled or disabled by setting or clearing its enable bit.

(2) ELVD: Low voltage detection interrupt enable bit.

0: Low voltage detection interrupt would be disabled;

1: Low voltage detection interrupt would be enabled.

(3) ES: Serial Port interrupt enable bit.

0: Interrupt would be disabled;

1: Interrupt would be enabled.

(4) EX1: External interrupt 1 enable bit.

1: Interrupt would be enabled;

0: Interrupt would be disabled.

(5) ET0: Timer 0 interrupt enable bit.

1: Interrupt would be enabled;

0: Interrupt would be disabled.

（1）EA：总中断开放控制位。EA= 0，屏蔽所有中断；EA= 1，开放 CPU 中断，各中断源的开放和屏蔽可通过相应的中断开放位单独加以控制。

（2）ELVD：片内电源低压检测中断（LVD）的中断开放位。ELVD= 0，屏蔽 LVD 中断；ELVD= 1，开放 LVD 中断。

（3）ES：串行口中断开放位。ES= 0，屏蔽串行口中断；ES= 1，开放串行口中断。

（4）EX1：外部中断 1 中断开放位。EX1= 1，开放外部中断 1 中断；EX1= 0，屏蔽外部中断 1 中断。

（5）ET0：定时器 T0 中断开放位。ET0= 1，开放 T0 中断；ET0= 0，屏蔽 T0 中断。

(6) EX0: External interrupt 0 enable bit.

1: Interrupt would be enabled;

0: Interrupt would be disabled.

(7) ET2: Timer 2 interrupt enable bit.

1: Interrupt would be enabled;

0: Interrupt would be disabled.

(8) EX4: External interrupt 4 enable bit.

1: Interrupt would be enabled;

0: Interrupt would be disabled.

(9) EX3: External interrupt 3 enable bit.

1: Interrupt would be enabled;

0: Interrupt would be disabled.

(10) EX2: External interrupt 2 enable bit.

1: Interrupt would be enabled;

0: Interrupt would be disabled.

(11) PIE, NIE: If the comparing result has changed from low to high and PIE has been set to "1"simultaneously, enable the comparator Interrupt. If the comparing result has changed from high to low and NIE has been set to "1", enable the comparator Interrupt.

4. Interrupt Priority Control

In STC15W201S MCU, all interrupts can be individually programmed to two levels priority interrupt that can form interrupt nesting, except External interrupt 2, External interrupt 3, Timer interrupt 2, External interrupt 4 and Serial Port interrupt, which are fixed at low priority. SFR IP is an interrupt priority control register used for setting interrupt priority which is shown in Table 7.7.

（6）EX0：外部中断 0 中断开放位。EX0= 1，开放外部中断 0 中断；EX0= 0，屏蔽外部中断 0 中断。

（7）ET2：定时器 T2 中断开放位。ET2= 1，开放 T2 中断；ET2= 0，屏蔽 T2 中断。

（8）EX4：外部中断 4 的中断开放位。EX4= 1，开放外部中断 4 中断；EX4= 0，屏蔽外部中断 4 中断。

（9）EX3：外部中断 3 的中断开放位。EX3= 1，开放外部中断 3 中断；EX3= 0，屏蔽外部中断 3 中断。

（10）EX2：外部中断 2 的中断开放位。EX2= 1，开放外部中断 2 中断；EX2= 0，屏蔽外部中断 2 中断。

（11）PIE、NIE：当 PIE 设置为 1 且比较器的比较结果由低到高时开放比较器中断；当 NIE 设置为 1 且比较器的比较结果由高到低时开放比较器中断。

4. 中断优先的控制

STC15W201S 单片机除外部中断 2、外部中断 3、定时器 T2 中断、外部中断 4 和串行口中断为固定低级优先级中断外，其他中断都具有 2 个中断优先级，可实现二级中断服务嵌套。IP 为中断优先级寄存器，锁存各中断源优先级控制位，详见表 7.7。

Table 7.7 Interrupt Priority Control Register 中断优先控制寄存器

IP	Address 地址	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value 复位值
	B8H	-	PLVD	-	PS	-	PX1	PT0	PX0	0000 0000

(1)PX0: External interrupt 0 priority control bit.

0: Interrupt is assigned low priority;

1: Interrupt is assigned high priority.

(2) PT0: Timer/Counter 0 interrupt priority control bit.

0: Interrupt is assigned low priority;

1: Interrupt is assigned high priority.

(3) PX1: External interrupt 1 priority control bit.

0: Interrupt is assigned low priority;

1: Interrupt is assigned high priority.

(4) PS: Serial Port interrupt priority control bit.

0: Interrupt is assigned low priority;

1: Interrupt is assigned high priority.

(5) PLVD: Low voltage detection interrupt priority control bit.

0: Interrupt is assigned low priority;

1: Interrupt is assigned high priority.

All the interrupt resources are set to low priority by clearing the priority control bits to “0” after reset.

If interrupt requests of the same priority level come simultaneously, an internal polling sequence determines which request could get response first through internal hardware query logic. The polling sequence is shown as below.

（1）PX0：外部中断 0 中断优先级控制位。

PX0= 0，外部中断 0 为低优先级中断；

PX0= 1，外部中断 0 为高优先级中断。

（2）PT0：定时/计数器 T0 中断的中断优先级控制位。

PT0= 0，定时/计数器 T0 中断为低优先级中断；

PT0= 1，定时/计数器 T0 中断为高优先级中断。

（3）PX1：外部中断 1 中断优先级控制位。

PX1= 0，外部中断 1 为低优先级中断；

PX1= 1，外部中断 1 为高优先级中断。

（4）PS：串行口中断的优先级控制位。

PS= 0，串行口中断为低优先级中断；

PS= 1，串行口中断为高优先级中断。

（5）PLVD：电源低电压检测中断的中断优先级控制位。

PLVD= 0, 电源低电压检测中断为低优先级中断；

PLVD= 1, 电源低电压检测中断为高优先级中断。

当系统复位后, 5 个中断优先权控制位全部清零，所有中断源均设定为低优先级中断。

如果几个同一优先级的中断源同时向 CPU 申请中断, CPU 通过内部硬件查询逻辑，按自然优先权顺序确定先响应哪个中断请求。自然优先权由内部硬件电路形成，排列如下。

Interrupt sources 中断源	
External interrupt 0	外部中断 0
Timer 0 interrupt	定时器 T0 中断
External interrupt 1	外部中断 1
Serial Port interrupt	串行口中断
LVD interrupt	LVD 中断
External interrupt 2	外部中断 2
External interrupt 3	外部中断 3
Timer 2 interrupt	定时器 T2 中断
External interrupt 4	外部中断 4
Comparator interrupt	比较器中断

Priority of interrupt in the same level
同级自然优先顺序

Highest 最高

Lowest 最低

7.2.2 Interrupt Response, Interrupt Service and Return From Interrupt 中断响应、中断服务与中断返回

The interrupt is that CPU responses to interrupt request, including breakpoint protection and jumping to vector address for ISR. CPU can't response interrupt at always, except that when response condition is fulfilled.

1. Timing of Interrupt Response

When the interrupt is enabled, CPU should response to the interrupt request except the cases below which will block the interrupt response and CPU response time in some degree.

(1)An interrupt of equal or higher priority level is already in progress.

(2)The instruction in progress is RETI (return from interrupt) or accessing registers related to interrupt such as IE or IP.

(3)The current instruction is not finished yet.

Any of these conditions above will abolish the interrupt query. CPU doesn't response to interrupt

中断响应是 CPU 对中断源中断请求的响应，包括保护断点和将程序转向中断响应的入口地址（通常称中断向量地址）。CPU 并非任何时刻都响应中断请求，而是在中断响应条件满足之后才会响应。

1. 中断响应时间问题

当在中断允许的条件下，中断源发出中断请求后，CPU 肯定会响应中断，但若有下列任何一种情况存在，则中断响应会受到阻断，会不同程度地增加 CPU 响应中断的时间。

（1）CPU 正在执行同级或高优先级的中断。

（2）正在执行 RETI（中断返回）指令或访问与中断有关的寄存器的指令，如访问 IE 和 IP 的指令。

（3）当前指令未执行完。

若存在上述任何一种情况，中断查询结果即被取消，CPU 不响应中

and go on querying in next instruction cycle. CPU will take response to interrupt in the next cycle when the condition is fulfilled.

In the last cycle of every instruction, CPU samples every interrupt and set corresponding interrupt flag: CPU will poll all flags in the last cycle of the next instruction cycle according to the priority polling sequence, if one of the flag is active, the corresponding interrupt will be served in the next instruction cycle according to the priority.

断请求而在下一指令周期继续查询，条件满足，CPU 在下一指令周期响应中断。

在每个指令周期的最后时刻，CPU 对各中断源采样，并设置相应的中断标志位：CPU 在下一个指令周期的最后时刻按优先级顺序查询各中断标志，如查到某个中断标志为 1，将在下一个指令周期按优先级的高低顺序进行处理。

2. Interrupt Response Process

The interrupt response process consists of breakpoint protection and jumping to the vector address for ISR.

The corresponding priority state register will be set to “1” when CPU responses to interrupt and an instruction Long Subroutine Call (LCALL) is generated by hardware which will first push the breakpoint address into stack, and then deliver the vector address of ISR to Program Counter (PC), which turns the process to ISR.

The vector addresses for all interrupt sources of STC15W201S are preseted by hardware, as shown in Table 7.8.

2. 中断响应过程

中断响应过程包括保护断点和将程序转向中断服务程序的入口地址。

CPU 响应中断时，将相应的优先级状态触发器置 1，然后由硬件自动产生一个长调用指令 LCALL，此指令首先把断点地址压入堆栈保护，再将中断服务程序的入口地址送入到程序计数器 PC，使程序转向相应的中断服务程序。

STC15W201S 单片机各中断源中断响应的入口地址由硬件事先设定，如表 7.8 所示。

Table 7.8 Vector Addresses for All Interrupt Sources (Interrupt Vector)

各中断源中断响应的入口地址（中断向量）

Interrupt Source 中断源	Vector Address (Interrupt Vector) 中断向量地址	Interrupt Number 中断号
External Interrupt 0 外部中断 0	0003H	0
Timer 0 Interrupt T0 中断	000BH	1
External Interrupt 1 外部中断 1	0013H	2
Serial Port Interrupt 串行口中断	0023H	4
LVD Interrupt 低压检测中断	0033H	6
External Interrupt 2 外部中断 2	0053H	10
External Interrupt 3 外部中断 3	005BH	11
Timer 2 Interrupt T2 中断	0063H	12
External Interrupt 4 外部中断 4	0083H	16
Comparator Interrupt 比较器中断	00ABH	21

An unconditional transfer instruction always will be arranged to vector address in practice to make program jump to the start address of ISR arranged by user.

使用时，通常在这些中断入口地址处存放一条无条件转移指令，使程序跳转到用户安排的中断服务程序的起始地址上去。

E.g.

例如：

```
ORG     000BH              ；T0 interrupt entrance   T0 中断入口
LJMP    T0_ISR             ；Jump to T0 ISR   转向 T0 中断服务程序
```

Interrupt number is used for distinguishing all interrupt sources in C program. Thus, in interrupt function, interrupt number and interrupt source is one-to-one mapping, which can't be confused.

其中，C 语言编程采用中断号来区分各中断源，在中断函数中中断号与各中断源是一一对应的，不能混淆。

E.g.

例如：

```
void   INT0_ISR（void）interrupt 0 {}          /*Interrupt function of External Interrupt 0
                                                外部中断 0 中断函数*/
void   Timer0_ISR（void）interrupt 1 {}        /*定时器 T0 中断函数*/
void   INT1_ISR（void）interrupt 2 {}          /*Interrupt function of External Interrupt 1
                                                外部中断 1 中断函数*/
void   UART_ISR（void）  interrupt 4 {}        /*Interrupt function of Serial Port Interrupt
                                                串行口中断函数*/
void   LVD_ISR（void）interrupt 6 {}           /*Interrupt function of LVD Interrupt
                                                LVD 中断函数*/
void   INT2_ISR（void）interrupt 10 {}         /*Interrupt function of External Interrupt 2
                                                外部中断 2 中断函数*/
void   INT3_ISR（void）interrupt11 {}          /*Interrupt function of External Interrupt 3
                                                外部中断 3 中断函数*/
void   Timer2_ISR（void）interrupt 12 {}       /*Interrupt function of T2   定时器 T2 中
                                                断函数*/
void   INT4_ISR（void）interrupt 16 {}         /*Interrupt function of External Interrupt 4
                                                外部中断 4 中断函数*/
void   CMP_ISR（void）interrupt 21 {}          /*Interrupt function of External Interrupt 0
                                                比较器中断的中断函数*/
```

NOTE：Interrupt function name is user defined, while interrupt number is fixed.
注：中断函数名是由用户自由定义的，但各中断源中断号是固定的。

3．Withdraw Interrupt

CPU enters ISR once response to interrupt request. CPU should withdraw interrupt request

3．中断请求标志的撤除问题

CPU 响应中断请求后即进入中断服务程序，在中断返回前，应撤除该中

before RETI, or it will retrigger interrupt again and error occurs. There are different ways for withdrawing interrupts.

1) Timer Interrupt Withdrawing

Timer/Counter 0 overflow interrupt flag TF0 will be cleared by hardware after CPU's response. No further operation needs.

However the interrupt request flag of Timer/Counter 2 is hidden, and invisible. These interrupt request flags will be cleared automatically after ISR.

2) Serial Port Interrupt Withdrawing

For Serial Port Interrupt, the interrupt request flag TI or RI won't be cleared by hardware after CPU's response. They must be cleared by software in ISR after distinguishing that the interrupt was caused by TI or RI.

3) External Interrupt 0～4 Withdrawing

The triggering ways of External Interrupt 0 and 1 can be configured by ITx (x=0, 1), with edge triggering. Interrupt flag IE0 or IE1 will be cleared by hardware after CPU's response. No further operation needed.

The interrupt request flags of external interrupt 2, 3 and 4 are hidden and invisible. These interrupt request flags will be cleared automatically after ISR.

4) LVD Interrupt Withdrawing

The interrupt flag of LVD Interrupt which won't be cleared automatically after response needs to be cleared by software.

5) Comparator Interrupt Withdrawing

The interrupt flag of Comparator Interrupt which won't be cleared automatically after response needs to be cleared by software.

断请求，否则，会重复引起中断而导致错误。STC15W201S 单片机各中断源中断请求撤除的方法各不相同。

1）定时器中断请求的撤除

对于定时/计数器 T0 溢出中断，CPU 在响应中断后即由硬件自动清零其中断标志位 TF0，不用采取其他措施。

然而，定时器 T2 中断的中断请求标志位被隐藏起来了，对用户是不可见的。当相应的中断服务程序执行后，这些中断请求标志位也会自动被清零。

2）串行口中断请求的撤除

对于串行口中断，CPU 在响应中断后，硬件不会自动清零中断请求标志位 TI 或 RI，必须在中断服务程序中，在判别出是 TI 还是 RI 引起的中断后，由软件将其清除。

3）外部中断 0～4 中断请求的撤除

外部中断 0 和外部中断 1 的触发方式可由 ITx（x=0，1）设置，属于边沿触发，CPU 在响应中断后由硬件自动清零其中断标志位 IE0 或 IE1，无须采取其他措施。

外部中断 2、3、4 的中断请求标志位被隐藏起来了，对用户是不可见的。当相应的中断服务程序执行后，这些中断请求标志位会自动被清零。

4）电源低电压检测中断请求的撤除

电源低电压检测中断的中断标志位在中断被响应后不会自动被清零，需要用软件清零。

5）比较器中断请求的撤除

比较器中断的中断请求标志位在中断被响应后不会自动被清零，需要用软件清零。

4. Interrupt Service and Return From Interrupt

Interrupt service and return from interrupt are implemented through executing Interrupt Service Routine (ISR). The ISR, which proceeds from vector address until RETI instruction is encountered, consists of Interrupt Context Saving, Interrupt Service, Interrupt Context Recovery and Return from Interrupt.

Interrupt Context Saving: Accumulator A, program status word registers PSW and other registers are used in both main program and ISR. However, the context stored in those registers will be destroyed when the registers are used in ISR, which will cause error after recovery from ISR. Thus, the context in the registers that needs saving will be pushed into stack before executing ISR, which is called Interrupt Context Saving.

Interrupt Service: Kernel of ISR. Interrupt source and request work here.

