SinoMCU 8 位单片机

MC9938

用户手册

V1.0

目录

1 产品概要

1.1 产品特性

- 8 位 CPU 内核
  - 精简指令集，8 级深度硬件堆栈
  - CPU 双时钟，可在系统高/低频时钟之间切换
  - 高频时钟下 FCPU 可配置为 2T/4T/8T/16T/32T/64T，低频时钟下 FCPU 固定为 2T
- 程序存储器
  - 8K×16 位 FLASH 型程序存储器
  - 可通过间接寻址读取程序存储器内容
  - 支持在线编程，擦写次数至少 1000 次
- 数据存储器
  - 384 字节 SRAM 通用数据存储器，支持直接寻址、间接寻址等多种寻址方式
  - 64 字节 EEPROM 型数据存储器，擦写次数至少 10000 次
- 3 组共 22 个 I/O
  - P0（P00~P07），P1（P10~P17），P2（P20~P25）
  - P01/P02 复用成 SDA/SCL 时为开漏输出
  - P1、P2 为大电流端口；P1 可复用为键盘中断输入
  - 所有端口均内置上/下拉电阻，均可单独使能/禁用
- 时钟系统
  - 内置高频 RC 振荡器（16MHz），可用作系统高频时钟源
  - 内置低频 RC 振荡器（32KHz），可用作系统低频时钟源
- 多种系统工作模式
  - 高速运行模式：CPU 在高频时钟下运行
  - 低速运行模式：CPU 在低频时钟下运行
  - HOLD 模式 1：CPU 停止运行，高频时钟源工作
  - HOLD 模式 2：CPU 停止运行，高频时钟源停止工作，低频时钟源工作
  - 休眠模式：CPU 停止运行，所有时钟源停止工作
- 内部自振式看门狗计数器（WDT）
  - 溢出时间可配置：64ms/2048ms
  - 工作模式可配置：始终开启、始终关闭、低功耗模式下关闭
- 4 个定时器
  - 8 位定时器 T0，可实现外部计数、BUZ 功能、PWM 功能
  - 16 位定时器 T1，可实现外部计数、16 位 PWM 功能
  - 16 位定时器 T2，可作为 UART 的波特率发生器
  - 8 位定时器 T3
- 1 个 12 位高精度 ADC
  - 14 路外部通道：AN0~AN13；2 路内部通道：GND、VDD/4
  - 参考电压可选：VDD、内部参考电压 VIR（2V）
  - ADC 时钟：FHIRC 的 8/16/32/64 分频
  - 支持零点校准
- 内置 IIC 通讯接口

- ✧ 支持单主机模式
- ✧ 支持 7 位地址编码
- ✧ 通讯速率可选 100Kbps、400Kbps、800Kbps、1Mbps（实际速率受芯片及外围电路影响）

■ 增强型 UART 接口
- ✧ 波特率可选择为系统时钟分频或者定时器溢出频率
- ✧ 增强功能包括帧出错检测及自动地址识别
- ✧ 支持 8 位同步半双工、8 位/9 位异步全双工等 4 种工作方式

■ 中断
- ✧ 外部中断（INT0~INT3）：INT0~INT1 可选三种触发方式，INT2~INT3 为下降沿触发
- ✧ 定时器中断（T0~T3）：溢出产生中断
- ✧ ADC 中断
- ✧ 键盘中断：8 路端口共用 1 个中断源，并可分别使能或屏蔽
- ✧ IIC 通讯中断
- ✧ UART 通讯中断

■ 低电压复位 LVR：2.0V/2.4V/2.8V/3.6V

■ 工作电压
- ✧ V_{LVR28} ~ 5.5V @ Fcpu = 8MHz（F_{HIRC}/2）
- ✧ V_{LVR24} ~ 5.5V @ Fcpu = 4MHz（F_{HIRC}/4）
- ✧ V_{LVR24} ~ 5.5V @ Fcpu = 2MHz（F_{HIRC}/8）
- ✧ V_{LVR20} ~ 5.5V @ Fcpu = 1MHz（F_{HIRC}/16）

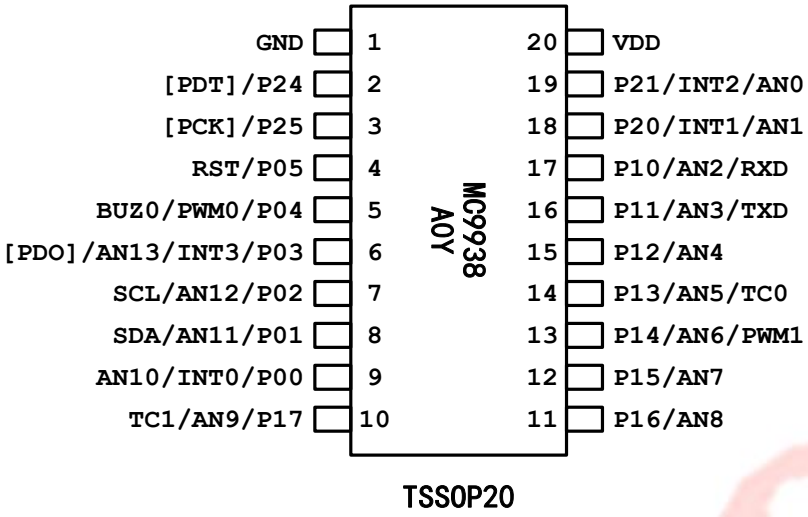
■ 封装形式
- ✧ TSSOP20/SOP16/SOP8

1.2 订购信息

产品名称	封装形式	备注
MC9938A0Y	TSSOP20	
MC9938A0K	SOP16	
MC9938A0H	SOP8	

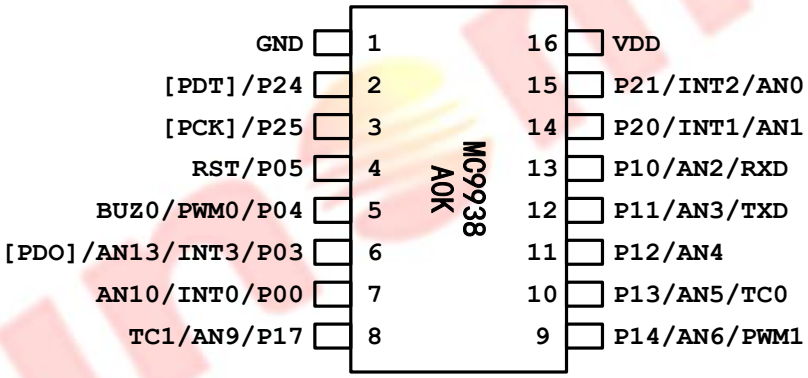
1.3 引脚排列

MC9938A0Y



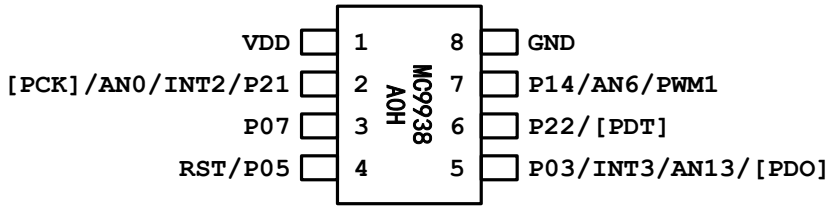
TSSOP20

MC9938A0K



SOP16

MC9938A0H



SOP8

1.4 端口说明

端口名称	类型	功能说明
VDD	P	电源
GND	P	地
P0，P1，P2	D	GPIO，内部上/下拉
INT0~INT3	DI	外部中断输入
TC0~TC1	DI	定时器 T0~T1 的外部计数输入
PWM0/BUZ0	DO	定时器 T0 的 PWM/BUZ 输出
PWM1	DO	定时器 T1 的 PWM 输出
AN0~AN13	AI	ADC 输入通道
SCL，SDA	D	IIC 通讯时钟/数据端口，开漏输出
RXD，TXD	D	UART 通讯接收/发送端口
RST	DI	外部复位输入
PCK，PDT，PDO	D	编程时钟/数据端口，开漏输出

注：P-电源，D-数字输入输出，DI-数字输入，DO-数字输出，A-模拟输入输出，AI-模拟输入，AO-模拟输出。

2 电气特性

2.1 极限参数

参数	符号	值	单位
电源电压	VDD	-0.3~6.0	V
输入电压	Vin	-0.3~VDD+0.3	V
工作温度	Ta	-40~85	℃
储存温度	Tstg	-65~150	℃
流入 VDD 最大电流	IVDDmax	100	mA
流出 GND 最大电流	IGNDmax	100	mA

注：若芯片工作条件超过极限值，将会造成永久性损坏；若芯片长时间工作在极限条件下，将会影响其可靠性。

2.2 直流电气特性

VDD=5V，T=25℃

特性	符号	端口	条件	最小	典型	最大	单位
工作电压	VDD	VDD	Fcpu=8MHz@HIRC 16MHz	VLVR28		5.5	V
			Fcpu=4MHz@HIRC 16MHz	VLVR24		5.5	
			Fcpu=2MHz@HIRC 16MHz	VLVR24		5.5	
			Fcpu=1MHz@HIRC 16MHz	VLVR20		5.5	
			Fcpu=16KHz@LIRC 32KHz	VLVR20		5.5	
输入漏电流	Ileak	所有输入脚	VDD=5V	-1		1	uA
输入高电平	Vih	所有输入脚		0.8VDD			V
输入低电平	Vil	所有输入脚				0.2VDD	V
输出拉电流	Ioh1	P0	Voh=0.9VDD		10		mA
	Ioh2	P1, P2	Voh=0.9VDD		20		mA
输出灌电流	Iol1	P0	Vol=0.1VDD		15		mA
	Iol2	P1, P2	Vol=0.1VDD		30		mA
上拉电阻	Rpu1	P0, P1, P2	VDD=5V, Vin=0		30		KΩ
	Rpu2	SDA. SCL (P01, P02)	VDD=5V, Vin=0, IICRS=0		4.7		KΩ
			VDD=5V, Vin=0, IICRS=1		1.3		KΩ
下拉电阻	Rpd	P0, P1, P2	Vin=VDD=5V		30		KΩ
动态功耗	Idd	VDD	VDD=5V, Fcpu=8MHz@HIRC		3.8		mA
			VDD=5V, Fcpu=4MHz@HIRC		2.5		mA
			VDD=5V, Fcpu=16KHz@LIRC		20		uA
HOLD1 功耗	Ihold1	VDD	VDD=5V, CPU 停, HIRC/LIRC 开		500		uA
HOLD2 功耗	Ihold2	VDD	VDD=5V, CPU 停, HIRC 关, LIRC 开		3	6	uA
休眠模式功耗	Istop	VDD	休眠模式, WDT 开, LVR 关		1	2	uA

低压复位功耗	I_{LVR}	VDD	VDD=5V, LVR 开		45	60	uA
低压复位电压	V_{LVR20}	VDD		-15%	2.0	+15%	V
	V_{LVR24}			-15%	2.4	+15%	V
	V_{LVR28}			-15%	2.8	+15%	V
	V_{LVR36}			-15%	3.6	+15%	V

注：功耗特性参数的条件说明中，诸如 HIRC/LIRC/WDT/LVR/LVD/ADC 等未注明的模块默认为关闭状态。

2.3 ADC 特性参数

VDD=5V，T=25℃

特性	符号	条件	最小	典型	最大	单位
ADC 有效工作电压	V_{ADC}	T=25℃, VDD=5V	2.6		5.5	V
积分线性误差	INL	V_{REF}=VDD, F_{ADC}=1MHz, Tcon=20us			±4	LSB
微分线性误差	DNL	V_{REF}=VDD, F_{ADC}=1MHz, Tcon=20us			±2	LSB
零点偏移误差	EZ	V_{REF}=VDD, F_{ADC}=1MHz, Tcon=20us			±4	LSB
增益误差	ET	V_{REF}=VDD, F_{ADC}=1MHz, Tcon=20us			±4	LSB
转换时钟	F_{ADC}	VDD=5V	0.25		2	MHz
转换时间	Tcon	F_{ADC}=1MHz	16		27	us
ADC 输入电压	V_{AIN}		GND		V_{REF}	V
ADC 输入阻抗	R_{AIN}		2			MΩ
ADC 输入电流	I_{AIN}				2	uA
ADC 动态电流	I_{ADD}	VDD=5V, AD 转换中		1	3	mA
ADC 静态电流	I_{ADS}	VDD=5V, ADEN=0		0.1	1	uA
模拟信号源推荐阻抗	Z_{AIN}				10	KΩ
ADC 参考电压	V_{REF}	选择 VDD		VDD		V
		选择内部参考 VIR, T=25℃	-1%	2.0	+1%	V
		选择内部参考 VIR, T=-40℃~85℃	-2%	2.0	+2%	V
内部参考供电电压	V_{PIR}	选择内部参考 VIR	V_{REF} +0.5		VDD	V