Interrupt Context Recovery: After ISR, the context that had been pushed in stack at Interrupt Context Saving stage is popped from stack in reverse order.

Return from Interrupt: CPU returns to the original location (breakpoint) in the main program after ISR. Return from Interrupt is executed by RETI instruction, which will pop the breakpoint address from stack into Program Counter (PC) , besides, inform interrupt system that interrupt service is accomplished and clear the priority status register simultaneously. Pay special attention to that RETI instruction can't be replaced by RET instruction.

In C51 programing, Interrupt Context Saving, Interrupt Context Recovery and Interrupt Context Recovery is not needed in interrupt function but the context of Interrupt Service.

4. 中断服务与中断返回

中断服务与中断返回是通过执行中断服务程序完成的。中断服务程序从中断入口地址开始执行，到返回指令“RETI”为止，一般包括四部分内容，其结构是：保护现场，中断服务，恢复现场与中断返回。

保护现场：通常，主程序和中断服务程序都会用到累加器 A、状态寄存器 PSW 及其他一些寄存器，当 CPU 进入中断服务程序用到上述寄存器时，会破坏原来存储在寄存器中的内容，一旦中断返回，将会导致主程序的混乱，因此，在进入中断服务程序后，一般要先保护现场，即用入栈操作指令将应保护寄存器的内容压入堆栈。

中断服务：中断服务程序的核心部分，是中断源、中断请求之所在。

恢复现场：在中断服务结束之后，中断返回之前，用出栈操作指令将保护现场中压入堆栈的内容弹回到相应的寄存器中，注意弹出顺序必须与压入顺序相反。

中断返回：中断返回是指中断服务完成后，计算机返回原来断开的位置（即断点），继续执行原来的程序。中断返回由中断返回指令 RETI 来实现。该指令的功能是把断点地址从堆栈中弹出，送回到程序计数器 PC，此外，还通知中断系统已完成中断处理，并同时清除优先级状态寄存器。特别要注意不能用“RET”指令代替“RETI”指令。

在 C51 编程中，只须将中断服务内容放在中断函数中即可，无须考虑保护现场和恢复现场，以及中断返回事项。

7.2.3 Applications of Interrupt 中断应用举例

E.g. Timing using Timer/Counter 0 Mode 0, output a square wave with a cycle of 10ms, implement by interrupt.

Solution: The detection of overflow flag of T0 changes to interrupt from query.

The reference C51 program is shown below:

例 利用定时/计数器 T0 方式 0 实现定时，在 P1.0 引脚输出周期为 10ms 的方波，采用中断方式实现。

解：本题要求 T0 的计满溢出标志的检测由查询方式改为中断方式。

C51 参考程序如下：

```
#include<reg51.h>
#define uchar unsigned char
#define uint unsigned int
sbit P10=P1^0;
/*---------------T0 Initial Function---------*/
void T0_init()
{
    TMOD=0x00;
    TH0=0xec;          // (65536-5000)/256
    TL0=0x78;          // (65536-5000)%256
    ET0=1;             // Enable T0 interrupt   开放定时器 T0 中断
    EA=1;              // Enable CPU interrupt   开放 CPU 中断
    TR0=1;             // Start T0   启动定时器 T0
}
/*---------------T0 Initial Function ---------*/

void   main()
{
    T0_init();
    while(1)；
}
/*---------------T0 Interrupt Function---------*/
void   Timer0_ISR（void）interrupt 1
{
    P10=!P10;          //每 5ms 对 P1.0 取反一次
}
```

E.g. Input single pulse from External Interrupt 0 (INT0), light LEDs (Low level drive) in P1 port bit-by-bit every negative pulse.

例 利用外部中断 0（INT0）输入引脚输入单次脉冲，每来一个负脉冲，逐位点亮 P1 口的发光二极管（设低电平驱动）。

Solution: Use INT0 and adopt negative edge trigger way; Set the initial value of LED's drive to FEH because of low level drive.

The reference C51 program (int0.c) is shown below:

解：根据题意采用外部中断 0，选择下降沿触发方式；因 LED 灯的驱动信号是低电平有效，设 LED 灯驱动初始值为 FEH。

C51 参考程序（int0.c）如下：

```
#include <REG51.H>   // include 8051 register head file   包含 8051 单片机的寄存器定义文件
unsigned   char   i = 0xfe ;
/*---------------- External Interrupt 0 Function--------------------*/
void   int0_isr( ) interrupt 0
{
     Pl = i;
     i <<= 1 ;
     if(i==0x80)   i = 0xfe;          /*After 6 times' shifting, "i" changes to "80", needs
                                      reassignment   移位 6 次后，i 将变为 80，需要重新赋值*/
}
/*----------------------- Main Function ----------------------*/
void   main(void)
{
     IT0 = l;                         //Set negative edge trigger   设置下降沿触发方式
     EX0 = 1 ;                        // Enable INT0   开放外部中断 0
     EA = 1 ;
     while(1);                        //原地踏步，模拟主程序
}
```

Summary 本章小结

The concept of interrupt, which was issued in the 1950s, is an important technology in computer. It is concerned with both hardware and software. In the help of interrupt, the task in computer became more flexible and more efficient. The physical base for the management schedule of modern computer operating system is the plentiful interrupt functions and perfect interrupt system. Conflict occurs when CPU face multitasking, while interrupt is a resource sharing technology essentially. The appearance of interrupt technology improves the development and application of computer greatly. Therefore, the competence of interrupt function is the key indicator which measures whether the computer is faultless or not.

A full interrupt process consists of 4 steps: interrupt request, interrupt response, interrupt service and return from interrupt (RETI).

There are ten interrupt request sources and two priority levels in STC15W201S MCU, which can implement two-level nested interrupt service. The interrupt enabling registers IE, IE2,

INT_CLKO and CMPCR1 residing in internal SFRs are used for controlling CPU whether to response interrupt request or not. The interrupt priority register IP is used for arranging priorities of all interrupt sources. The response order of interrupts in the same priority level is determined by the internal query logic.

Exercise 练习题

1. Completions

1) The service mode from CPU to I/O including_____, _____ and DMA.

2) An interruption process includes interrupt request, _____, _____and RETI.

3) In interruption mode, CPU and I/O device are working_____.

4) According to the response of CPU, interrupt request is divided into_____ and_____. All the interrupts in STC15W201S belong to_____.

5) When requires T0 interrupt, besides ET0 set to 1, _____ should be set to 1.

6) The interrupt priority of STC15W201S is divided into_____ levels. When the interrupt are at the same priority level, the first 5 interrupt are_____, T0 interrupt, _____, _____ and serial port interrupt from high to low.

7) The interrupt request signal input pin of external interrupt 0 is_____. The interrupt request signal input pin of external interrupt 1 is_____. The trigger modes of external interrupt 0 and external interrupt 1 are_____ and _____. The trigger mode of external interrupt 0 is_____ when IT0=1.

8) The keyword of interrupt function definition is_____.

9) The interrupt vector and number of external interrupt 0 are_____ and _____ separately.

10) The interrupt vector and number of external interrupt 1 are_____ and _____ separately.

11) The interrupt vector and number of T0 are_____ and _____ separately.

12) The interrupt vector and number of T2 are_____ and _____ separately.

13) The interrupt vector and number of T0 are_____ and _____ separately.

14) The interrupt vector and number of serial port interrupt are_____ and _____ separately.

15) The timer/counter of STC15W201S is working at_____ mode and initial value is set to_____ extends the external interrupt source, and the counting input pin of the timer/counter is the external interrupt request signal input pin.

2. Choice Questions

1) After the execution of “EA=1;EX0=1;EX1=1;ES=1;”, which one below is correct? _____

A. External interrupt 0, external interrupt 1 and serial port interrupt are enable
B. External interrupt 0, T0 and serial port interrupt are enable
C. External interrupt 0, T2 and serial port interrupt are enable
D. T0, T2 and serial port interrupt are enable

2) After the execution of "PS=1;PT0=1;", according to the interrupt priority level, which one below is correct? _____

A. External interrupt 0 → T0 → External interrupt 1→ serial port interrupt
B. External interrupt 0 → External interrupt 1 → T0 → serial port interrupt
C. T0→ serial port interrupt → External interrupt 0 → External interrupt 1
D. T0→ serial port interrupt → External interrupt 1 → External interrupt 0

3) After the execution of "PS=1;PT1=0;", which one below is correct? _____

A. External interrupt 1 can discontinue the processing of external interrupt 0
B. External interrupt 0 can discontinue the processing of external interrupt 1
C. External interrupt 1 can discontinue the processing of serial port interrupt
D. Serial port interrupt 1 can discontinue the processing of external interrupt 1

4) To enable T0 interrupt and set high level, which processing below is correct? _____

A. ET0=1;EA=1;PT0=1;
B. ET0=1;IT0=1;PT0=1;
C. ET0=1;EA=1;IT0=1;
D. IT0=1;EA=1;PT0=1;

5) When IT0=1, the trigger mode of external interrupt 0 is _____.

A. Active high
B. Active low
C. Negative Edge
D. Both positive edge and negative Edge

6) When IT1=1, the trigger mode of external interrupt 1 is _____.

A. Active high
B. Active low
C. Negative Edge
D. Both positive edge and negative Edge

3. True or False

1) Once there is interrupt request from interrupt source, the CPU must respond to the interrupt request, in STC15W201S. ( )

2) When a certain interrupt request enable bit is 1 and CPU interrupt enable bit (EA) is 1, CPU will respond to its interrupt request. ( )

3) When a certain interrupt source is interrupt enable, CPU will respond once there is an interrupt request from such source. ( )

4) The first step CPU response to interrupt request is to save the breakpoint address, and then turn to the interrupt vector corresponding to the interrupt source and executes automatically. ( )

5) The IRQ of external interrupt 0 is 1. ( )

6) The IRQ of T2 interrupt is 3. ( )

7) External interrupt 0 can discontinue the processing of serial port interrupt when they are in the same interrupt priority level. ( )

8) High level interrupt can discontinue the processing of low level interrupt. ( )

9) Parameters can be transported in interrupt function. ( )

10) Interrupt function can return data of any type. ( )

11) Using is keyword in interrupt function definition. ( )

12) Main function can call interrupt function actively. ( )

4. Question and Answer

1) What are the factors that can influence the response time of CPU to interrupt request?

2) What's the advantage of interrupt service over query service?

3) What kind of function should be included in an interrupt system?

4) What's the breakpoint address?

5) Are there interrupt priority level in STC15W201S MCU? According to natural priority, what're the first 5 interrupt from high to low?

Unit 8　Serial Port Communication of STC15W201S

第 8 单元　STC15W201S 单片机的串行口通信

8.1　Serial Port　串行口

STC15W201S MCU has integrated a programmable full duplex serial port data communication port with full function of Universal Asynchronous Receiver/Transmitter (UART). The serial port consists of two data buffer, a shift register, a serial controller and a baud rate generator. The data buffer is made up of two individual transmit buffer and receive buffer, and it can transmit and receive data simultaneously. Since the data receive buffer is write-only and the data transmit buffer is read-only, these two buffers can share the same address code, which is 99H. These two data buffers are named Serial Communication Control Register SBUF (Table 8.1). The operand of write operation to SBUF, such as x=SBUF, is the data receive buffer of serial port. While, the operand of read operation to SBUF, such as SBUF=x, is the data transmit buffer of serial port.

The default pins of transmit and receive are TxD/P3.1 and RxD/P3.0, which can switch to

STC15W201S 单片机内部有 1 个可编程全双工串行通信接口，它具有 UART 的全部功能。此串行口由两个数据缓冲器、一个移位寄存器、一个串行控制器和一个波特率发生器等组成。每个数据缓冲器由两个相互独立的接收、发送缓冲器构成，可以同时发送和接收数据。发送数据缓冲器只能写入而不能读出，接收缓冲器只能读出而不能写入，因而两个缓冲器可以共享一个地址码（99H）。串行口的两个数据缓冲器的共享地址码是 99H，串行口的两个数据缓冲器统称串行通信控制寄存器 SBUF（见表 8.1）。当对 SBUF 进行读操作（如：x=SBUF;）时，操作对象是串行口的接收数据缓冲器；当对 SBUF 进行写操作(如：SBUF = x;）时，操作对象是串行口的发送数据缓冲器。

STC15W201S 单片机串行口默认对应的发送、接收引脚是：TxD/P3.1、

P3.7 and P3.6 by setting the S1_S0 control bit in AUXR1(P_SW1).

RxD/P3.0，通过设置 AUXR(P_SW1)的 S1_S0 控制位，串行口的 TxD、RxD 硬件引脚可切换为 P3.7、P3.6。

8.1.1 Control Register of Serial Port 串行口控制寄存器

The SFRs related to serial port are serial port control register, Timer/Counter 2 registers related to baud rate setting and registers related to interrupt control, as shown in Table 8.1.

与单片机串行口有关的特殊功能寄存器有：单片机串行口的控制寄存器、与波特率设置有关的定时/计数器 T2 的相关寄存器、与中断控制相关的寄存器，详见表 8.1。

Table 8.1 SFRs related to Serial Port 1

与单片机串行口 1 有关的特殊功能寄存器

	Address	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value
SCON	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	0000 0000
SBUF	99H	Serial Port Data Buffer								xxxx xxxx
PCON	87H	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL	0011 0000
AUXR	8EH	T0x12	-	UART_M0x6	T2R	T2_C/$\overline{T}$	T2x12	-	S1ST2	0x00 00x0
T2L	D7H	Lower 8 bits of T2								0000 0000
T2H	D6	Upper 8 bits of T2								0000 0000
IE	A8H	EA	ELVD	-	ES	-	EX1	ET0	EX0	00x0 x000
IP	B8H	-	PLVD	-	PS	-	PX1	PT0	PX0	x0x0 x000
P_SW1 (AUXR1)	A2H	-	S1_S0	-	-	-	-	-	-	x0xx xxxx

1. Serial Port Control Register SCON

SCON is used for setting the mode, receiving method and state flag of serial port. SCON is bit-addressable and its byte address is 98H. All bits in SCON should be “0” after reset, which is shown in Table 8.2.

1. 串行口控制寄存器 SCON

串行口控制寄存器 SCON 用于设定串行口的工作方式、接收控制以及设置状态标志。字节地址为 98H，可进行位寻址，单片机复位时，所有位全为 0，其格式如表 8.2 所示。

Table 8.2 SCON

SCON	Address 地址	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value 复位值
	98H	SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI	0000 0000

The definitions of all bits are as below:

SM0/PE, SM1:

① When the PCON in register SMOD0 is "1", SM0/PE is used for frame error detecting. It will be set through UART receiver while an invalid stop bit is detected. Cleared by software only.

When the PCON in register SMOD0 is "0", SM0/PE is used for setting the serial port working mode in company with SM1, as shown in Table 8.3 (f_{SYS} is the frequency of system clock).

对各位的说明如下：

SM0/PE、SM1：

① PCON寄存器中的SMOD0位为1时，SM0/PE用于帧错误检测，当检测到一个无效停止位时，通过UART接收器设置该位，它必须由软件清零。

PCON寄存器中的SMOD0为0时，SM0/PE和SM1一起指定串行通信的工作方式，如表8.3所示（其中，f_{SYS}为系统时钟频率）。

Table 8.3　Configuration of Serial Port Mode

串行口方式选择位

SM0 SM1	Mode	Function	Baud Rate
0　0	Mode 0	8-bit synchronous shift register	f_{SYS}/12 or f_{SYS}/2
0　1	Mode 1	10-bit UART	Adjustable, 1/4(T2's overflow rate)
1　0	Mode 2	11-bit UART	f_{SYS}/64 or f_{SYS}/32
1　1	Mode 3	11-bit UART	Adjustable, 1/4(T2's overflow rate)

② SM2: Control bit of multiprocessor communication, applied in Mode 2 and Mode 3. If (SM2)=1, RI will not be set unless the 9th bit of received data bit RB8 is "1"; if (SM2)=1 and RB8=1, set RI flag. In the receiving mode of Mode 2 and Mode 3, RI will be set normally regardless of the value of RB8.

③ REN: Control bit of serial reception enabling. Cleared or set by software.

0: disable reception;

1: enable reception.

④ TB8: The 9th bit of transmit data. In Mode 2 and Mode 3, it's cleared or set by software, and can be used as parity check digit and flag which can distinguish address frame and data frame in multiprocessor communication. Generally, TB8 is "1" when there is address frame, and TB8 is "0" when there is data frame.