2.4 EEPROM 特性参数

特性	符号	条件	最小	典型	最大	单位
EEPROM 读电压	V_{EERD}	T=-40℃~85℃	1.8		5.5	V
EEPROM 写电压	V_{EEWR}	T=-40℃~85℃	2.6		5.5	V
EEPROM 字节写入时间	T_{EEWR}	T=25℃, VDD=5V			1	ms
		T=-40℃~85℃, VDD=2.6V~5.5V			20	ms
擦写次数(字节写)		T=25℃, VDD=5V	10000			cycle

2.5 交流电气特性

T=25℃

特性	符号	条件	最小	典型	最大	单位
HIRC 振荡频率	FHIRC	T=25℃, VDD=5V	-2%	16	+2%	MHz
		T=-40℃~85℃, VDD=2.2V~5.5V	-4%	16	+4%	MHz
LIRC 振荡频率	FLIRC	T=25℃, VDD=5V	-50%	32	+50%	KHz

3 CPU 及存储器

3.1 指令集

本芯片指令集为精简指令集。除程序跳转类指令，其余指令均为单周期指令，即执行时间为 1 个指令周期；所有指令均为单字指令，即指令码只占用 1 个程序存储器地址空间。

指令汇总表

助记符	说明	操作	周期	长度	标志
ADDAR R	寄存器 R 和 ACC 相加，结果存到 ACC	R+ACC→ACC	1	1	C,DC,Z
ADDRA R	寄存器 R 和 ACC 相加，结果存到 R	R+ACC→R	1	1	C,DC,Z
ADCAR R	寄存器 R 和 ACC 相加（带 C 标志），结果存到 ACC	R+ACC+C→ACC	1	1	C,DC,Z
ADCRA R	寄存器 R 和 ACC 相加（带 C 标志），结果存到 R	R+ACC+C→R	1	1	C,DC,Z
RSUBAR R	寄存器 R 和 ACC 相减，结果存到 ACC	R-ACC→ACC	1	1	C,DC,Z
RSUBRA R	寄存器 R 和 ACC 相减，结果存到 R	R-ACC→R	1	1	C,DC,Z
RSBCAR R	寄存器 R 和 ACC 相减（带 C 标志），结果存到 ACC	R-ACC-/C→ACC	1	1	C,DC,Z
RSBCRA R	寄存器 R 和 ACC 相减（带 C 标志），结果存到 R	R-ACC-/C→R	1	1	C,DC,Z
ASUBAR R	ACC 和寄存器 R 相减，结果存到 ACC	ACC-R→ACC	1	1	C,DC,Z
ASUBRA R	ACC 和寄存器 R 相减，结果存到 R	ACC-R→R	1	1	C,DC,Z
ASBCAR R	ACC 和寄存器 R 相减（带 C 标志），结果存到 ACC	ACC-R-/C→ACC	1	1	C,DC,Z
ASBCRA R	ACC 和寄存器 R 相减（带 C 标志），结果存到 R	ACC-R-/C→R	1	1	C,DC,Z
ANDAR R	寄存器 R 和 ACC 与操作，结果存到 ACC	R and ACC→ACC	1	1	Z
ANDRA R	寄存器 R 和 ACC 与操作，结果存到 R	R and ACC→R	1	1	Z
ORAR R	寄存器 R 和 ACC 或操作，结果存到 ACC	R or ACC→ACC	1	1	Z
ORRA R	寄存器 R 和 ACC 或操作，结果存到 R	R or ACC→R	1	1	Z
XORAR R	寄存器 R 和 ACC 异或操作，结果存到 ACC	R xor ACC→ACC	1	1	Z
XORRA R	寄存器 R 和 ACC 异或操作，结果存到 R	R xor ACC→R	1	1	Z
COMAR R	对 R 取反，结果存到 ACC	R 取反→ACC	1	1	Z
COMR R	对 R 取反，结果存到 R	R 取反→R	1	1	Z
CLRA	对 ACC 清零	0→ACC	1	1	Z
CLRR R	对 R 清零	0→R	1	1	Z
RLA	ACC 循环左移（带 C 标志）	ACC[7]→C ACC[6:0]→ACC[7:1] C→ACC[0]	1	1	C
RLAR R	寄存器 R 循环左移（带 C 标志），结果存到 ACC	R[7]→C R[6:0]→ACC[7:1] C→ACC[0]	1	1	C
RLR R	寄存器 R 循环左移（带 C 标志），结果存到 R	R[7]→C R[6:0]→R[7:1] C→R[0]	1	1	C

RRA	ACC 循环右移（带 C 标志）	C→ACC[7] ACC[7:1]→ACC[6:0] ACC[0]→C	1	1	C
RRAR R	寄存器 R 循环右移（带 C 标志），结果存到 ACC	C→ACC[7] R[7:1]→ACC[6:0] R[0]→C	1	1	C
RRR R	寄存器 R 循环右移（带 C 标志），结果存到 R	C→R[7] R[7:1]→R[6:0] R[0]→C	1	1	C
SWAPAR R	交换 R 的高低半字节，结果存到 ACC	R[7:4]→ACC[3:0] R[3:0]→ACC[7:4]	1	1	-
SWAPR R	交换 R 的高低半字节，结果存到 R	R[7:4]→R[3:0] R[3:0]→R[7:4]	1	1	-
MOVAR R	将 R 存到 ACC	R→ACC	1	1	Z
MOVR R	将 R 存到 R	R→R	1	1	Z
MOVRA R	将 ACC 存到 R	ACC→R	1	1	-
INCA	ACC 自加 1	ACC+1→ACC	1	1	-
INCAR R	R 加 1，结果存到 ACC	R+1→ACC	1	1	Z
INCR R	R 加 1，结果存到 R	R+1→R	1	1	Z
DECA	ACC 自减 1	ACC-1→ACC	1	1	-
DECAR R	R 减 1，结果存到 ACC	R-1→ACC	1	1	Z
DECR R	R 减 1，结果存到 R	R-1→R	1	1	Z
JZA	ACC 自加 1；结果为 0 则跳过下一条指令	ACC+1→ACC；结果为 0 则 PC+2→PC	1/2	1	-
JZAR R	R 加 1，结果存到 ACC；结果为 0 则跳过下一条指令	R+1→ACC；结果为 0 则 PC+2→PC	1/2	1	-
JZR R	R 加 1，结果存到 R；结果为 0 则跳过下一条指令	R+1→R；结果为 0 则 PC+2→PC	1/2	1	-
DJZA	ACC 自减 1；结果为 0 则跳过下一条指令	ACC-1→ACC；结果为 0 则 PC+2→PC	1/2	1	-
DJZAR R	R 减 1，结果存到 ACC；结果为 0 则跳过下一条指令	R-1→ACC；结果为 0 则 PC+2→PC	1/2	1	-
DJZR R	R 减 1，结果存到 R；结果为 0 则跳过下一条指令	R-1→R；结果为 0 则 PC+2→PC	1/2	1	-
BCLR R,b	对 R 的第 b 位清 0	0→R[b]	1	1	-
BSET R,b	对 R 的第 b 位置 1	1→R[b]	1	1	-
JBCLR R,b	若 R 的第 b 位为 0 则跳过下一条指令	若 R[b]=0，则 PC+2→PC	1/2	1	-
JBSET R,b	若 R 的第 b 位为 1 则跳过下一条指令	若 R[b]=1，则 PC+2→PC	1/2	1	-
ADDAI K	立即数 K 和 ACC 相加，结果存到 ACC	K+ACC→ACC	1	1	C,DC,Z
ADCAI K	立即数 K 和 ACC 相加（带 C 标志），结果存到 ACC	K+ACC+C→ACC	1	1	C,DC,Z
ISUBAI K	立即数 K 和 ACC 相减，结果存到 ACC	K-ACC→ACC	1	1	C,DC,Z
ISBCAI K	立即数 K 和 ACC 相减（带 C 标志），结果存到 ACC	K-ACC-/C→ACC	1	1	C,DC,Z
ASUBAI K	ACC 和立即数 K 相减，结果存到 ACC	ACC-K→ACC	1	1	C,DC,Z
ASBCAI K	ACC 和立即数 K 相减（带 C 标志），结果存到 ACC	ACC-K-/C→ACC	1	1	C,DC,Z
ANDAI K	立即数 K 和 ACC 与操作，结果存到 ACC	K and ACC→ACC	1	1	Z
ORAI K	立即数 K 和 ACC 或操作，结果存到 ACC	K or ACC→ACC	1	1	Z
XORAI K	立即数 K 和 ACC 异或操作，结果存到 ACC	K xor ACC→ACC	1	1	Z
MOVAI K	将立即数 K 存到 ACC	K→ACC	1	1	-
RETURN	从子程序返回	TOS→PC	2	1	-