⑤ RB8: The 9th bit of receive data in Mode 2 and Mode 3. It can be used as parity check digit

② SM2：多机通信控制位，用于方式2和方式3中。若SM2=1，且接收到的第9位数据RB8为0时，不激活RI；若SM2=1，且RB8=1时，则置位RI标志。在方式2和方式3的接收状态下，不论接收到第9位RB8为0还是为1，RI都以正常方式被激活。

③ REN：允许串行接收控制位。由软件置位或清零。REN=1时，启动接收；REN=0时，禁止接收。

④ TB8：串行发送数据的第9位。在方式2和方式3中，由软件置位或复位，可作为奇偶校验位。在多机通信中，可作为区别地址帧或数据帧的标识位，一般约定地址帧时TB8为1，数据帧时TB8为0。

⑤ RB8：在方式2和方式3中，是串行接收到的第9位数据，作为

and flag which can distinguish address frame and data frame.

⑥ TI: Transmit interrupt flag. Set by hardware when 8-bit of data has been transmitted in Mode 0. While in other modes, set by hardware at the beginning of the STOP bit. This bit indicates that a frame is transmitted. It can be responded by query or interrupt and must be cleared manually by software in the corresponding Query Service Routine or ISR.

⑦ RI: Receive interrupt flag. Set by hardware when 8-bit of data has been received in Mode 0. While in other modes, set by hardware at the arrival of the STOP bit. This bit indicates that a frame is received. It can be responded by query or interrupt and must be cleared manually by software in the corresponding Query Service Routine or ISR.

2. Power and Baud Rate Control Register PCON

PCON is a specific register used for power control. It is non bit-addressable, and its byte address is 87H. Its reset value is 30H. The SMOD and SMOD0 bit in PCON are related to serial port control. The format of PCON is shown in Table 8.4.

奇偶校验位或地址帧、数据帧的标识位。

⑥ TI：发送中断标志位。在方式 0 中，发送完 8 位数据后，由硬件置位；在其他方式中，在发送停止位之初由硬件置位。TI 是发送完一帧数据的标志，既可以用查询的方法也可以用中断的方法来响应该标志，然后在相应的查询服务程序或中断服务程序中，由软件清零 TI。

⑦ RI：接收中断标志位。在方式 0 中，接收完 8 位数据后，由硬件置位；在其他方式中，在接收停止位时由硬件置位。RI 是接收完一帧数据的标志，同 TI 一样，既可以用查询的方法也可以用中断的方法来响应该标志，然后在相应的查询服务程序或中断服务程序中，由软件清零 RI。

2. 电源及波特率选择寄存器 PCON

PCON 是为单片机的电源控制而设置的专用寄存器，不可以位寻址，字节地址为 87H，复位值为 30H。其中 SMOD、SMOD0 与串口控制有关，其格式与说明如表 8.4 所示。

Table 8.4 PCON

PCON	Address	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value
	87H	SMOD	SMOD0	LVDF	POF	GF1	GF0	PD	IDL	0011 0000

SMOD: Double baud rate control bit. In Mode 1, Mode 2 and Mode 3, the baud rate of serial communication is related to SMOD.

0: Disable double baud rate of the UART;

1: Enable double baud rate of the UART.

SMOD0: Frame Error (FE) detection control bit.

SMOD：SMOD 为波特率倍增系数选择位。在方式 1、2 和 3 时，串行通信的波特率与 SMOD 有关。当 SMOD=0 时，通信速度为基本波特率；当 SMOD=1 时，通信速度为基本波特率乘 2。

SMOD0：帧错误检测有效控制位。SMOD0＝1，SCON 寄存器中的

0: SM0/FE in SCON is SM0 function, setting the serial port working mode in the company of SM1; 1: SM0/FE in SCON is FE detection function.

SM0/FE 用于帧错误检测（FE）；SMOD0 = 0，SCON 寄存器中的 SM0/FE 用于 SM0 功能，与 SM1 一起指定串行口的工作方式。

3．Auxiliary Register AUXR

The format of AUXR is shown as Table 8.5, where UART_M0x6, T2x12, T2R, T2_C/$\overline{T}$ and S1ST2 are related to baud rate of the serial port.

3．辅助寄存器 AUXR

辅助寄存器 AUXR 的格式如表 8.5 所示，其中 UART_M0x6、T2x12、T2R、T2_C/$\overline{T}$、S1ST2 与串行口的波特率有关。

Table 8.5　AUXR

AUXR	Address	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value
	8EH	T0x12	-	UART_M0x6	T2R	T2_C/$\overline{T}$	T2x12	-	S1ST2	0x00 00x1

UART_M0x6: Baud rate select bit of UART in Mode 0.

0: The baud rate of UART is $f_{SYS}/12$, the same as traditional 8051.

1: The baud rate of UART is $f_{SYS}/2$, six times the speed of traditional 8051.

UART_M0x6：串行口方式 0 通信速度设置位。UART_M0x6＝0，串行口方式 0 的通信速度与传统 8051 单片机一致，波特率为系统时钟频率的 12 分频，即 $f_{SYS}/12$；UART_M0x6＝1，串行口方式 0 的通信速度是传统 8051 单片机通信速度的 6 倍，波特率为系统时钟频率的 2 分频，即 $f_{SYS}/2$。

T2x12, T2R, T2_C/$\overline{T}$: These bits are used for setting and startup of Timer/Counter 2, when Timer/Counter 2 serves as baud-rate generator.

T2x12、T2R、T2_C/$\overline{T}$ ：定时/计数器 T2 作为波特率发生器时，T2x12、T2R、T2_C/$\overline{T}$ 用于设置定时/计数器 T2 以及启动。

8.1.2　Working Mode of Serial Port　串行口的工作方式

The serial communication of STC15W201S MCU can work in four different modes which are configured by setting SM0 and SM1 in SFR SCON (when SMOD0 = 0).

STC15W201S 单片机串行通信有 4 种工作方式，当 SMOD0=0 时，通过设置 SCON 中的 SM0、SM1 位来选择。

1. Mode 0

In Mode 0, the UART is used as shift register, whose baud-rate is $f_{SYS}/12$ (UART_M0x6 = 0) or $f_{SYS}/2$ (UART_M0x6 = 1). Serial data enters and exits through RxD (P3.0).TxD(P3.1) outputs the synchronous shift clock. This mode is usually used in extended I/O port.

1) Transmit

When TI=0, after the data is written into SBUF, 8-bit data will be transmitted with the Least-significant Bit (LSB) first by serial port at the baud-rate $f_{SYS}/12$ or $f_{SYS}/2$ on RxD. Set the interrupt flag TI and send an interrupt request to CPU after transmission finished. Cleared TI by software before transmit again. The timing sequence of transmission in Mode 0 is shown in Figure 8.1.

1. 方式 0

在方式 0 下，串行口作为同步移位寄存器使用，其波特率为 $f_{SYS}/12$（UART_M0x6 为 0 时）或 $f_{SYS}/2$（UART_M0x6 为 1 时）。串行数据从 RxD（P3.0）端输入或输出，同步移位脉冲由 TxD(P3.1)送出。这种方式常用于扩展 I/O 口。

1）发送

当 TI=0，有数据写入串行口发送缓冲器 SBUF 时，串行口将 8 位数据以 $f_{SYS}/12$ 或 $f_{SYS}/2$ 的波特率从 RxD 引脚输出（低位在前），发送完毕置位中断标志 TI，并向 CPU 请求中断。在再次发送数据之前，必须由软件清零 TI 标志。方式 0 发送时序如图 8.1 所示。

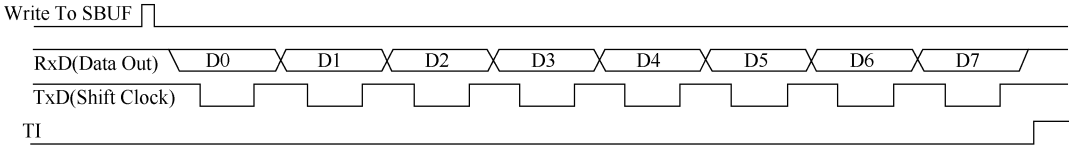


Figure 8.1 Transmission Timing Sequence of Mode 0

方式 0 发送时序

In the transmitting of Mode 0, serial parallel output shift register, such as 74LS164, CD4094 and other chips of this kind, can be attached to serial port, which is used for extending parallel output port. The logic circuit is shown in Figure 8.2.

方式 0 发送时，串行口可以外接串行输入并行输出的移位寄存器，如 74LS164、CD4094 等芯片，用来扩展并行输出口，其逻辑电路如图 8.2 所示。

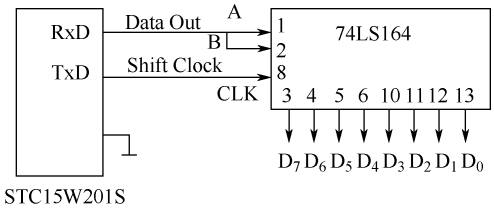


Figure 8.2 Extending I/O Output of Mode 0 方式 O 用于扩展 I/O 口输出

2) Receive

When RI=0, set REN, and then input data with the Least-significant Bit (LSB) first at the baud-rate $f_{SYS}/12$ or $f_{SYS}/2$ on RxD. Set the interrupt flag RI and send an interrupt request to CPU after the 8-bit data reception finish. Clear RI by software before transmitting it again. The timing sequence of reception in Mode 0 is shown in Figure 8.3.

2）接收

当 RI=0 时，置位 REN，串行口即开始从 RxD 端以 $f_{SYS}/12$ 或 $f_{SYS}/2$ 的波特率输入数据（低位在前），当接收完 8 位数据后，置位中断标志 RI，并向 CPU 请求中断。在再次接收数据之前，必须由软件清零 RI 标志。方式 0 接收时序如图 8.3 所示。

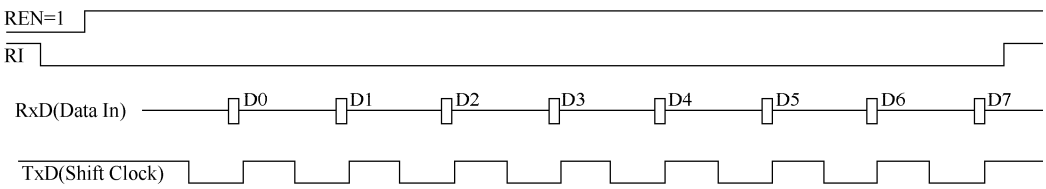


Figure 8.3　Reception Timing Sequence of Mode 0

方式 0 接收时序

In the receiving of Mode 0, serial parallel output shift register, such as 74LS165 and other chips of this kind, can be attached to serial port, which is used for extending parallel input port. The logic circuit is shown in Figure 8.4.

方式 0 接收时，串行口可以外接并行输入串行输出的移位寄存器，如 74LS165 芯片，用来扩展并行输入口，其逻辑电路如图 8.4 所示。

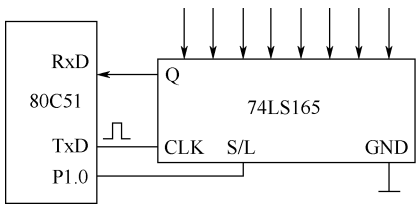


Figure 8.4　Extending I/O Input of Mode 0

方式 0 用于扩展 I/O 口输入

TB8 and RB8 in SCON are unavailable in Mode 0. It should be noted that TI or RI will be set by hardware automatically after every 8 bits of data is transmitted or received and must be cleared by software manually after CPU's response.

串行控制寄存器 SCON 中的 TB8 和 RB8 在方式 0 中未用。值得注意的是，每当发送或接收完 8 位数据后，硬件会自动置 TI 或 RI 为 1，CPU 响应 TI 或 RI 中断后，必须由用户用软件清零。

NOTE：SM2 must be set to 0 in Mode 0.
注：方式 0 时，SM2 必须为 0。

2. Mode 1

When working in Mode 1, the serial port is some kind of variable baud-rate 10-bit UART with a data frame including 1 start bit (0), 8 bit data and 1 bit stop bit (1). The frame format is shown in Figure 8.5.

2. 方式 1

串行口工作在方式 1 时，串行口为波特率可调的 10 位通用异步接口 UART，数据帧包括 1 位起始位（0），8 位数据位和 1 位停止位（1），其帧格式如图 8.5 所示。

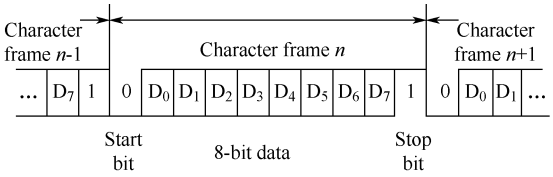


Figure 8.5 Format of 10-bit Frame
10 位帧的格式

1) Transmit

When TI=0, serial port start transmission once the data has been written into SBUF. With the synchronization of transmitting shift clock, the TxD sends the start bit first, and then 8-bit data, finally stop bit. Interrupt flag TI will be set when the all 10-bit data is sent. The transmitting timing sequence of Mode 1 is shown in Figure 8.6. The baud-rate of transmitting depends on the overflow rate of Timer 2.

1）发送

当 TI=0 时，数据写入发送缓冲器 SBUF 后，就启动了串行口发送过程。在发送移位时钟的同步下，从 TxD 引脚先送出起始位，然后是 8 位数据位，最后是停止位。一帧 10 位数据发送完后，中断标志 TI 置 1。方式 1 的发送时序如图 8.6 所示。方式 1 数据传输的波特率取决于定时器 T2 的溢出率。

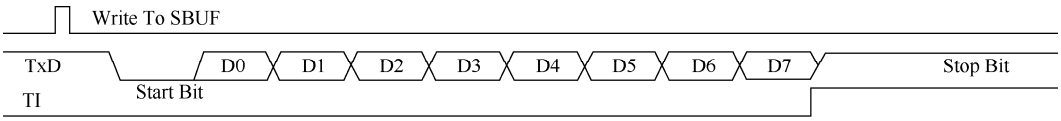


Figure 8.6 Transmission Timing Sequence of Mode 1
方式 1 发送时序

2) Receive

When RI=0, set REN to start reception of serial port. When a negative transition of input level on the pin RxD is detected, RxD is sampled

2）接收

当 RI＝0 时，置位 REN，启动串行口接收过程。当检测到 RxD 引脚输入电平发生负跳变时，接收器以所选择

at a rate of 16 times the established baud rate by receiver, where the 7th, 8th and 9th pulse are the sampling points. The start bit is valid when the sampled values that were seen in low level at least 2 times of sample. Receive the rest bits of this frame in the same way. The 8-bit data is put into SBUF in receiving process. Set RI once receives a stop bit, and sends an interrupt request to CPU. The timing sequence of reception in Mode 1 is shown in Figure 8.7.

波特率的 16 倍速率采样 RxD 引脚电平，以 16 个脉冲中的第 7、8、9 三个脉冲为采样点，取两个或两个以上相同值为采样电平，若检测电平为低电平，则说明起始位有效，并以同样的检测方法接收这一帧信息的其余位。接收过程中，8 位数据装入接收 SBUF，接收到停止位时，置位 RI，向 CPU 请求中断。方式 1 的接收时序如图 8.7 所示。

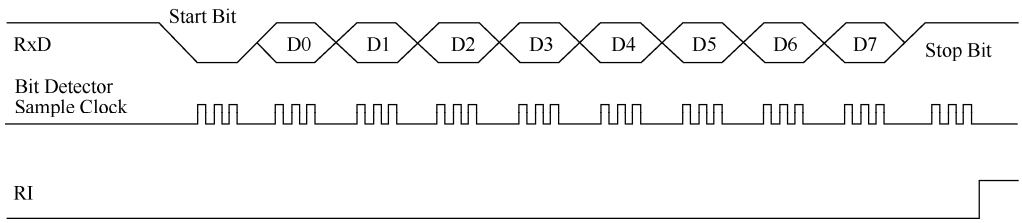


Figure 8.7　Reception Timing Sequence of Mode 1

方式 1 的接收时序

3. Mode 2

In Mode 2, serial port works in a kind of 11-bit UART. The data frame includes a start bit (0), 8 data bits, a programmable bit (such as parity checking) and a stop bit (1). The frame format is shown in Figure 8.8.

3. 方式 2

串行口工作在方式 2 时，串行口为 11 位 UART。数据帧包括 1 位起始位（0）、8 位数据位、1 位可编程位（如用于奇偶校验）和 1 位停止位（1），其帧格式如图 8.8 所示。

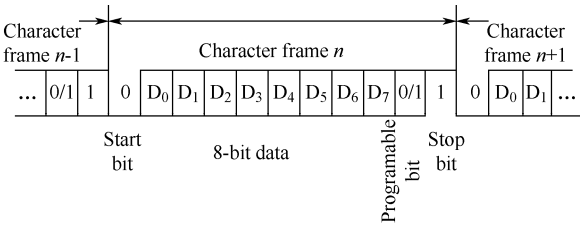


Figure 8.8　Format of 11-bit Frame

11 位帧的格式

1) Transmit

TB8 (programmable bit) is configured by software according to communication protocol

1）发送

发送前，先根据通信协议由软件设置好 TB8（即可编程位）。当 TI＝0 时，

before transmission. When TI=0, once the data going to be transmitted is put into SBUF by instruction, start the transmission of transmitter. With the synchronization of transmitting shift clock, TxD sends the start bit first, and then 8-bit data and TB8, finally stop bit. Interrupt flag TI will be set and interrupt request will be sent to CPU when the all 11-bit data is sent. TI must be cleared by ISR or query routine before the next frame's transmission. The transmitting timing sequence of Mode 2 is shown in Figure 8.9.

用指令将要发送的数据写入 SBUF，则启动发送器的发送过程。在发送移位时钟的同步下，从 TxD 引脚先送出起始位，依次是 8 位数据位和 TB8，最后是停止位。一帧 11 位数据发送完毕后，置位中断标志 TI，并向 CPU 发出中断请求。在发送下一帧信息之前，TI 必须由中断服务程序或查询程序清零。

方式 2 的发送时序如图 8.9 所示。

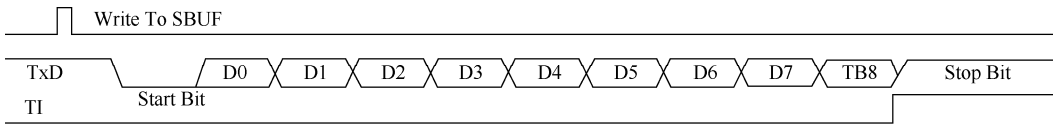


Figure 8.9 Transmission Timing Sequence of Mode 2

方式 2 的发送时序

2) Receive

When RI=0, set REN to start reception of serial port. When a negative transition of input level on the pin RxD is detected, RxD is sampled at a rate of 16 times the established baud rate by receiver, where the 7^{th}, 8^{th} and 9^{th} pulse are the sampling points. The start bit is valid when the sampled value that was seen in low level at least 2 times of sample. Receive the rest bits of this frame in the same way. In receiving process, 8-bit data will be put into SBUF, the 9^{th} bit will enter RB8. Set RI and sent interrupt request to CPU, when SM2=0 or SM2=1 and RB8=1 simultaneously at the arrival of stop bit; otherwise, will not set RI and data loss. The timing sequence of reception in Mode 2 is shown in Figure 8.10.

2）接收

当 RI＝0 时，置位 REN，启动串行口接收过程。当检测到 RxD 引脚输入电平发生负跳变时，接收器以所选择波特率的 16 倍速率采样 RxD 引脚电平，以 16 个脉冲中的第 7、8、9 三个脉冲为采样点，取两个或两个以上相同值为采样电平，若检测电平为低电平，则说明起始位有效，并以同样的检测方法接收这一帧信息的其余位。接收过程中，8 位数据装入接收 SBUF，第 9 位数据装入 RB8，接收到停止位时，若 SM2＝0 或 SM2＝1 且接收到的 RB8＝1，则置位 RI，向 CPU 请求中断；否则不置位 RI 标志，接收数据丢失。方式 2 的接收时序如图 8.10 所示。

4．Mode 3

In Mode 3, serial port works in a kind of 11-bit UART, the same way as Mode 2. The difference between Mode 3 and Mode 2 lies in the configuration of baud-rate, where the baud-rate of Mode 2 is $f_{SYS}/64$ (SMOD=0) or $f_{SYS}/32$ (SMOD=1), while the baud-rate of Mode 3 depends on the overflow rate of T2 and the bit SMOD of SFR PCON in the same way as Mode 1.

4．方式 3

工作在方式 3 时，串行口同方式 2 一样为 11 位 UART。方式 2 与方式 3 的区别在于波特率的设置方法不同，方式 2 的波特率为 $f_{SYS}/64$（SMOD 为 0）或 $f_{SYS}/32$（SMOD 为 1）；方式 3 数据传输的波特率同方式 1 一样取决于定时器 T2 的溢出率和 SMOD 控制位。

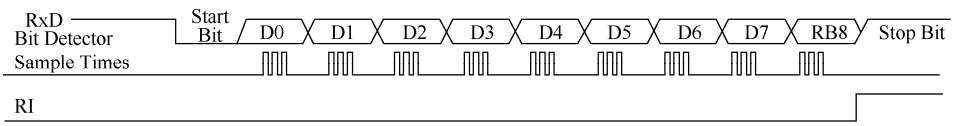


Figure 8.10　Reception Timing Sequence of Mode 2

方式 2 的接收时序

The transmission and reception process of Mode 3 are the same as Mode 2 except the rate of transmission and reception. In the receiving process, only when both SM2=0 or SM2=1 and the received RB8=1, set RI and ask CPU for an interrupt to receive data; otherwise, will not set RI and data loss. Thus, Mode 2 and Mode 3 often apply to multiprocessor communication.

方式 3 除发送过程与接收过程速率不同以外，其他过程和方式 2 完全一致。在方式 2 和方式 3 下的接收过程中，只有当 SM2＝0 或 SM2＝1 且接收到的 RB8＝1 时，才会置位 RI，向 CPU 请求中断接收数据；否则不会置位 RI 标志，接收数据丢失，因此，方式 2 和方式 3 常用于多机通信中。

8.1.3　Baud Rate of Serial Port　串行口的波特率

In serial communication, a certain convention of data transfer rate (baud rate) should be followed between transmitter and receiver to carry out normal communication. There are four modes for MCU communication, where baud-rate of Mode 0 and Mode 2 are fixed, baud-rate of Mode 1 and Mode 3 are variable and depend on the overflow rate of T2.

在串行通信中，收发双方对传送数据的速率（即波特率）要有一定的约定，才能进行正常的通信。单片机的串行通信有 4 种工作方式。其中方式 0 和方式 2 的波特率是固定的；方式 1 和方式 3 的波特率可变，由定时器 T2 的溢出率决定。

1)　Mode 0 and Mode 2

Mode 0, baud-rate is $f_{SYS}/12$ (when

1）方式 0 和方式 2

在方式 0 中，波特率为 $f_{SYS}/12$

UART_M0x6=0) or $f_{SYS}/2$ (when UART_M0x6=1).

Mode 2, the baud-rate depends on the value of bit SMOD in PCON. If SMOD=0, baud-rate is $f_{SYS}/64$, if SMOD=1, baud-rate is $f_{SYS}/32$.

That is:

（UART_M0x6 为 0 时）或 $f_{SYS}/2$（UART_M0x6 为 1 时）。

在方式 2 中，波特率取决于 PCON 中的 SMOD 值，当 SMOD=0 时，波特率为 $f_{SYS}/64$；当 SMOD=1 时，波特率为 $f_{SYS}/32$。

即：

$$\text{Baud-rate} = \frac{2^{\text{SMOD}}}{64} \cdot f_{\text{SYS}}$$

2) Mode 1 and Mode 3

The baud-rate of Mode 1 and Mode 3 are determined by T2 overflow rate.

2）方式 1 和方式 3

在方式 1 和方式 3 下，波特率由定时器 T2 的溢出率决定。

$$\text{Baud-rate} = (\text{T2 overflow rate})/4$$

The T2 overflow rate is the reciprocal of the T2 timing and determined by the counting rate and preset value of T2. The counting rate is concerned with bit T2_C/$\overline{T}$ and bit T2x12 of register AUXR. If T2_C/$\overline{T}$ =0, the counting rate is $f_{SYS}/12$ (T2x12=0) or f_{SYS} (T2x12=1); if T2_C/$\overline{T}$ =1, the counting rate is clock frequency from outside.

其中 T2 溢出率为 T2 定时时间的倒数，取决于单片机定时器 T2 的计数速率和定时器的预置值。计数速率与 AUXR 寄存器中的 T2_C/$\overline{T}$ 和 T2x12 位有关，当 T2_C/$\overline{T}$ =0 时，计数速率为 $f_{SYS}/12$（T2x12＝0 时）或 f_{SYS}（T2x12＝1 时）；当 T2_C/$\overline{T}$ =1 时，计数速率为外部输入时钟频率。

Tools such as software delay calculator and baud-rate calculator are included in STC-ISP on-line programming software (V6.58). Enter the baud-rate the serial port needs and MCU clock frequency (such as 11.0592MHz) and set the MCU device cycle mode as 1T, and then after the parameter configuration finished, click “Generate C program code” button, the STC-ISP on-line programming software will generate C program function automatically, whose replication can be used directly, shown as Figure 8.11.

STC-ISP 在线编程软件（V6.58）中包含了软件延时计算器、波特率计算器等使用工具，在波特率计算工具界面中只要输入串口所需的波特率以及单片机采用的时钟频率（如：11.0592MHz）、单片机机器周期模式为 1T 等，完成参数设置后再单击“生成 C 代码”按钮，STC-ISP 在线编程软件就可自动生成 C 语言函数，可直接复制使用，如图 8.11 所示。

Figure 8.11　Parameter Configuration and Baud-rate Generation Subroutine of Baud-rate Calculator (C Program)

波特率计算器参数设置与生成的波特率子程序（C 语言格式）

8.1.4　Applications of Serial Port Application　串行口的应用举例

1. Program and Application of Mode 0

Mode 0 of serial port is a kind of synchronous shift register. Parallel I/O port can be extended in Mode 0. For example, in keyboard and display interface, serial in and parallel out shift register (such as 74LS164) is used for extension, which can extend an 8-bit parallel output port each, and it can be used for connecting a LED display as static display or 8 row lines in keyboard.

E.g. Apply two pieces of 74HC595 to extend a 16-bit parallel port, and connect to sixteen pieces of LEDs. The circuit is shown as Figure 8.12. Take advantage of its serial in parallel out function, light up the LEDs from left to right and circulate (16-bit pipeline LED).

Here gives the solution: 74HC595 is an 8-bit serial in parallel out shift register nearly the same as 74LS164. The drive current of 74LS164 (25mA) is less than 74HC595 (35mA). The main advantage of 74HC595 is data storing register, which will maintain the data in output port in the process of shifting. It's very useful in the slow serial scene,

1．方式 0 的编程和应用

串行口方式 0 是同步移位寄存器方式。应用方式 0 可以扩展并行 I/O 口，比如在键盘、显示器接口中，外扩串行输入并行输出的移位寄存器（如 74LS164），每扩展一片移位寄存器可扩展一个 8 位并行输出口。可以用来连接一个 LED 显示器作为静态显示或键盘中的 8 根列线使用。

例　使用 2 块 74HC595 芯片扩展 16 位并行口，外接 16 只发光二极管，电路连接图如图 8.12 所示。利用它的串入并出功能，把发光二极管从右向左依次点亮，并不断循环之（16 位流水灯）。

解：74HC595 和 74LS164 功能相仿，都是 8 位串行输入转并行输出移位寄存器。74LS164 的驱动电流(25mA)比 74HC595 (35mA)的要小。74HC595 的主要优点是具有数据存储寄存器，在移位的过程中，输出端的数据可以保持不变。这在串行速度

where there won't be flicker in LED. Besides, 74HC595 has a function of cascade, which can extend more output port.

慢的场合很有用处，数码管没有闪烁感。而且 74HC595 具有级联功能，通过级联能扩展更多的输出口。

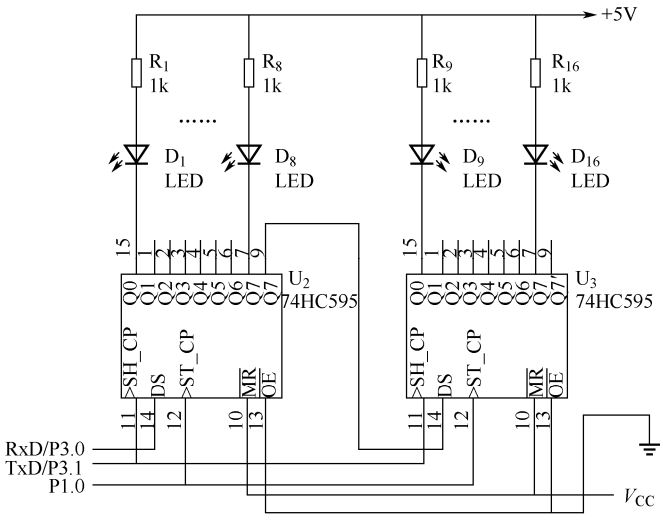


Figure 8.12 Output port extension in serial port Mode 0

串口方式 0 扩展输出口

Q0~Q7 are the parallel outputs ports, that is, the data outputs ports of storing register. DS is a serial data input port, Q7' is a serial output port used for connecting the serial data input port DS of the cascade chip. ST_CP is the clock input of the storing register (lock at low level), SH_CP is the clock input of shift register (shift at positive edge), $\overline{OE}$ is the enable of tri-state output, $\overline{MR}$ is the reset of the chip (active low, reset shift register at low level).

Q0～Q7 是并行数据输出口，即存储寄存器的数据输出口，DS 是串行数据输入端，Q7'是串行输出口用于连接级联芯片的串行数据输入端 DS，ST_CP 是存储寄存器的时钟脉冲输入端（低电平锁存），SH_CP 是移位寄存器的时钟脉冲输入端（上升沿移位），$\overline{OE}$ 是三态输出使能端，$\overline{MR}$ 是芯片复位端（低电平有效，低电平时移位寄存器复位）。

The reference C program is shown as below:

C 语言参考程序如下：

```
#include<reg51.h>
sbit ST_CP = P1^0;
unsigned char table1[8]={0xfe,0xfd,0xfb,0xf7,0xef,0xdf,0xbf,0x7f};
unsigned char table2[8]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};
void delayms(int ms)
{
    int i,j;
    for(i=ms;i>0;i--)
        for(j=110;j>0;j--);
```

```
}

void main()
{
    unsigned char i;
    SCON=0x00;
    while(1)
    {
        for(i=0;i<8;i++)
        {
            SBUF=table1[i];
            while(!TI);
            TI=0;
            SBUF=table2[i];
            while(!TI);
            TI=0;
            ST_CP = 1;          // Input lock pulse    输入锁存脉冲
            delayms(200);
            ST_CP = 0;
            delayms(200);
        }
    }
}
```

2. Dual-processor Communication

The dual-processor communication is used for information exchange between MCUs. There are two methods for dual-processor asynchronous communication program: query and interrupt. But in most applications, interrupt is used for receiving data in receiver in dual-processor communication in order to improve the CPU's efficiency, while query is used for transmitting in transmitter.

The two MCUs in dual-processor communication can connect each other directly, which is shown in Figure 8.13. The TxD, RxD and GND of MCU A connect to the RxD, TxD and GND of MCU B respectively. But, the distance of communication couldn't be more than 5m because the communication

2. 双机通信

双机通信用于单片机之间交换信息。对于双机异步通信的程序通常采用两种方法：查询方式和中断方式。但在很多应用中，双机通信的接收方都采用中断的方式来接收数据，以提高 CPU 的工作效率；发送方仍然采用查询方式发送。

双机通信的两个单片机的硬件连接可直接连接，如图 8.13 所示，A 机的 TxD 接 B 机的 RxD，A 机的 RxD 接 B 机的 TxD，A 机的 GND 接 B 机的 GND。但单片机的通信采用 TTL 电平传输信

adopts the TTL level transmission. Actually, RS-232C standard level is used for point to point communication connection in practice, shown as Figure 8.14, where MA232 is a level conversion chip. RS-232C standard level is the PC serial communication standard which will be discussed in detail in next section.