RETAI K	从子程序返回，并将立即数 K 存到 ACC	TOS→PC K→ACC	2	1	-
RETIE	从中断返回	TOS→PC 1→GIE	2	1	-
CALL K	子程序调用	PC+1→TOS K→PC[12:0]	2	1	-
GOTO K	无条件跳转	K→PC[12:0]	2	1	-
NOP	空操作	空操作	1	1	-
DAA	BCD 码加法后，将 ACC 的值调整为 BCD 码	ACC(十六进制)→ACC(十进制)	1	1	C
DSA	BCD 码减法后，将 ACC 的值调整为 BCD 码	ACC(十六进制)→ACC(十进制)	1	1	-
CLRWDT	清看门狗定时器	0→WDT	1	1	TO,PD
STOP	进入低功耗模式	0→WDT；进入低功耗模式	1	1	TO,PD

注：对于条件跳转类指令，若跳转条件成立，则指令需 2 个周期，否则只需 1 个周期。

3.2 程序存储器

本芯片的程序存储器为 FLASH 型存储器，8K×16 位的地址为 0000H~1FFFH。
程序存储器地址分配如下图所示：

程序存储器可通过 INDF3 间接访问，如下例所示：通过 INDF3 访问程序存储器地址 0155H 中的内容，高 8 位存放在数据寄存器区 11H，低 8 位存放在数据寄存器区 10H

```
MOVAI   55H
MOVRA   FSR0            ；将 55H 写入 FSR0
MOVAI   01H
MOVRA   FSR1            ；将 01H 写入 FSR1
MOVAR   INDF3           ；读取 FSR1×256+FSR0 指向的程序存储器地址（0155H）
                        ；中的内容，高 8 位存入 HIBYTE，低 8 位存入 A 寄存器
MOVRA   10H             ；低 8 位放到数据寄存器 10H 地址
MOVAR   HIBYTE          ；从 HIBYTE 读取高 8 位
MOVRA   11H             ；高 8 位放到数据寄存器 11H 地址
```

3.3 数据存储器

数据存储器包括通用数据寄存器 GPR 和特殊功能寄存器 SFR，具体地址分配参照下表。GPR0 可直接寻址或通过 INDF0/INDF2 间接寻址，GPR1 和 SFR 可直接寻址或通过 INDF1/INDF2 间接寻址。

数据存储器还包括 EEPROM 存储器，需通过特殊寄存器 SFR 进行读写操作。

数据存储器区地址映射表

地址	类型	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
000H-0FFH	GPR0	通用数据存储器区 0							
100H-17FH	GPR1	通用数据存储器区 1							
180H-187H	SFR	INDF0	INDF1	INDF2	HIBYTE	FSR0	FSR1	PCL	PFLAG
188H-18FH		MCR	INDF3	INTE	INTF	OSCM	INTE1	INTF1	KBCR
190H-197H		IOP0	OEP0	PUP0	PDP0	IOP1	OEP1	PUP1	PDP1
198H-19FH		IOP2	OEP2	PUP2	PDP2				
1A0H-1A7H		T0CR	T0CNT	T0LOAD	T0DATA				
1A8H-1AFH		T1CR	T1CNTH	T1CNTL	T1LOADH	T1LOADL	T1DATAH	T1DATAL	
1B0H-1B7H		IICCR	IICSR	IICDR		EECR	EEMASK	EEAR	EEDR
1B8H-1BFH		ADCR0	ADCR1	ADRH	ADRL	ADIOS0	ADIOS1	OSADJCR	
1C0H-1C7H		T2CR	T2CNTH	T2CNTL	T2LOADH	T2LOADL	T3CR	T3CNT	T3LOAD
1C8H-1CFH		SCON	SBUF	SADDR	SADEN				
1D0H-1FFH	保留								

注：上表中灰色部分地址为系统保留区，读出数据不确定，写入操作可能会影响芯片正常工作。

数据存储器地址组成

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	寻址方式
0	0	0	0	0	0	0	来自指令的 9 位地址									直接寻址模式
0	0	0	0	0	0	0	0	FSR0								间接寻址模式 0
0	0	0	0	0	0	0	1	FSR1								间接寻址模式 1
FSR1								FSR0								间接寻址模式 2

直接寻址模式：以指令的低 9 位作为数据存储器地址。例：通过直接寻址模式把 55H 数据写入 10H 地址

```
MOVAI    55H
MOVRA    10H          ：把数据 55H 写入 10H 地址数据存储器中
```

间接寻址模式 0：当访问 INDF0 时，FSR0 作为数据存储器地址。例：通过 INDF0 把 55H 数据写入 10H 地址

```
MOVAI    10H
MOVRA    FSR0
MOVAI    55H
MOVRA    INDF0        ：把数据 55H 写入 FSR0 指向的数据存储器中
```

间接寻址模式 1：当访问 INDF1 时，FSR1 作为数据存储器地址。例：通过间接寻址模式 1 把 55H 数据写入 110H 地址

```
MOVAI    10H
MOVRA    FSR1
MOVAI    55H
MOVRA    INDF1        ：把数据 55H 写入 FSR1 指向的数据存储器中
```

间接寻址模式 2：当访问 INDF2 时，FSR1×256+FSR0 作为数据存储器地址。例：通过间接寻址模式 2 把 55H 数据写入 0110H 地址数据存储器

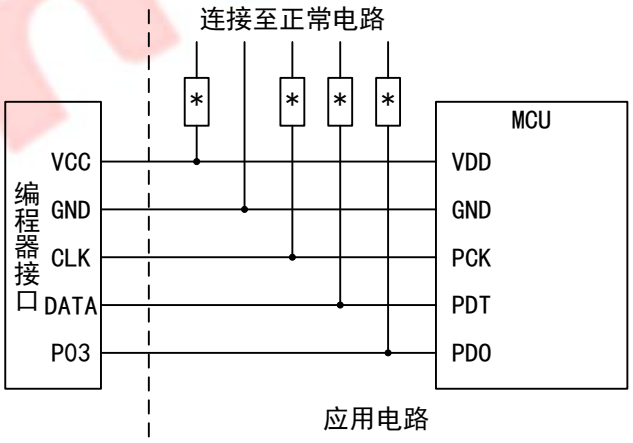
```
MOVAI    10H
MOVRA    FSR0
MOVAI    01H
MOVRA    FSR1
MOVAI    55H
MOVRA    INDF2        ：把数据 55H 写入 FSR1×256+FSR0 指向的数据存储器中
```

3.4 在线编程

本芯片支持在线编程，即可在最终应用电路中通过芯片的串行编程接口将用户程序代码烧录进程序存储器中。在线编程功能，可让用户先采用未编程的空芯片制造电路板而仅在产品交付前才将程序代码烧录进芯片，也方便用户直接在电路板上升级芯片中的程序代码。

本芯片的在线编程通过引脚 VDD、GND、PDT、PDO、PCK 实现，这些编程引脚的外围电路需进行针对性设计，以保证外围电路不会影响在线编程时端口上的电压/电流/时序等特性。

下图是典型的在线编程连接图：



* 隔离电路（根据需要而定）

3.5 堆栈

8 级堆栈深度，当程序响应中断或执行子程序调用指令时 CPU 会将 PC 自动压栈；当执行中断返回指令或子程序返回指令时，栈顶数据赋予 PC。

3.6 控制寄存器

间接寻址寄存器 0

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INDF0	INDF07	INDF06	INDF05	INDF04	INDF03	INDF02	INDF01	INDF00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **INDF0[7:0]** – 间接寻址寄存器 0

INDF0：INDF0 不是物理寄存器，对 INDF0 寻址实际上是对 FSR0 指向的数据存储器地址进行访问，从而实现间接寻址模式。

间接寻址寄存器 1

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INDF1	INDF17	INDF16	INDF15	INDF14	INDF13	INDF12	INDF11	INDF10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **INDF1[7:0]** – 间接寻址寄存器 1

INDF1：INDF1 不是物理寄存器，对 INDF1 寻址实际上是对 FSR1+256 指向的数据存储器地址进行访问，从而实现间接寻址模式。

间接寻址寄存器 2

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INDF2	INDF27	INDF26	INDF25	INDF24	INDF23	INDF22	INDF21	INDF20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **INDF2[7:0]** – 间接寻址寄存器 2

INDF2：INDF2 不是物理寄存器，对 INDF2 寻址实际上是对 FSR1×256+FSR0 指向的数据存储器地址进行访问，从而实现间接寻址模式。

间接寻址寄存器 3

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INDF3	INDF37	INDF36	INDF35	INDF34	INDF33	INDF32	INDF31	INDF30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **INDF3[7:0]** – 间接寻址寄存器 3

INDF3：INDF3 不是物理寄存器，对 INDF3 寻址实际上是对 FSR1×256+FSR0 指向的程序存储器地址进行访问，从而实现间接寻址模式。

注：对 INDF3 仅可使用读取指令（MOVAR INDF3）进行读取访问，读取内容高 8 位存放在 HIBYTE，低 8 位存放在 A 寄存器。

字操作高 8 位缓存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
HIBYTE	HIBYTE7	HIBYTE6	HIBYTE5	HIBYTE4	HIBYTE3	HIBYTE2	HIBYTE1	HIBYTE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **HIBYTE[7:0]** – 字操作高字节缓冲器

HIBYTE：用于读取 INDF3 时存放 FSR1×256+FSR0 指向的程序存储器内容高 8 位数据。

数据指针寄存器 0

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSR0	FSR07	FSR06	FSR05	FSR04	FSR03	FSR02	FSR01	FSR00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **FSR0[7:0]** – 数据指针寄存器 0

FSR0：间接寻址模式 0 的指针，或间接寻址模式 2、3 的指针低 8 位。

数据指针寄存器 1

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSR1	FSR17	FSR16	FSR15	FSR14	FSR13	FSR12	FSR11	FSR10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **FSR1[7:0]** – 数据指针寄存器 1

FSR1：间接寻址模式 1 的指针，或间接寻址模式 2、3 的指针高 8 位。

程序指针计数器低字节

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCL	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **PC[7:0]** – 程序指针计数器低 8 位

程序指针计数器（PC）有以下几种操作模式：

- ✧ 顺序运行指令：PC = PC + 1；
- ✧ 分支指令 GOTO/CALL：PC = 指令码低 13 位；
- ✧ 子程序返回指令 RETIE/RETURN/RETAI：PC = 堆栈栈顶（TOS）；

对 PCL 操作指令：

- ✧ 对 PCL 操作的加法指令：PC = (PC[15:0]+ALU[7:0])；
- ✧ 对 PCL 操作的其它指令：PC = {PC[15:8], ALU[7:0](ALU 运算结果)}；

CPU 状态寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PFLAG	-	-	-	-	-	Z	DC	C
R/W	-	-	-	-	-	R/W	R/W	R/W
初始值	-	-	-	-	-	X	X	X