息，其传输距离一般不超过 5m，所以实际应用中通常采用 RS-232C 标准电平进行点对点的通信连接，如图 8.14 所示，MA232 是电平转换芯片。RS-232C 标准电平是 PC 机串行通信标准，详细内容见下节。

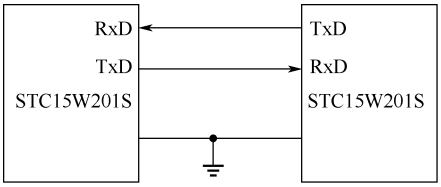


Figure 8.13 Dual-processor Asynchronous Communication Interface Circuit

双机异步通信接口电路

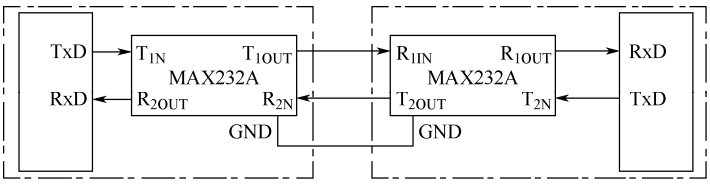


Figure 8.14 Point to Point Communication Interface Circuit

点对点通信接口电路

E.g. A program for communication between MCU A and MCU B with requirements as follow: P3.6 and P3.7 of MCU A connect to switch K1 and K2 respectively; input high level when switch is off, input low level when switch is on; K1 and K2 are used for inputting control command and control the output of the MCU B's P1 port through serial communication. P1.5~P1.0 of MCU B connect a LED individually, low level drive; all LEDs are off at initiation; the LEDs are controlled by P1's output according to the switch signals of K1 and K2 after serial reception. The control relationship is shown in Table 8.6.

例　编制程序，使 A、B 双方单片机能够进行通信。要求：A 机的 P3.6、P3.7 分别接开关 K1、K2，开关断开时输入高电平，开关闭合时输入低电平，K1、K2 用于输入控制命令，通过串口通信控制 B 机的 P1 口的输出状态；B 机的 P1.5～P1.0 各接 1 只 LED 灯电路，低电平驱动，初始时，LED 灯全灭，串行接收后，根据接收到 K1、K2 开关信号向 P1 输出信号控制 LED 灯的工作状态，控制关系如表 8.6 所示。

Table 8.6　Communication Control Relationship between MCU A and MCU B

A、B 机通信控制关系表

MCU A		MCU B
K2(P3.7)	K1(P3.6)	LED5～LED0（P1.5～P1.0）
1	1	LED5、LED3、LED1 on，LED4、LED2、LED0 off
1	0	LED5、LED3、LED1 off，LED4、LED2、LED0 on
0	1	LED5、LED4、LED3 on，LED2、LED1、LED0 off
0	0	All the six LEDs are on

Solution: Let the frequency of oscillator is 11.0592MHz, baud-rate of data transmission is 2400.

解：设晶振频率为 11.0592MHz，数据传输波特率为 2400。

1)　Transmitting program of MCU A (send.c)　　1）A 机发送程序（send.c）

```
#include <reg52.h>
#define uchar unsigned char
#define uint    unsigned int
/*-------------Delay Subroutine -------------*/
void delay(uchar i)
{
        uchar x,y,z;                /* Delay function, delay about 1ms when i=1
                                       延时函数，当 i 为 1 时约为 1ms*/
        for(z = i; z>0; z--)
            for(y = 11; y>0; y--)
                  for(x = 195; x>0; x--);
}
/*-------------2400bps@11.0592MHz Serial Port Initialization Function -------------*/
void UartInit(void)        //2400bps@11.0592MHz
{
        SCON = 0x40;                /* Mode 1, 8-bit data, variable baud-rate
                                       方式 1，8 位数据，可变波特率*/
        AUXR |= 0x04;               /* The clock of Time 2 is fosc, that is 1T
                                       定时器 2 时钟为 fosc，即 1T*/
        T2L = 0x80;                 // Set timing initiation    设定定时初值
        T2H = 0xFB;                 // Set timing initiation    设定定时初值
        AUXR |= 0x01;               /* Choose T2 as baud-rate generator for serial port 1
                                       串口 1 选择定时器 2 为波特率发生器*/
        AUXR |= 0x10;               // Enable T2    启动定时器 2
}
/*-------------Transmitting Subroutine -------------*/
void Send_Byte()                    /* A function for serial port sending a byte of data
                                       串口发送一个字节的程序*/
{
      uchar x;
      P3|=0xc0;
      x =P3;
```

```
    SBUF = x;                    // Enable serial port transmitting   启动串口发送
    while(TI==0);
    TI = 0;
}
/*-------------Main Function-------------*/
void main(void)
{
    UartInit();
    while(1)
    {
        Send_Byte();             /* Read switch state, send to MCU B in series
                                    读开关数据，并串行发送给 B 机*/
        delay();                 //设置发送间隔
    }
}
```

2) Receiving program of MCU B (receive.c) 2）B 机接收程序

```
#include <reg51.h>
#include <intrins.h>
#define uchar unsigned char
#define uint   unsigned int
uchar x;
/*-----------2400bps@11.0592MHz Serial Port Initialization Function ------------*/
void UartInit(void)        //2400bps@11.0592MHz
{
    SCON = 0x40;             /* Mode 1, 8-bit data, variable baud-rate
                                方式 1，8 位数据，可变波特率*/
    AUXR |= 0x04;            /* The clock of Time 2 is fSYS，that is 1/T
                                定时器 2 时钟为 fSYS，即 1/T*/
    T2L = 0x80;              // Set timing initiation   设定定时初值
    T2H = 0xFB;              // Set timing initiation   设定定时初值
    AUXR |= 0x01;            /* Choose T2 as baud-rate generator for serial port 1
                                串口 1 选择定时器 2 为波特率发生器*/
    AUXR |= 0x10;            // Enable T2   启动定时器 2
    ES=1;                    // Enable serial port interrupt   开放串行口中断
    EA=1;                    // Enable the total interrupt   开放总中断
    REN=1;                   // Enable receiving   启动接收
}
/*-------------Main Function-------------*/
void main(void)
{
    UartInit();              // Call serial port initialization function   调用串口初始化函数
    RI=0;
    while(1)
    {
        if(x&0xc0==0x00)
        {
            P1=0x00;
```

```
            }
            if(x&0xc0==0x40)
            {
                P1=0x07;
            }
            if(x&0xc0==0x80)
            {
                P1=0x2a;
            }
            if(x&0xc0==0xc0)
            {
                P1=0x15;
            }
        }
}
/*-------------Interrupt Reception Function of Serial Port-------------*/
void Uart_int(void) interrupt 4
{
    RI=0;
    x=SBUF;
}
```

3. Multiprocessor Communication

Mode 2 and Mode 3 of STC15W201S MCU serial port are applied in a specific field, multiprocessor communication, which is a kind of master-slave multiprocessor communication that consists of one master processor and several slave processors. The data from master processor can be sent to all the slave processors or specified slave processor and all the data from slave processor can be received by master processor only. Communication between slave processors can't be carried out. The schematic diagram of multiprocessor communication is shown in Figure 8.15.

3. 多机通信

STC15W201S 单片机串行口的方式 2 和方式 3 有一个专门的应用领域，即多机通信。这一功能通常采用主从式多机通信方式，包含一台主机和多台从机。主机发送的信息可以传送到各个从机或指定的从机，各从机发送的信息只能被主机接收，从机与从机之间不能进行通信。图 8.15 是多机通信的连接示意图。

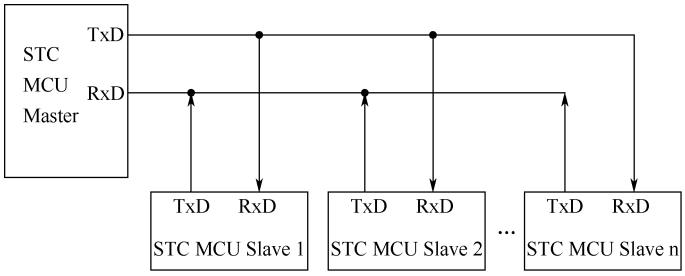


Figure 8.15　Schematic Diagram of Multiprocessor Communication

多机通信连接示意图

1）Implementation

The implementation relies on the appropriate configuration of master and slave, the state of SM2 and the 9th bit data (TB8 or RB8) in the data of transmission or reception. When the serial port reception works in Mode 2 or Mode 3, there are two situations:

(1) If SM2=1, enable multiprocessor com -munication. Set RI and send interrupt request to CPU when 9th bit data (RB8) of reception is "1"; otherwise, doesn't set RI and send no interrupt request to CPU, the data will be lost, that is no data can be accepted.

(2) If SM2=0, Set RI no matter the 9th bit data is "0" or "1", that is reception begins.

2）Define the address number of all the slave processors before programming

256 slaves are supported in system, which address from 00H to FFH. When the master processor wants to transmit a block of data to one of slaves, it first sends out an address byte which identified the target slave. The process of multiprocessor communication is summarized as below:

(1) An address frame is sent by master, connecting to appropriate slave. Set the TB8 of master at "1", indicating the data is address frame. For instance:

```
SCON=0xC8;      /* Set serial port working in Mode 3, TB8=1, send address data
                   设串行口为方式 3，（TB8）=1，发送地址数据信息*/
```

(2) Set SM2 of each slave to "1", preparing for receiving a frame of address. For instance:

```
SCON=0xF0;      /* Set serial port working in Mode 3, SM2=1,REN=1, enable receiving
                   设串行口为方式 3，（SM2）=1，（REN）=1，允许接收*/
```

(3) An address byte will interrupt all the slaves, so that each slave should examine the received byte

1）多机通信的实现

多机通信的实现主要依靠主、从机之间正确地设置与判断 SM2 和发送或接收的第 9 位数据（TB8 或 RB8）来完成。在单片机串行口以方式 2 或方式 3 接收时，有两种情况：

（1）若 SM2=1，表示启用多机通信功能，当接收到的第 9 位数据（RB8）为 1 时，会置位 RI 标志，向 CPU 发中断请求；当接收到第 9 位数据为 0，即不会置位 RI 标志，不产生中断，信息将被丢弃，即不能接收数据。

（2）若 SM2=0，则接收到的第 9 位信息无论是 1 还是 0，都会置位 RI 中断标志，即接收数据。

2）编程前要给各从机定义地址编号

系统中允许接 256 台从机，地址编码为 00H～FFH，在主机想发送一个数据块给某个从机时，它必须先送出一个地址字节，以辨认从机。多机通信的过程简述如下：

（1）主机发送一帧地址信息，与所需的从机联络。主机应置 TB8 为 1，表示发送的是地址帧。例如：

（2）所有从机的 SM2=1，处于准备接收一帧地址信息的状态。例如：

（3）各从机都能接收到地址信息，每个从机要判断主机送过来的

and check whether it matches the address of its own. If it matches, the slave will clear its SM2 bit and prepare to receive the data byte that will be coming. The other slaves leave their SM2s set and ignore the data received until the next address frame comes.

(4) Master sends control instruction and data to addressed slave. The TB8 of master should be set to "0", indicating sending data or control instruction. The slave, which is not addressed, would issue interrupt because SM2=1 and RB8=0, and ignores the data from master. While the addressed slave receives the data from master properly.

地址信息与自己的地址是否相符。对于地址相符的从机，清零 SM2，以便接收主机随后发来的数据信息。对于地址不相符的从机，保持 SM2 为 1 的状态，对主机随后发来的信息不理睬，直到发送新的一帧地址信息。

（4）主机发送控制指令和数据信息给被寻址的从机。其中主机置 TB8 为 0，表示发送的是数据或控制指令。对于没被选中的从机，因为 SM2=1，RB8=0，所以不会产生中断，对主机发送的信息不接收；而符合地址的从机，因 SM2 为 0，能正常接收主机发过来的数据。

8.2 Communication between STC15W201S MCU and PC STC15W201S 单片机与 PC 的通信

8.2.1 Serial Communication Interface between MCU and RS-232 of PC 单片机与 PC 的 RS-232 串行通信接口设计

In MCU application system, asynchronous serial communication is used in the communication with upper computer. In the design of communication interface, standard interface should be selected as required, and such matter as transmission medium and level conversion should be taken into consideration. MCU and peripheral, measuring instrument can be linked organically to form a measure and control system. For example, RS-232 is used for level conversion when MCU communicates with PC.

There mainly are three groups of asynchronous serial communication interface: RS-232 interface; RS-449, RS-422 and RS-485 interface; 20mA current loop.

1. RS-232C Interface

RS-232C is such a kind of asynchronous serial

在单片机应用系统中，与上位机的数据通信主要采用异步串行通信。在设计通信接口时，必须根据需要选择标准接口，并考虑传输介质、电平转换等问题。采用标准接口后，能够方便地把单片机和外设、测量仪器等有机地连接起来，从而构成一个测控系统。例如当需要单片机和 PC 机通信时，通常采用 RS-232 接口进行电平转换。

异步串行通信接口主要有三类：RS-232 接口；RS-449、RS-422 和 RS-485 接口，以及 20mA 电流环。

1. RS-232C 接口

RS-232C 是使用最早、应用最

communication bus standard that it is one of earliest used and most widely used. It's released in 1962 and revised in 1969 by Electronic Industries Association (EIA), where RS represents Recommended Standard, 232 is the identification number and C stands for the last revise.

RS-232C is usually used for defining the electrical performance between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) in computer system. It is adopted often in the communication between 51 MCU and PC.

RS-232C serial interface bus applies to such scenes as communication distance is less than 15m and data rate is less than 20kB/s.

1) Standard formation of RS-232C message

RS-232C adopts serial formation, as shown in Figure 8.16. Here are the regulations: the beginning of message is start bit, the last is stop bit; the message itself can be 5, 6, 7 or 8 bits plus one parity bit. If there is no information between two messages, fulfill "1", indicating void.

多的一种异步串行通信总线标准。它是美国电子工业协会(EIA)1962年公布、1969年最后修订而成的。其中RS表示Recommended Standard，232是该标准的标识号，C表示最后一次修订。

RS-232C主要用来定义计算机系统的一些数据终端设备（DTE）和数据电路终端设备（DCE）之间的电气性能。51单片机与PC机的通信通常采用该种类型的接口。

RS-232C串行接口总线适用于设备之间的通信距离不大于15米，传输速率最大为20kB/s的应用场合。

1）RS-232C信息格式标准

RS-232C采用串行格式，如图8.16所示。该标准规定：信息的开始为起始位，信息的结束为停止位；信息本身可以是5、6、7、8位再加一位奇偶位。如果两个信息之间无信息，则写"1"，表示空。

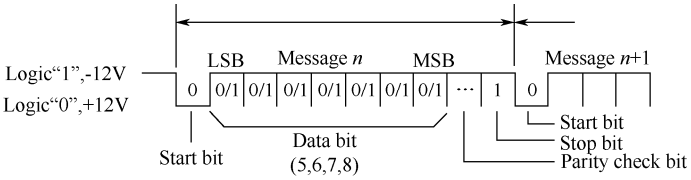


Figure 8.16 Formation of RS-232C Message

RS-232C信息格式

2) Level converter of RS-232C

RS-232C provides electrical standard of its own. Its level is not +5V or GND, but negative logic, because it's developed before TTL circuit. That is:

Logic "0": +5V～+15V.

Logic "1": -5V～-15V.

Thus, RS-232C can't connect to TTL level directly, unless level conversion is carried out in use,

2）RS-232C电平转换器

RS-232C规定了自己的电气标准，由于它是在TTL电路之前研制的，所以它的电平不是+5V和地，而是采用负逻辑，即：

逻辑"0"：+5V～+15V。

逻辑"1"：-5V～-15V。

因此，RS-232C不能和TTL电平直接相连，使用时必须进行电

or it will burn TTL circuit, which must be taken into consideration.

Currently, common level conversion circuit is MAX232 or STC232. The logic structure is shown as Figure 8.17.

平转换，否则将使 TTL 电路烧坏，实际应用时必须注意！

目前，常用的电平转换电路是 MAX232 或 STC232，MAX232 的逻辑结构图如图 8.17 所示。

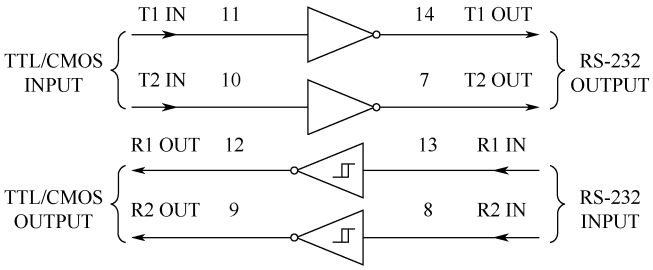


Figure 8.17　Pins of MAX232

MAX232 功能引脚图

3)　Regulation of RS-232C Bus

There 25 lines in RS-232C standard bus, with a connector of 25 pins. The definitions of all the 25 pins are shown in Table 8.7.

3）RS-232C 总线规定

RS-232C 标准总线为 25 根，使用 25 个引脚的连接器，各信号引脚的定义如表 8.7 所示。

Table 8.7　RS-232C Standard Bus

RS-232C 标准总线

Pin	Definition	Pin	Definition
1	Protect GND (PG)	14	Secondary Transmitted Data (STxD)
2	Transmit Data (TxD)	15	Transmit Clock（TxC）
3	Received Data（RxD）	16	Secondary Received Data (SRxD)
4	Request to Send（RTS）	17	Received Clock（RxC）
5	Clear to Send（CTS）	18	Undefined
6	Data Set Ready（DSR）	19	Secondary Request To Send (SRTS)
7	Signal Ground（SG）	20	Data Terminal Ready（DTR）
8	Data Carrier Detection（DCD）	21	Signal Detection (SD)
9	Receiving Line Detection	22	Ring Indicator(RI)
10	Line Detection	23	Data Rate Selection
11	Undefined	24	Transmit Clock（TxC）
12	Secondary Carrier Detect (SDCD)	25	Undefined
13	Secondary Clear To Send (SCTS)		

The mechanical characteristic of connector：Because the mechanical characteristic of connector isn't defined in RS-232C, thus, many different kinds of connector came out, with different pin definitions.

连接器的机械特性：由于 RS-232C 并未定义连接器的物理特性，因此，出现了各种类型的连接器，其引脚的定义也各不相同。下面分别

Here are two kinds of connector.

① DB-25: The shape and signal cable distribution are shown in Figure 8.18 (a), and the pins' functions are the same as Table 8.4.

介绍两种连接器。

① DB-25：DB-25 型连接器的外形及信号线分配如图 8.18(a)所示，各引脚功能与表 8.4 一致。

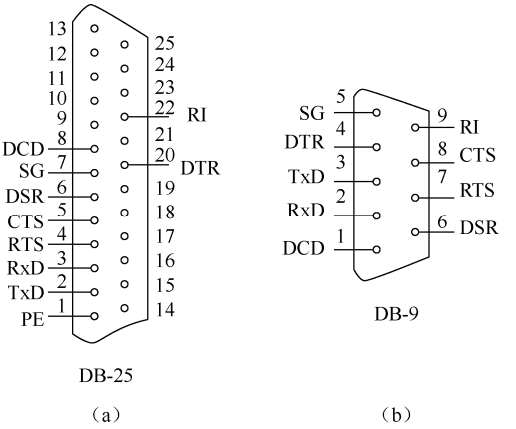


Figure 8.18 Pins of DB-9 Connector and DB-25 Connector

DB-9、DB-25 连接器引脚图

② DB-9: DB-9 connector provides 9 signals for asynchronous communication only, as shown in Figure 8.18 (b). The pins distribution of DB-9 connector is different from DB-25. Therefore, specific cable must be used in the connection to the DCE with DB-25 connector.

When the communication rate is less than 20kB/s, the maximum physical distance for RS-232 is 15m (50 feet).

2. RS-232C Interface Interact with STC15W201S Communication Interface

Asynchronous communication adapter, of which asynchronous communication can be archived in the use, is often equipped in PC system. The core component of adapter is programmable Intel 8250 chip, which provides PC the ability to communicate with other computers or devices with standard RS-232C interface. STC15W201S MCU equips with full duplex serial port itself, with which drive circuit and isolation circuit accompany can form a simple and feasible

② DB-9 连接器：DB-9 连接器只提供异步通信的 9 个信号，如图 8.18(b)所示。DB-9 型连接器的引脚分配与 DB-25 型引脚信号完全不同。因此，若与配接 DB-25 型连接器的 DCE 设备连接，必须使用专门的电缆线。

在通信速率低于 20kB/s 时，RS-232C 所直接连接的最大物理距离为 15m（50 英尺）。

2. RS-232C 接口与 STC15W201S 的通信接口设计

在 PC 系统内都装有异步通信适配器，利用它可以实现异步串行通信。该适配器的核心组件是可编程的 Intel 8250 芯片，它使 PC 有能力与其他具有标准的 RS-232C 接口的计算机或设备进行通信。STC15W201S 单片机本身具有一个全双工的串行口，因此只要配以电平转换的驱动电路、隔离电路就可组成一个简单可行的通信接口。同样，PC 和单片机之间

communication interface. The communications between PC and MCU are divided into dual -processor and multiprocessor communication.

The simplest hardware connection between PC and MCU is Zero Modulation Tri-lane Economical, which occupies the least lanes for full duplex communication. Only three pins of the 9-pin serial port are used: GND of the 5th pin, RxD of the 2nd pin and TxD of the 3rd pin, as shown in Figure 8.19, which also is the program download circuit of STC15W201S MCU.

的通信也分为双机通信和多机通信。

PC 和单片机进行串行通信的硬件连接，最简单的连接是零调制三线经济型，这是进行全双工通信所必需的最少线路，计算机的 9 针串口只连接其中的三根线：第 5 脚的 GND，第 2 脚的 RxD，第 3 脚的 TxD，如图 8.19 所示。这也是 STC15W201S 单片机的程序下载电路。

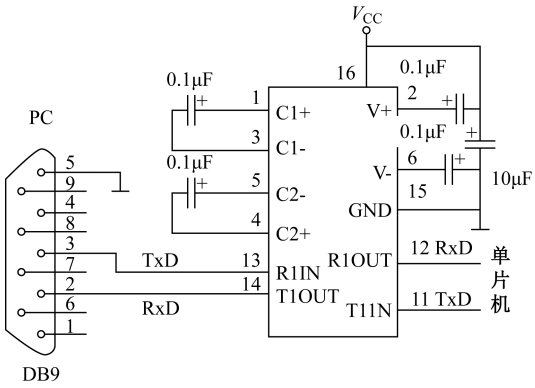


Figure 8.19　Tri-lane Connection Circuit for the Serial Communication between PC and MCU

PC 和单片机串行通信的三线制连接电路

8.2.2　Communication Interface of USB between MCU and PC
单片机与 PC 的 USB 总线通信接口设计

Nowadays, the typical serial communication interface is USB. RS-232C is not standard equipment any more. Thus, for serial communica -tion between modern PC and 51 MCU, CH340G is used for converting USB to UART, which is adopting USB bus simulating UART communica -tion. The circuit of USB bus to UART conversion is shown in Figure 8.20.

目前，PC 常用串行通信接口是 USB 接口，绝大多数已不再将 RS-232C 串行接口作为标配了。为此，为了现代 PC 能与 51 单片机进行串行通信，采用 CH340G 芯片将 USB 总线转串口 UART，采用 USB 总线模拟 UART 通信。USB 总线转 UART 电路如图 8.20 所示。

NOTE：In practice use, install the driver for USB converting to UART first, and then check the serial port number of which is USB converting to after the driver's installation. Such port number could be used for communication the same way as RS-232 serial port.

注：使用时，先安装 USB 转串口的驱动程序，安装成功后在 PC 的设备管理器中查看 USB 转串口的串口号，以后就可以采用此串口号按 RS-232 串口的一样的方法进行串口通信了。

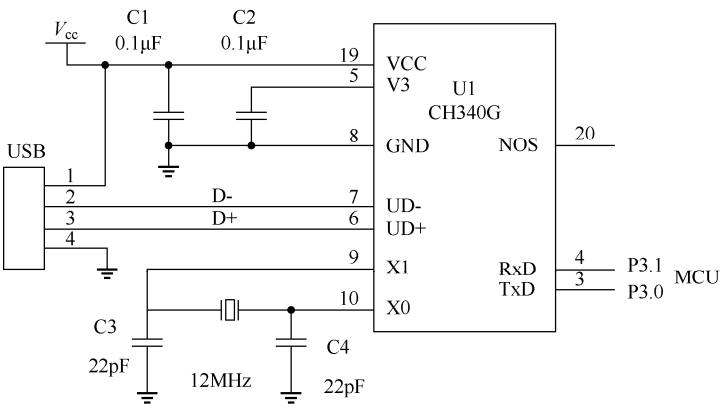


Figure 8.20　Circuit of USB Converting to Serial Port (TTL)

USB 转串口（TTL）电路

8.2.3　Program Design of Serial Communication between MCU and PC　单片机与 PC 串行通信的程序设计

Communication program design is divided into computer (upper machine) program design and MCU (lower machine) program design.

In order to implement serial port communication between MCU and PC, corresponding serial port communication program should be developed in such high level programming languages as VC and VB. In the actual development of debugging the serial port communication of MCU end, we can use the serial port debug program embedded in STC series MCU download program or other serial port debugging software (such as CommTone) to emulate the serial port communication program of PC end. That is the widely used method in actual project development, especially for a team.

Serial port debug program, without any programming, can achieve serial communication of RS-232C, which can make serial port debugging more transparent and more convenient and improve work

通信程序设计分为计算机（上位机）程序设计与单片机（下位机）程序设计。

为了实现单片机与 PC 的串口通信，PC 端需要开发相应的串口通信程序，这些程序通常采用各种高级语言来开发，比如 VC、VB 等。在实际开发调试单片机端的串口通信程序时，我们也可以使用 STC 系列单片机下载程序中内嵌的串口调试程序或其他串口调试软件（如串口调试精灵软件）来模拟 PC 端的串口通信程序。这也是在实际工程开发中，特别是团队开发时常用的办法。

串口调试程序不需要任何编程即可实现 RS-232C 的串口通信，能有效提高工作效率，使串口调试能够方便、透明地进行。

efficiency. It can set variable communication rates, parity check and communication port online without restart program. It supports transmitting data in hexadecimal (HEX) and ASCII code format and the timing of data for transmitting and interval can be set. It can display the data received automatically in HEX or ASCII format. It's such an essential tool for engineering and technical personnel monitoring and debugging serial port program.

Set serial port and communicate with PC in the use of serial port according to the function requirements of different projects in MCU program design.

E.g. Transmit the input from keyboard to MCU; loopback the same data to PC after receiving the data and then display in the monitor. Program a MCU communication application of which PC makes use to transmit and receive data and display.

Solution: The both sides of communication agree: baud rate is 2400; information format is 8-bit digit, with one stop bit and no parity check bit. Let the system frequency of crystal oscillator is 11.0592MHz.

C51 reference program (pc_uart.c) is shown as below:

它可以在线设置各种通信速率、奇偶校验、通信口而不用重新启动程序。发送数据可发送十六进制(HEX)数据和 ASCII 码，可以设置定时发送的数据以及时间间隔。可以自动显示接收到的数据，支持 HEX 或 ASCII 码显示，是工程技术人员监视、调试串口程序的必备工具。

单片机程序设计根据不同项目的功能要求，设置串口并利用串口与 PC 进行数据通信。

例　将 PC 键盘的输入发送给单片机，单片机收到数据后，回送同一数据给 PC，并在屏幕上显示出来。PC 端采用串口调试程序进行数据发送与接收数据并显示，请编写单片机通信程序。

解：通信双方约定：波特率为 2400 ;信息格式为 8 个数据位，1 个停止位，无奇偶校验位。设系统晶振频率为 11.0592MHz。

C51 参考程序（pc_uart.c）如下：

```
#include <reg51.h>
unsigned  char  temp;
/*--------2400bps@11.0592MHz Serial Port Initialization Function---------*/
void UartInit(void)        //2400bps@11.0592MHz
{
    SCON = 0x40;                /* Mode 1, 8-bit digit data, variable baud-rate
                                   方式 1，8 位数据，可变波特率*/
    AUXR |= 0x04;               /* The clock of T2 is fosc, that 1/T
                                   定时器 2 时钟为 fosc，即 1/T*/
    T2L = 0x80;                 // Set initial value of timer    设定定时初值
    T2H = 0xFB;                 // Set initial value of timer    设定定时初值
    AUXR |= 0x01;               /* Serial Port 1 select the T2 as baud-rate generator
                                   串口 1 选择定时器 2 为波特率发生器*/
    AUXR |= 0x10;               // Start T2    启动定时器 2
    ES=1;                       // Enable serial port interrupt    开放串行口中断
```

```
        EA=1;                       // Enable the global interrupt    开放总中断
        REN=1;                      // Start receiving    启动接收
    }
    /*----------------- Serial Interrupt Service Subroutine-----------------*/

    void Uart_int(void) interrupt 4
    {
        RI =0 ;                     // Clear serial receiving flag    串行接收标志清零
        temp = SBUF;                // Receiving    接收数据
        SBUF = temp;                // Send the data received    发送接收到的数据
        While(TI==0);               // Waiting for the end of transmission    等待发送结束
        TI = 0;                     // Clear TI    清零 TI
    }

    /*-----------------Main Function-----------------*/
    void main(void)
    {
        Uart_int();          // Calling the serial initialization function    调用串口初始化函数
        while(1);
    }
```

8.3 Relay Broadcast Mode of Serial Port
串行口的中继广播方式

The relay broadcast mode is that the output of TxD pin can reflect the input level state of RxD pin in real time.

The relay broadcast mode of STC15W201S MCU serial port is configured by setting B4 bit of the SFR CLK_DIV. The format of CLK_DIV is shown in Table 8.8.

所谓串行口的中继广播方式是指单片机串行口发送引脚（TxD）的输出可以实时反映串行口接收引脚（RxD）输入的电平状态。

STC15W201S 单片机串行口具有中继广播功能，它是通过设置 CLK_DIV 特殊功能寄存器的 B4 位来实现的。CLK_DIV 的格式如表 8.8 所示。

Table 8.8 CLK_DIV

CLK_DIV	Address 地址	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value 复位值
	97H	MCKO_S1	MCKO_S0	-	Tx_Rx	-	CLKS2	CLKS1	CLKS0	0000 x000

Tx_Rx: The setting bit of relay broadcast mode.
0: serial port works on normal mode;
1: serial port works on relay broadcast mode.

Tx_Rx：串行口中继广播方式设置位。Tx_Rx=0，串行口为正常工作方式；Tx_Rx=1，串行口 1 为中继广播方式。

8.4 Switch of Hardware Pins of Serial Port 串行口硬件引脚的切换

The hardware pin switch within the transmission and reception of different serial ports is implemented by configuring the S1_S0 bit in the SFR P_SW1 (AUXR1). The format of P_SW1 (AUXR1) is shown in Table 8.9.

通过对特殊功能寄存器 P_SW1 (AUXR1)中的 S1_S0 位进行控制，可实现串行口的发送与接收硬件引脚在不同口进行切换。P_SW1(AUXR1)的数据格式如表 8.9 所示。

Table 8.9 P_SW1(AUXR1)

P_SW1 (AUXR1)	Address 地址	D7	D6	D5	D4	D3	D2	D1	D0	Reset Value 复位值
	A2H	-	S1_S0	-	-	-	-	-	-	x0xx xxxx

The hardware pin switch of serial port transmission and reception is controlled by S1_S0, as shown in Table 8.10.

串行口发送、接收硬件引脚切换由 S1_S0 进行控制，具体的切换情况如表 8.10 所示。

Table 8.10 Hardware Pin Switch of Serial Port
串行口硬件引脚切换

S1_S0	Serial Port	
	TxD	RxD
0	P3.1	P3.0
1	P3.7（TxD_2）	P3.6（RxD_2）

Summary 本章小结

Serial communication is often used in the information exchange of the Distributed Control and Multiprocessor System and modern Measuring and Controlling System, such as the ISP online programming of STC series MCU. There are two kinds of serial communication, asynchronous and synchronous. Asynchronous communication is transmitting by character, using the start bit to synchronization between transmitting and receiving in each character's transmission. Synchronous communication is transmitting by data block, sending a synchronous

pulse for data synchronization in transmission to ensure the full synchronization between transmitting and receiving, which requires the same clock both in transmitter and receiver. The synchronous communication can improve the transmission rate (up to 56KB/s or higher), but more complicate.

According to the direction of data flow at a time, there are three transfer modes in serial communication: Simplex, Half-duplex and Full-duplex.

STC15W201S MCU is equipped with a programmable serial port.

There exist four working modes in serial port: synchronous shift register input/output mode, 8-bit asynchronous communication mode and two 9-bit communication modes with different baud rate. The baud rates of Mode 0 and Mode 2 are fixed, while are variable in Mode 1 and Mode 3, which is determined by the overflow rate of T2. Mode 0 is used for expending I/O port, Mode 1 for implementing 8-bit UART, Mode 2 and Mode 3 for implementing 9-bit UART.

The hardware transmitting and receiving pins of serial port are P3.1 and P3.0 by default, while they can be switched to P3.7 and P3.6 by software.

By using serial communication, dual-processor communication or multiprocessor commu -nication can be achieved not only between or among MCUs, but also between or among MCUs and PCs.