BIT[2] **Z** – 零标志位
0： 算术或逻辑运算的结果不为零；
1： 算术或逻辑运算的结果为零；

BIT[1] **DC** – 半字节进/借位标志位
0： 加法运算时半字节无进位；减法运算时半字节有借位；
1： 加法运算时半字节有进位；减法运算时半字节无借位；

BIT[0] **C** – 进/借位标志位
0： 加法运算时无进位；减法运算时有借位；移位后移出逻辑 0；
1： 加法运算时有进位；减法运算时无借位；移位后移出逻辑 1；

杂用寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MCR	GIE	-	TO	PD	MINT11	MINT10	MINT01	MINT00
R/W	R/W	-	R	R	R/W	R/W	R/W	R/W
初始值	0	-	0	0	0	0	0	0

BIT[7] **GIE** – 总中断使能位
0： 屏蔽所有中断；
1： 中断源是否产生中断由相应的中断控制位决定；

BIT[5] **TO** – 看门狗溢出标志位
0： 上电复位；执行 CLRWDT 或 STOP 指令；
1： 发生 WDT 溢出；

BIT[4] **PD** – 进入低功耗模式标志位
0： 上电复位；执行 CLRWDT 指令；
1： 执行 STOP 指令；

BIT[3:2] **MINT1[1:0]** – 外部中断 INT1 触发方式选择位

MINT1[1:0]	INT1 触发方式
00	上升沿触发
01	下降沿触发
1X	电平变化触发

BIT[1:0] **MINT0[1:0]** – 外部中断 INT0 触发方式选择位

MINT0[1:0]	INT0 触发方式
00	上升沿触发

01	下降沿触发
1X	电平变化触发

3.7 用户配置字

芯片为保证系统正常工作，会将关键模块的配置信息预先存储于单独的存储器区域中，在上电或其他复位发生后将配置信息载入寄存器中，通过寄存器配置关键模块的工作状态。该部分存储器中用户可选的内容即为用户配置字，可在烧录用户程序代码时进行配置与烧录。

本芯片的用户配置字，定义如下：

符号	功能说明
FCPUS	高速模式 CPU 速度选择： 指令周期为 2 个高频时钟周期，FCPU=FHOSC/2； 指令周期为 4 个高频时钟周期，FCPU=FHOSC/4； 指令周期为 8 个高频时钟周期，FCPU=FHOSC/8； 指令周期为 16 个高频时钟周期，FCPU=FHOSC/16； 指令周期为 32 个高频时钟周期，FCPU=FHOSC/32； 指令周期为 64 个高频时钟周期，FCPU=FHOSC/64； 指令周期为 128 个高频时钟周期，FCPU=FHOSC/128； 指令周期为 256 个高频时钟周期，FCPU=FHOSC/256；
RSTEN	外部复位使能控制： 不使能外部复位；　　使能外部复位；
WDTM	WDT 工作模式选择： 始终开启； 始终关闭； 低功耗模式下关闭；
WDTT	WDT 溢出时间选择： 上电延时=WDT 溢出时间=64ms； 上电延时=64ms，WDT 溢出时间=2048ms；
VLVRS	LVR 电压选择： 2.0V；2.4V；2.8V；3.6V；
LVRSLP	低功耗模式 LVR 控制： 低功耗模式关闭 LVR；　　低功耗模式不关闭 LVR；
ENCR	代码加密选择： 使能代码加密；　　不使能代码加密；

4 系统时钟

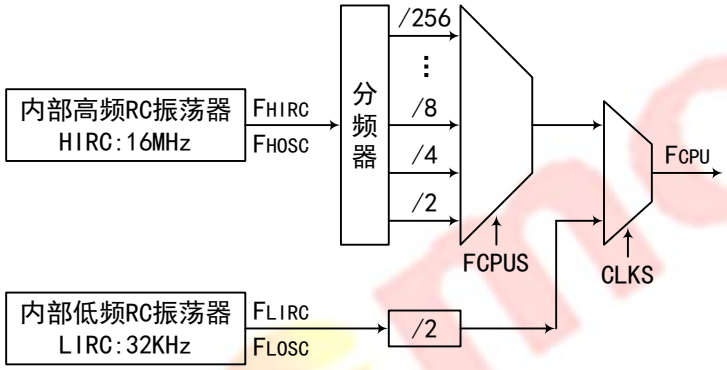
本芯片为双时钟系统，内部电路均在系统高频时钟 FHOSC 或系统低频时钟 FLOSC 下工作，部分模块还可在 FHOSC 和 FLOSC 之间切换。

本芯片系统高频时钟源固定为内部 16MHz 高频 RC 振荡器；系统低频时钟源固定为内部 32KHz 低频 RC 振荡器。

CPU 的时钟源可在系统高频时钟 FHOSC 和系统低频时钟 FLOSC 之间切换。选择 FHOSC 时，CPU 时钟 FCPU 通过 FCPUS 配置；选择 FLOSC 时，FCPU 固定为 FLOSC 的 2 分频。

WDT（看门狗）电路的时钟源固定为内部低频 RC 振荡器。

系统时钟示意图



4.1 内置高频 RC 振荡器

本芯片内置一个高精度 16MHz 的 HIRC 振荡器，该振荡器可用作系统高频时钟源。

4.2 内置低频 RC 振荡器

本芯片内置一个典型值为 32KHz 的 LIRC 振荡器，该振荡器可用作系统低频时钟源，同时用于上电延时定时器、WDT 定时器等电路。

4.3 工作模式

本芯片支持高速运行、低速运行、HOLD 模式 1、HOLD 模式 2 和休眠模式等多种系统工作模式。

工作模式	切换条件	系统状态
高速运行	1、复位 2、低速运行模式下，CLKS 清0 3、HOLD 模式1/HOLD 模式2/休眠模式下，唤醒	CPU 高速运行，高/低频时钟源均工作
低速运行	1、高速运行模式下，CLKS 置1 2、HOLD 模式1/HOLD 模式2/休眠模式下，唤醒	CPU 低速运行，高频时钟源状态由 HFEN 决定
HOLD1	高/低速运行模式下，HFEN 置1，执行 STOP	CPU 暂停，高频时钟源工作，低频时钟源状态由 LFEN 决定
HOLD2	高/低速运行模式下，HFEN 清0，LFEN 置1，执行 STOP	CPU 暂停，高频时钟源停止，低频时钟源工作
休眠	高/低速运行模式下，HFEN 清0，LFEN 清0，执行 STOP	CPU 暂停，高/低频时钟源均停止

注：WDT 时钟源为 LIRC，WDT 工作时 LIRC 将不受工作模式影响。

工作模式寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSCM	-	-	STBL	STBH	-	CLKS	LFEN	HFEN
R/W	-	-	R	R	-	R/W	R/W	R/W
初始值	-	-	X	1	-	0	0	0