STC15W201S is equipped with relay broadcast function, where the output of TxD pin can reflect the input level state of RxD pin in real time. A line shift register is formed in the serial port and it can improve the time consumption of received and sending mode by using serial port.

RS-232C is a communication interface that widely used with less line and variable data rate, but a mere dozens of meter's signal transmission distance. RS-422A and RS-485 communication interface use the differential circuit transmission, which has a better transmission rate and distance.

In the design practice of Industrial Control System (especially the Multi-point Filed Industrial Control System), a distributed control system that consists of MCUs and PCs is an important future developing direction.

Exercise 练习题

1. Completions

1) The communication modes of microcomputer are including_____ and serial communication.

2) Serial communication, according to the direction of data transporting, can be divided into 3 modes, _____, half-duplex and _____.

3) Serial communication, according to the type of synchronous clock, can be divided into 2

modes, _____, and synchronous serial communication.

4) Asynchronous serial communication transports in character frame. Every character frame include_____, data bit and_____.

5) The start bit is_____ and stop bit is_____ in asynchronous serial communication.

6) The serial communication port in STC15W201S consists of 2 _____, 1 shift register, 1 serial port control register and 1 _____.

7) The data buffer for serial port of STC15W201S MCU is_____. Actually one address is mapping to 2 registers. When writing the data buffer, sending start construction at the same time, the address is mapping to_____ data register; When reading the data buffer, the address is mapping to_____ data register.

8) There are 4 working modes for serial port of STC15W201S MCU, Mode 0 is_____, Mode 1 is_____, Mode 2 is_____ and Mode 3 is_____.

9) The control bit if multiprocessor communication of STC15W201S MCU serial port is_____.

10) The baud rate of STC15W201S MCU serial port Mode 0 is_____, of Mode1 and Mode 3 is_____ and of Mode 2 is_____.

11) There are 2 interrupt request flags in STC15W201S MCU serial port interrupt. Sending interrupt request flag is_____, and receiving interrupt request flag is_____.

2. Choice Questions

1) After the execution of "SM0=0、SM1=1;", STC15W201S MCU serial port is working at_____.

A. Mode 0　　B. Mode 1　　C. Mode 2　　D. Mode 3

2) SM0 and SM1 should be set to be_____ and_____ to make STC15W201S MCU serial port work at Mode 2.

A. 0, 0　　B. 0, 1　　C. 1, 0　　D. 1, 1

3) When serial reception is enabled in STC15W201S MCU serial port, it will not set serial reception interrupt request RI when serial reception is over if_____.

A. SM2=1, RB8=1　　B. SM2=0, RB8=1

C. SM2=1, RB8=0　　D. SM2=0, RB8=0

4) When STC15W201S MCU serial port is working at Mode 2 or Mode 3, _____ should be set to 1 before serial transmitting if the 9th bit of transmit data is 1.

A. RB8　　B. TB8

C. TI　　D. RI

5) When STC15W201S MCU serial port is working at Mode 2 or Mode 3, TB8 should be_____ if transmit data is parity checked.

A. Set 1　　B. Set 0

C. Set =P　　D. Set = $\overline{P}$

6) How many bits are there in a character frame when STC15W201S MCU serial port is working at Mode 1? (　　)

A. 8　　B. 9　　C. 10　　D. 11

3. True or False

1) The clocks of sending and receiving must be synchronous in synchronous serial communication. (　　)

2) In asynchronous serial communication, sending and receiving could have their own synchronous clock, but the speed of their communication should be consistent. (　　)

3) When STC15W201S MCU serial port is working at Mode 0 or Mode 2, the value of S1ST2 will not influence the baud rate. (　　)

4) When STC15W201S MCU serial port is working at Mode 0, control bit SMOD in PCON will influence the baud rate. (　　)

5) When STC15W201S MCU serial port is working at Mode 1, control bit SMOD in PCON will influence the baud rate. (　　)

6) When STC15W201S MCU serial port is working at Mode 1 or Mode 3, T1 will be selected as a baud rate generator when S1ST2=1. (　　)

7) When STC15W201S MCU serial port is working at Mode 1 or Mode 3, when SM2=1 and the 9th bit of transmit data is 1, serial receiving interrupt flag RI will not be set 1. (　　)

8) The enabling bit for STC15W201S MCU serial port serial receiving is REN. (　　)

9) There are also 4 working modes for STC15W201S MCU serial port 2, port 3 and port 4. (　　)

10) There are 4 working modes for STC15W201S MCU serial port 1, but only 2 modes for port 2, port 3 and port 4. (　　)

11) In the application of STC15W201S MCU, both serial transmitting and receiving pin for serial port are fixed. (　　)

12) Via program, the output signal from serial transmitting pin of STC15W201S MCU serial port can reflect the input signal from serial receiving pin on real time. (　　)

4. Question and Answer

1) What are the 2 kinds of working mode in microcomputer data communication? And what about their features?

2) What is the data format of the character frame in asynchronous serial communication?

3) What is the baud rate? How to get STC15W201S MCU serial port baud rate application from in-system program tool STC-ISP?

4) What are the 4 kinds of working mode in STC15W201S MCU serial port? What’re their setting and features?

5) Please make a brief description of the similarities and differences between serial port

working Mode 2 and Mode 3.

6) What are the 2 kinds of working mode in STC15W201S MCU serial port 2, port 3 and port4? What're their setting and features?

7) Please make a brief description of implementation of multiprocessor communication of STC15W201S MCU.

8) Please make a brief description of implementation of broadcast relay of STC15W201S MCU.

Unit 9　Comparator of STC15W201S MCU

第 9 单元　STC15W201S 单片机比较器

Comparator is built in STC15W series MCU, such as STC15W408AD series, STC15W201S series, STC15W408S series, STC15W1K16S series and STC15W4K32S4 series.

STC15W 系列单片机（如 STC15W408AD 系列、STC15W201S 系列、STC15W408S 系列、STC15W1K16S 系列及 STC15W4K32S4 系列）内置了比较器。

9.1　Internal Structure and Control of Comparator
比较器的内部结构与控制

1．Internal Structure of Comparator

The internal structure of STC15W201S comparator consists of Operational Amplifier Comparator, Filter and Interrupt flag production (including interrupt enable control), as shown in Figure 9.1.

1．比较器的内部结构

STC15W201S 单片机比较器的内部结构如图 9.1 所示，由集成运放比较电路、过滤电路、中断标志形成电路（含中断允许控制）3 部分电路组成。

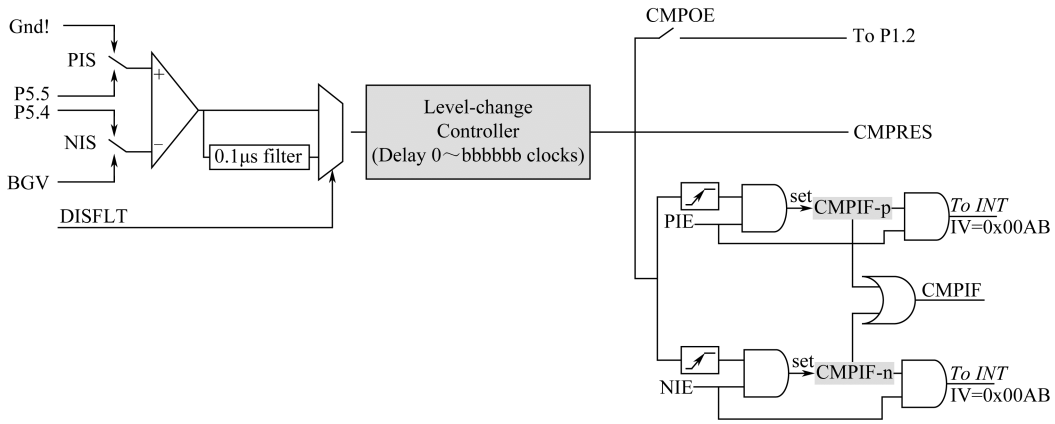


Figure 9.1　Internal Structure of STC15W201S Comparator　STC15W201S 单片机比较器的内部结构

1) Operational amplifier comparator

Whether the in-phase signal and reversed -phase signal are connecting to internal signal or external signal can be selected through Compare Control Register 1 (CMPCR1). The output of Operational Amplifier Comparator becomes a stable comparator output after filter.

2) Filter (or de-jitter)

The jump transition won't be accepted immediately, but after a certain delay.

3) Interrupt flag formation

The selection of interrupt types, the formation of interrupt and interrupt flag enabling are configured in CMPCR1.

2. Control Register of Comparator

The comparator of STC15W201S is controlled and managed by CMPCR1 and CMPCR2.

1) Compare Control Register 1 (CMPCR1)

The definition of CMPCR1 is shown in Table 9.1.

1）集成运放比较电路

集成运放的同相、反相输入端的输入信号可通过 CMPCR1 比较器控制寄存器 1 进行选择，是接内部信号，还是外接输入信号；集成运放比较电路的输出通过滤波器形成稳定的比较器输出信号。

2）滤波（或称去抖动）电路

当比较电路输出发生跳变时，不立即认为是跳变，而是经过一定延时后，再确认是否为跳变。

3）中断标志形成电路

中断标志类型的选择、中断标志的形成以及中断标志的允许在 CMPCR1 中设置。

2. 比较器的控制寄存器

STC15W201S 单片机比较器由 CMPCR1 比较器控制寄存器 1 和 CMPCR1 比较器控制寄存器 2 进行控制管理。

1）比较器控制寄存器 1(CMPCR1)

CMPCR1 的格式如表 9.1 所示。

Table 9.1 CMPCR1

CMPCR1	Address 地址	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value 复位值
	E6H	CMPEN	CMPIF	PIE	NIE	PIS	NIS	CMPOE	CMPRES	0000 0000

(1) CMPEN: Enable bit of Comparator.

0: Disable comparator, comparator power down.

1: Enable comparator.

(2) CMPIF: Comparator Interrupt Flag.

When CMPEN = 1:

① If the comparing result has changed from low to high and PIE has been set to 1, a built-in register bit PIF would be set to 1.

② If the comparing result has changed from high to low and NIE has been set to " 1 ",

（1）CMPEN：比较器模块使能位。

CMPEN=0，禁用比较器模块，比较器的电源关闭。

CMPEN=1，使能比较器模块。

（2）CMPIF：比较器中断标志位。

在 CMPEN 为 1 的情况下：

① 当比较器的比较结果由低变成高时，若是 PIE 被设置成 1，那么内建 PIF 标志置 1。

② 当比较器的比较结果由高变成低时，若是 NIE 被设置成 1，那么内建

a built-in register bit NIF would be set to “ 1 ”.

When CPU reads the value of CMPIF, it gets the result of (PIF || NIF);When CPU writes “ 0 ” into CMPIF, PIF and NIF will be cleared.

(3) PIE: Pos-edge interrupt enabling bit.

0: disable the comparator interrupt responding to the comparing result changed from low to high.

1: enable the comparator interrupt responding to the comparing result changed from low to high; set CMPIF and sent interrupt request to CPU.

(4) NIE: Neg-edge interrupt enabling bit.

0: disable the comparator interrupt responding to the comparing result changed from high to low.

1: enable the comparator interrupt responding to the comparing result changed from high to low; set CMPIF and sent interrupt request to CPU.

(5) PIS: Bit to choose the positive pole input of comparator.

0: choose external pin P5.5 as the positive pole input of comparator.

1: choose Gnd! as the positive pole input of comparator.

NOTE：Gnd! is internal ground here.

注：Gnd！是指内部地。

(6) NIS: Bit to choose the negative pole of comparator.

0: choose internal BandGap Voltage (BGV) as the negative pole input of comparator.

1: choose the external pin P5.4 as the negative pole input of comparator.

NOTE：The internal BandGap Voltage (BGV) is measured by channel 9 of ADC; There is no A/D transfer module in STC15W201S MCU; “1” is unavailable in NIS.

注：内部 BandGap 电压 BGV 是通过 ADC 第 9 通道测到的，STC15W201S 单片机无 A/D 转换模块，NIS 置 1 选项无效。

NIF 标志置 1。

当 CPU 读取 CMPIF 的数值时，读到的是 PIF 与 NIF 的或；当 CPU 对 CMPIF 写入 0 后，PIF 以及 NIF 都会被清零。

（3）PIE：比较器上升沿中断使能位。

（PIE）= 0，禁用比较器由低变高事件设定比较器中断。

（PIE）= 1，使能比较器由低变高时，置位 CMPIF，并向 CPU 申请中断。

（4）NIE：比较器下降沿中断使能位。

（NIE）= 0，禁用比较器由高变低事件设定比较器中断。

（NIE）= 1，使能比较器由高变低时，置位 CMPIF，并向 CPU 申请中断。

（5）PIS：比较器正极选择位。

（PIS）= 0，选择外部 P5.5 为比较器的正极输入源。

（PIS）= 1，选择 Gnd！作为比较器的正极输入源。

（6）NIS：比较器负极选择位。

（NIS）= 0，选择内部 BandGap 电压 BGV 为比较器的负极输入源。

（NIS）= 1，选择外部引脚 P5.4 为比较器的负极输入源。

(7)CMPOE: Comparing result output enabling bit.

0: Disable the comparing result outputting of comparator.

1: Enable the comparing result of comparator outputting to P1.2.

(8)CMPRES: Flag bit of comparator result.

0: the level of CMP+ is lower than CMP- (or the reference voltage of internal BandGap).

1: the level of CMP+ is higher than CMP- (or the reference voltage of internal BandGap).

（7）CMPOE：比较结果输出控制位。

（CMPOE）= 0，禁止比较器的比较结果输出。

（CMPOE）= 1，允许比较器的比较结果输出到 P1.2。

（8）CMPRES：比较器比较结果(Comparator Result)标志位。

（CMPRES）= 0，CMP+的电平低于 CMP-的电平（或内部 BandGap 参考电压的电平）。

（CMPRES）= 1，CMP+的电平高于 CMP-的电平（或内部 BandGap 参考电压的电平）。

NOTE：The bit CMPRES is read-only, so it doesn't make sense to be written by software. And the value read from CMPRES is the result after "filter" control, not the direct output of Operational Amplifier Comparator.

注：CMPRES 是一个只读位，软件对它进行写入的动作没有任何意义。且软件所读到的结果是经过“过滤”控制后的结果，而非集成运放比较电路的直接输出结果。

2) Comparator Control Register 2

The definition of CMPCR2 is shown in Table 9.2.

2）比较器控制寄存器 2

CMPCR2 的格式如表 9.2 所示。

Table 9.2 CMPCR2

CMPCR2	Address	B7	B6	B5	B4	B3	B2	B1	B0	Reset Value
	E7H	INVCMPO	DISFLT	LCDTY[5:0]						0000 1001

(1) INVCMPO: Inverse control bit of comparator output.

0: Normal output the comparing result of comparator on P1.2.

1: Inverse and then output the comparing result of comparator on P1.2.

（1）INVCMPO：比较器输出取反控制位。

（INVCMPO）= 0，比较器正常输出。

（INVCMPO）= 1，比较器取反后再输出到 P1.2。

NOTE：The output of comparator is the result after "filter" control, not the direct output of Operational Amplifier Comparator.

注：比较器的输出为经过“过滤”控制后的结果，而非集成运放比较电路的直接输出结果。

(2) DISFLT: Disable the 0.1μs filter output by comparator.

0: Enable the 0.1μs Filter output by comparator.

1: Disable the 0.1μs Filter output by comparator.

(3) LCDTY[5:0]: Set the duty of Level-change control filter in the output terminal of comparator.

① If the comparing result has changed from low to high, only when the high state has lasted at least LCDTY[5:0] cycles would the comparing result of comparator be affirmed to have changed from low to high, otherwise CPU would consider that there is no change happening and the output of comparator maintains in low level.

② If the comparing result has changed from high to low, only when the low state has lasted at least LCDTY[5:0] cycles would the comparing result of comparator be affirmed to have changed from high to low, otherwise CPU would consider that there is no change happen and the output of comparator maintains in high level.