BIT[5] **STBL** – 低频时钟源稳定标志位
0： 低频时钟源停振或未稳定；
1： 低频时钟源已稳定运行；

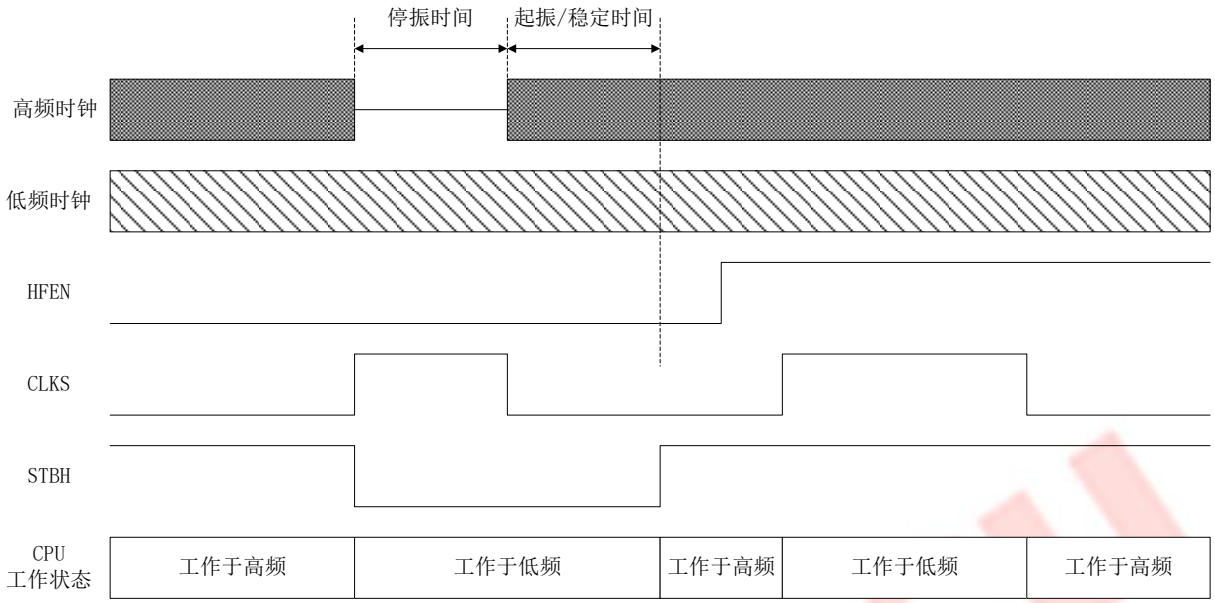
BIT[4] **STBH** – 高频时钟源稳定标志位
0： 高频时钟源停振或未稳定；
1： 高频时钟源已稳定运行；

BIT[2] **CLKS** – CPU 时钟源选择位
0： 系统高频时钟作为 CPU 时钟源；
1： 系统低频时钟作为 CPU 时钟源；

BIT[1] **LFEN** – 低频时钟源使能位
0： 在休眠/HOLD 模式下，低频时钟源停止工作；
1： 低频时钟源始终工作；

BIT[0] **HFEN** – 高频时钟源使能位
0： 在低速/休眠/HOLD 模式下，高频时钟源停止工作；
1： 高频时钟源始终工作；

高低频时钟切换时序图



4.4 低功耗模式

本芯片的低功耗模式包括休眠模式、HOLD 模式 1、HOLD 模式 2。

STOP 指令可使系统进入低功耗模式，同时对系统会产生以下影响：

- ✧ CPU 停止运行；
- ✧ 根据不同模式停止相应时钟源的振荡；
- ✧ RAM 内容保持不变；
- ✧ 所有的输入输出端口保持原态不变；
- ✧ 定时器若其时钟源未停止，则可以保持继续工作；

以下情况可使系统退出低功耗模式：

- ✧ 有外部中断请求发生（若有外部中断功能）；
- ✧ 定时器溢出中断发生（若低功耗模式下定时器保持继续工作）；
- ✧ 有键盘中断请求发生（若有键盘中断功能）；
- ✧ 键盘扫描端口有电平变化发生（若有键盘扫描功能）；
- ✧ 有 WDT 溢出（若低功耗模式下 WDT 保持继续工作）；
- ✧ 外部复位（若有外部复位功能）；
- ✧ 上电复位；

注：低功耗模式下产生中断请求时，若相应中断使能位关闭，则不会退出低功耗模式；若仅中断使能位打开而总中断使能位关闭，则仅唤醒 CPU 执行下一条指令；若中断使能位和总中断使能位均打开，则唤醒 CPU 后执行中断服务程序。