（2）DISFLT：禁止比较器输出的0.1μs过滤。

DISFLT= 0，比较器的输出有 0.1μs 的 Filter（过滤）。

DISFLT= 1，关掉比较器输出的 0.1μs Filter（过滤）。

（3）LCDTY[5:0]：设置比较器输出端的电平变化控制滤波的工作内容。

① 当比较输出结果由低变高，必须侦测到后来的高电平持续至少 LCDTY[5:0] 个时钟，此芯片线路才认定比较器的输出是由低电平转成了高电平；如果在 LCDTY[5:0]个时钟内，集成运放比较电路的输出又恢复到低电平，此芯片线路认为什么都没发生，视同比较器的输出一直维持在低电平。

② 当比较输出结果由高变低，必须侦测到后来的低电平持续至少 LCDTY[5:0] 个时钟，此芯片线路才认定比较器的输出是由高电平转成了低电平；如果在 LCDTY[5:0]个时钟内，集成运放比较电路的输出又恢复到高电平，此芯片线路认为什么都没发生，视同比较器的输出一直维持在高电平。

9.2 Applications of Comparator 比较器的应用

1. Example for Comparator Interrupt Mode

E.g. P1.0, P1.1 and P1.2 are connected to LED0, LED1 and LED2 individually; LED is low level drive; LED2 is controlled by the output of comparator directly. When there is a positive edge of interrupt request, light up LED0; when there is a negative edge of interrupt request, light up LED1. P1.3 is connected to a switch. When the switch is off, output high level and comparator works in positive edge interrupt request mode, with in-phase output; when the switch is

1. 比较器中断方式程序举例

例 P1.0、P1.1、P1.2 分别接 LED0、LED1、LED2 共 3 只 LED 灯，采用低电平驱动，LED2 直接由比较器输出控制，当有上升沿中断请求时，点亮 LED0，当有下降沿中断请求时，点亮 LED1。P1.3 接 1 只开关，开关断开时输出高电平，比较器处于上升沿中断请求工作方式，直接

on, output low level and comparator works in negative edge interrupt request mode, with reversed-phase output. Interrupt mode program is required.

同相输出；开关合上时输出低电平，比较器处于下降沿中断请求工作方式，直接反相输出。要求中断方式编程。

```
#include "reg51.h"
#include "intrins.h"
sfr CMPCR1 = 0xE6;                  // Compare Control Register 1  比较器控制寄存器1
#define CMPEN 0x80                  //CMPCR1.7 : Enable bit of comparator
                                    //CMPCR1.7 : 比较器模块使能位
#define CMPIF 0x40                  //CMPCR1.6 : Interrupt flag of comparator
                                    //CMPCR1.6 : 比较器中断标志位
#define PIE 0x20                    /*CMPCR1.5 : Enable bit of comparator positive edge
                                      interrupt //CMPCR1.5 : 比较器上升沿中断使能位*/
#define NIE 0x10                    /*CMPCR1.4 : Enable bit of comparator negative edge
                                      interrupt //CMPCR1.4 : 比较器下降沿中断使能位*/
#define PIS 0x08                    /*CMPCR1.3 : Bit to choose the positive pole of
                                      comparator //CMPCR1.3 : 比较器正极选择位*/
#define NIS 0x04                    /*CMPCR1.2 : Bit to choose the negative pole of
                                      comparator //CMPCR1.2 : 比较器负极选择位*/
#define CMPOE 0x02                  //CMPCR1.1 : Control bit of outputting comparing result
                                    //CMPCR1.1 : 比较结果输出控制位
#define CMPRES 0x01                 //CMPCR1.0 : Flag bit of comparator result
                                    //CMPCR1.0 : 比较器比较结果标志位
sfr CMPCR2 = 0xE7;                  // Compare Control Register 2  比较器控制寄存器2
#define INVCMPO 0x80                /*CMPCR2.7 : Control bit for inversing comparator
                                      output //CMPCR2.7 : 比较结果反向输出控制位*/
#define DISFLT 0x40                 /*CMPCR2.6 : Disable the 0.1μs Filter output by
                                      comparator //CMPCR2.6 : 比较器输出端滤波使能控
                                      制位*/
#define LCDTY 0x3F                  /*CMPCR2.[5:0] : The timing control of interval jitter of
                                      comparator output //CMPCR2.[5:0] : 比较器输出的区
                                      抖时间控制*/
sbit LED0 = P1^0;                   // LED for positive edge interrupt request
                                    //上升沿中断请求指示灯
sbit LED1 = P1^1;                   // LED for negative edge interrupt request
                                    //下降沿中断请求指示灯
sbit LED2 = P1^2;                   // LED for direct output of comparator
                                    //比较器直接输出指示灯
sbit K = P1^3;                      // Switch of interrupt request mode
                                    //中断请求方式控制开关
```

```
void cmp_isr() interrupt 21 using 1      // Vector entrance of comparator interrupt
                                         //比较器中断向量入口
{
    CMPCR1 &= ~CMPIF;                    // Clear the finish flag   清除完成标志
    if(K==1)
    {
        LED0=0;                          // Light up LED for positive edge interrupt request
                                         //点亮上升沿中断请求指示灯
        LED0=1;                          //关闭下降沿中断请求指示灯
    }
    Else
    {
        LED1=0;                          // Light up LED for negative edge interrupt request
                                         //点亮下升沿中断请求指示灯
        LED1=1;                          //关闭上升沿中断请求指示灯
    }
}
void main()
{
    CMPCR1 = 0;                          // Initialize CMPCR1   初始化比较器控制寄存器1
    CMPCR2 = 0;                          // Initialize CMPCR2   初始化比较器控制寄存器2
    CMPCR1 &= ~PIS;                      /* Choose external pin P5.5 (CMP+) as the positive pole
                                         input source of comparator   选择外部引脚P5.5（CMP+）
                                         为比较器的正极输入源*/
    CMPCR1 &= ~NIS;                      /* Choose internal BandGap voltage BGV as the negative
                                         pole input source of comparator   选择内部BandGap电
                                         压BGV为比较器的负极输入源*/
    CMPCR1 |= CMPOE;                     // Enable the result output of comparator
                                         //  允许比较器的比较结果输出
    CMPCR2 &= ~DISFLT;                   // Enable the 0.1μs Filter output by comparator
                                         //  不禁用(使能)比较器输出端的0.1μs滤波电路
    CMPCR2 &= ~LCDTY;                    /* Output the result of comparator directly without
                                         de-jitter   比较器结果不去抖动，直接输出*/
    CMPCR1 |= PIE;                       // Enable the positive edge interrupt of comparator
                                         //  使能比较器的上升沿中断

    CMPCR1 |= CMPEN;                     // Enable comparator   使能比较器
    EA = 1;
    while (1)
    {
```

```
        if(K==1)
        {
            CMPCR2 &= ~INVCMPO;   /*Normal output the comparing result of comparator
                                    on P1.2   比较器的比较结果正常输出到P1.2*/
            CMPCR1 |= PIE;        // Enable the positive edge interrupt of comparator
                                  //  使能比较器的上升沿中断
            CMPCR1&=~ NIE;        // Disable the negative edge interrupt of comparator
                                  //  关闭比较器的下降沿中断

        }
        else
        {
            CMPCR2 |= INVCMPO;    /* Reverse the result of comparator and then output
                                    to P1.2比较器的比较结果取反后输出到P1.2*/
            CMPCR1 |= NIE;        // Enable the negative edge interrupt of comparator
                                  //  使能比较器的下降沿中断
            CMPCR1&=~ PIE;        // Disable the positive edge interrupt of comparator
                                  //  关闭比较器的上升沿中断
        }
    }
}
```

2．Example for Comparator Interrupt Query Mode

E.g. P1.0, P1.1 and P1.2 are connected to LED0, LED1 and LED2 individually; LED is low level drive; LED2 is controlled by the output of comparator directly. When there is a positive edge of interrupt request, light up LED0; when there is a negative edge of interrupt request, light up LED1. P1.3 is connected to a switch. When the switch is off, output high level and comparator works in positive edge interrupt request mode, with in-phase output; when the switch is on, output low level and comparator works in negative edge interrupt request mode, with reversed-phase output. Query mode program is required.

2．比较器查询方式程序举例

例　P1.0、P1.1、P1.2 分别接 LED0、LED1、LED2 共 3 只 LED 灯，由低电平驱动，LED2 直接由比较器输出控制，当有上升沿中断请求时，点亮 LED0，当有下降沿中断请求时，点亮 LED1。P1.3 接 1 只开关，开关断开时输出高电平，比较器处于上升沿中断请求工作方式，直接同相输出 ;开关合上时输出低电平，比较器处于下降沿中断请求工作方式，直接反相输出。要求查询方式编程。

```
#include "reg51.h"
#include "intrins.h"
```

```
sfr CMPCR1 = 0xE6;                  // Compare Control Register 1  比较器控制寄存器1
#define CMPEN 0x80                  //CMPCR1.7 : Enable bit of comparator
                                    // CMPCR1.7 : 比较器模块使能位
#define CMPIF 0x40                  //CMPCR1.6 : Interrupt flag of comparator
                                    // CMPCR1.6 : 比较器中断标志位
#define PIE 0x20                    /*CMPCR1.5 : Enable bit of comparator positive edge
                                    interrupt // CMPCR1.5 : 比较器上升沿中断使能位*/
#define NIE 0x10                    /*CMPCR1.4 : Enable bit of comparator negative edge
                                    Interrupt // CMPCR1.4 : 比较器下降沿中断使能位*/
#define PIS 0x08                    /*CMPCR1.3 : Bit to choose the positive pole of
                                    comparator // CMPCR1.3 : 比较器正极选择位*/
#define NIS 0x04                    /*CMPCR1.2 : Bit to choose the negative pole of
                                    comparator // CMPCR1.2 : 比较器负极选择位*/
#define CMPOE 0x02                  //CMPCR1.1 : Control bit of outputting comparing result
                                    //CMPCR1.1 : 比较结果输出控制位
#define CMPRES 0x01                 //CMPCR1.0 : Flag bit of comparator result
                                    //CMPCR1.0 : 比较器比较结果标志位
sfr CMPCR2 = 0xE7;                  //Compare Control Register 2  比较器控制寄存器2
#define INVCMPO 0x80                /*CMPCR2.7 : Control bit for inversing comparator
                                    output //CMPCR2.7 : 比较结果反向输出控制位*/
#define DISFLT 0x40                 //CMPCR2.6 : Disable the 0.1μs Filter output by comparator
                                    //CMPCR2.6：比较器输出端0.1μs滤波控制位
#define LCDTY 0x3F                  /*CMPCR2.[5:0] : The timing control of interval jitter of
                                      comparator output //CMPCR2.[5:0] : 比较器输出的区
                                      抖时间控制*/
sbit LED0 = P1^0;                   // LED for positive edge interrupt request
                                    //上升沿中断请求指示灯
sbit LED1 = P1^1;                   // LED for negative edge interrupt request
                                    //下降沿中断请求指示灯
sbit LED2 = P1^2;                   // LED for direct output of comparator
                                    //比较器直接输出指示灯
sbit K = P1^3;                      // Switch of interrupt request mode
                                    //中断请求方式控制开关
void main()
{
    CMPCR1 = 0;                     // Initialize CMPCR1  初始化比较器控制寄存器1
    CMPCR2 = 0;                     // Initialize CMPCR2  初始化比较器控制寄存器2
    CMPCR1 &= ~PIS;                 /* Choose external pin P5.5 (CMP+) as the positive pole
                                    input source of comparator  选择外部引脚P5.5（CMP+）
                                    为比较器的正极输入源*/
```

```
CMPCR1 &= ~NIS;                 /* Choose internal BandGap voltage BGV as the negative
                                pole input source of comparator  选择内部BandGap电
                                压BGV为比较器的负极输入源*/
CMPCR1 &= ~CMPOE;               // Disable the result output of comparator
                                //  禁用比较器的比较结果输出
CMPCR2 &= ~DISFLT;              // Enable the 0.1μs Filter output by comparator
                                //  不禁用(使能)比较器输出端的0.1μs滤波电路
CMPCR2 &= ~LCDTY;               /* Output the result of comparator directly without
                                de-jitter  比较器结果不去抖动，直接输出*/
CMPCR1 |= CMPEN;                // Enable comparator  使能比较器
while (1)
{
    if(K==1)
    {
        CMPCR2 &= ~INVCMPO;     /* Normal output the comparing result of comparator
                                    on P1.2  比较器的比较结果正常输出到P1.2*/
        CMPCR1 |= PIE;          // Enable the positive edge interrupt of comparator
                                //  使能比较器的上升沿中断
        CMPCR1&=~ NIE;          // Disable the negative edge interrupt of comparator
                                //  关闭比较器的下降沿中断
    }
    else
    {
        CMPCR2 |= INVCMPO;      /* Reverse the result of comparator and then output
                                to P1.2  比较器的比较结果取反后输出到P1.2*/
        CMPCR1 |= NIE;          // Enable the negative edge interrupt of comparator
                                //  使能比较器的下降沿中断
        CMPCR1&=~ PIE;          // Disable the positive edge interrupt of comparator
                                //  关闭比较器的上升沿中断
    }
    if(CMPCR1 & CMPIF)
    {
        CMPCR1 &= ~CMPIF;       // Clear the finish flag  清除完成标志
        if(K==1)
        {
            LED0=0;             // Light up LED for positive edge interrupt request
                                //  点亮上升沿中断请求指示灯
            LED1=1;             //  关闭下降沿中断请求指示灯
        }
        Else
```

```
            {
                LED1=0;                 // Light up LED for negative edge interrupt request
                                        // 点亮下升沿中断请求指示灯
                LED0=1;                 // 关闭上升沿中断请求指示灯
            }


        }
    }
}
```

Summary 本章小结

A comparator is built in STC15W201S MCU. Internal signal and external signal can be configured as the in-phase or reversed-phase input source of the comparator. Comparator is equipped with de-jitter (filter) function. The output of comparator is not used for output signal or interrupt request signal directly, but de-jitter through a method of delay with adjustable delay time. The result of comparator can not only output in-phase or reversed-phase, but also detect the result state of comparator. There are two forms, positive edge interrupt and negative edge interrupt, for comparator interrupt.

Comparator of STC15W201S MCU replaces the functions of general comparator, such as temperature, humidity, pressure and other control comparator, and it is more reliable and powerful. Moreover, the comparator can simulate A/D transfer function.

Exercise 练习题

1. Completions

1) The comparator of STC15W201S MCU consists of _____, _____ and_____.

2) There are_____and_____ of 2 kinds of signal source in STC15W201S MCU comparator in-phase input (positive pole), which is selected by control bit_____ in comparator control register CMPCR1.

3) There are_____and_____ of 2 kinds of signal source in STC15W201S MCU comparator inverse input (negative pole), which is selected by control bit_____in comparator control register CMPCR1.

4) The interrupt request flag of STC15W201S MCU comparator is_____, interrupt vector address is_____, interrupt number is_____, and interrupt priority is_____.

2. Choice Questions

1) That signal sources of in-phase and inverse input of STC15W201S MCU comparator are_____ respectively whenCMPCR1=84H.

A. Select analog input channel, P5.4 B. Select analog input channel, BGV

C. P5.5, P5.4 D. P5.5, BGV

2) The interrupt request flag of STC15W201S MCU comparator will be set to 1 and send interrupt request to CPU when_____, if CMPCR1=90H.

A. Output signal of comparator goes to high from low.

B. Output signal of comparator goes to low from high.

C. Output signal of comparator goes to low from high, or high from low.

3) The acknowledging time of comparator is_____ system clocks after reset.

A. 6 B. 9 C. 16 D. 30

4) If CMPOE in CMPCR1 is 1, the compare result will be sent to_____.

A. P1.0 B. P1.1 C. P1.2 D. P1.3

3. True or False

1) There is a 0.1μs filter in STC15W201S MCU comparator, which is uncontrollable. ( )

2) Interrupt request flag of comparator will be set to 1 and interrupt request will be sent to CPU when output signal of STC15W201S MCU comparator goes to low from high. ( )

3) The result of STC15W201S MCU comparator is outputted via P1.2. ( )

4) The CMPRES in STC15W201S MCU comparator CMPCR1 is a compare result flag and read only. It won't make any sense when it's written by software. ( )

4. Question and Answer

1) How is the interrupt request flag of comparator generated?

2) How to inverse the result of comparator and through which pin will it be outputted?

3) What is the meaning of the control bit LCDTY[5:0] in comparator CMPCR2?

4) What is the meaning of the input signal source BGV of STC15W201S MCU comparator inverse input (negative pole)?

参 考 文 献

[1] 宏晶科技．STC15 系列单片机技术手册[Z]．2014
[2] 丁向荣．单片机原理与应用（C 语言版）．北京：电子工业出版社，2014.12
[3] 朱派龙．机械工程专业英语图解教程（第 2 版）．北京：北京大学出版社，2013.9