5 复位

5.1 复位条件

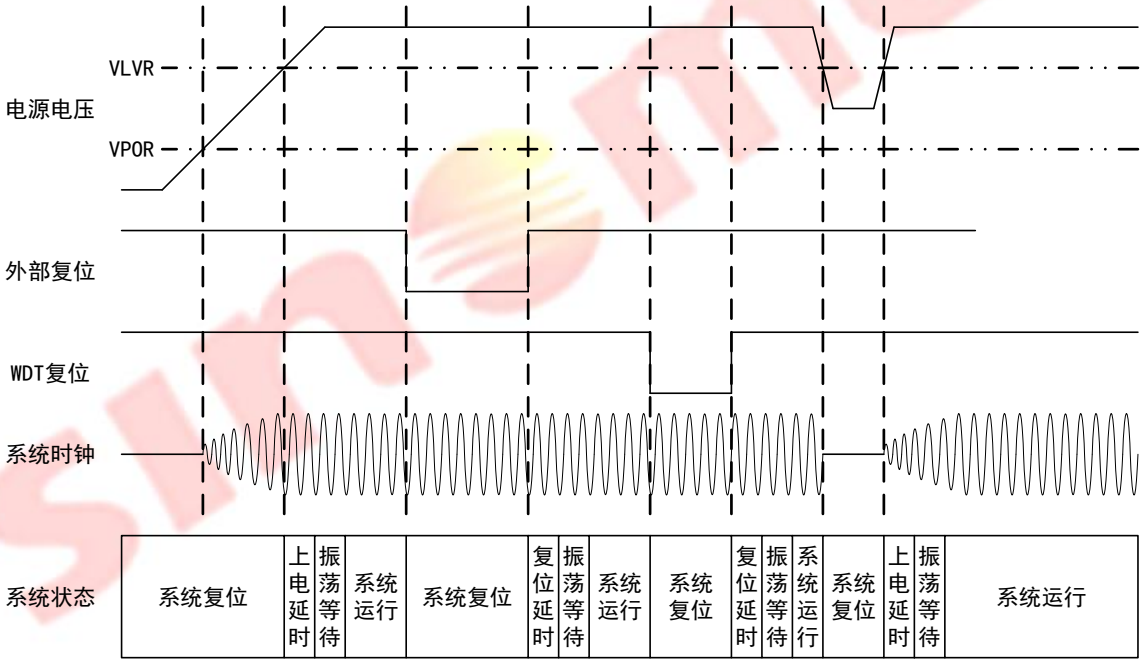
本芯片有如下几种复位方式：

- ✧ 上电复位 POR；
- ✧ 外部复位；
- ✧ 低电压复位 LVR；
- ✧ WDT 看门狗复位；

任何一种复位发生时，系统将会重新从 0000H 地址处开始执行指令；另外系统还会将所有特殊功能寄存器 SFR 重置为默认初始值。

上电复位和 LVR 复位会关闭系统主时钟振荡器，复位解除后才重新打开振荡器，由于振荡器起振和稳定需要一定的时间，所以系统会保持一定时间的上电延时和振荡等待后才开始工作；外部复位和 WDT 复位不会关闭系统主时钟振荡器，复位解除时系统会在较短的复位延时和振荡等待后即开始工作。

下图是复位产生和系统工作状态之间的关系示意图：



5.2 上电复位

本芯片的上电复位电路可以适应快速、慢速上电的情况，并且当芯片上电过程中出现电源电压抖动时都能保证系统可靠的复位。

上电复位过程可以概括为以下几个步骤：

(1) 检测系统工作电压，等待电压高于 V_{POR} 并保持稳定；

(2) 若有 LVR 功能，则需等待电压高于 V_{LVR} 并保持稳定；
(3) 若有外部复位功能，则需等待复位引脚电压高于 Vih；
(4) 初始化所有寄存器；
(5) 开启主时钟振荡器，并等待一段时间以待振荡器稳定；
(6) 上电结束，系统开始执行指令。

5.3 外部复位

外部复位功能是否开启可以通过用户配置字位配置，选择外部复位功能后复位引脚的内部上拉电阻自动有效。外部复位引脚 RST 是施密特结构的，低电平有效。当外部复位引脚为高电平时，系统正常运行；为低电平时，系统产生复位。

5.4 低电压复位

本芯片的 LVR 电压通过 VLVRS 位进行配置，详见用户配置字。电压检测电路有一定的回滞特性，通常回滞电压为 0.1V 左右，当电源电压下降到 LVR 电压时 LVR 复位有效，而电压需要上升到 LVR 电压+0.1V 时 LVR 复位才会解除。

5.5 看门狗复位

看门狗（WDT）复位是一种对程序正常运行的保护机制。正常情况下，用户软件需要按时对 WDT 定时器进行清零操作，保证 WDT 不溢出。若出现异常状况，程序未按时对 WDT 定时器清零，WDT 会溢出从而产生看门狗复位，系统重新初始化，返回受控状态。

注：在低功耗模式下，CPU 停止工作，若此时有 WDT 溢出，则仅唤醒 CPU，而不产生复位。

6 I/O 端口

6.1 I/O 工作模式

芯片共有两组 8 位端口 P0、P1 和一组 6 位端口 P2。除用作通用输入/输出端口外，部分端口还可复用为 ADC 模拟输入端口、外部中断输入端口、PWM/BUZ 输出端口等工作模式。

端口数据寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOP0	P07D	P06D	P05D	P04D	P03D	P02D	P01D	P00D
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **P0nD** – P0 口数据位（n=7-0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOP1	P17D	P16D	P15D	P14D	P13D	P12D	P11D	P10D
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **P1nD** – P1 口数据位（n=7-0）

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOP2	-	-	P25D	P24D	P23D	P22D	P21D	P20D
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	X	X	X	X	X	X

BIT[5:0] **P2nD** – P2 口数据位（n=5-0）

端口方向寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OEP0	P07OE	P06OE	P05OE	P04OE	P03OE	P02OE	P01OE	P00OE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P0nOE** – P0 口输出使能位（n=7-0）

0： 端口作为输入口，读端口操作将读取端口状态；

1： 端口作为输出口，读端口操作将读取数据寄存器值；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OEP1	P17OE	P16OE	P15OE	P14OE	P13OE	P12OE	P11OE	P10OE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P1nOE** – P1 口输出使能位（n=7-0）

0： 端口作为输入口，读端口操作将读取端口状态；
1： 端口作为输出口，读端口操作将读取数据寄存器值；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OEP2	-	-	P25OE	P24OE	P23OE	P22OE	P21OE	P20OE
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	0	0	0	0	0	0

BIT[5:0] **P2nOE** – P2 口输出使能位（n=5-0）
0： 端口作为输入口，读端口操作将读取端口状态；
1： 端口作为输出口，读端口操作将读取数据寄存器值；

6.2 上/下拉电阻控制

所有端口都有内部上/下拉电阻，均有独立的上/下拉电阻寄存器控制位，控制其上/下拉电阻在端口作为输入状态时是否有效，端口处于输出状态时，上/下拉电阻控制位无效。

上拉电阻控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PUP0	P07PU	P06PU	P05PU	P04PU	P03PU	P02PU	P01PU	P00PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P0nPU** – P0 口上拉电阻控制位（n=7-0）
0： 端口内部上拉电阻无效；
1： 端口内部上拉电阻有效；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PUP1	P17PU	P16PU	P15PU	P14PU	P13PU	P12PU	P11PU	P10PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P1nPU** – P1 口上拉电阻控制位（n=7-0）
0： 端口内部上拉电阻无效；
1： 端口内部上拉电阻有效；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PUP2	-	-	P25PU	P24PU	P23PU	P22PU	P21PU	P20PU
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	0	0	0	0	0	0

BIT[5:0] **P2nPU** – P2 口上拉电阻控制位（n=5-0）
0： 端口内部上拉电阻无效；
1： 端口内部上拉电阻有效；

下拉电阻控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDP0	P07PD	P06PD	P05PD	P04PD	P03PD	P02PD	P01PD	P00PD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P0nPD** – P0 口下拉电阻控制位（n=7-0）

0： 端口内部下拉电阻无效；

1： 端口内部下拉电阻有效；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDP1	P17PD	P16PD	P15PD	P14PD	P13PD	P12PD	P11PD	P10PD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P1nPD** – P1 口下拉电阻控制位（n=7-0）

0： 端口内部下拉电阻无效；

1： 端口内部下拉电阻有效；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDP2	-	-	P25PD	P24PD	P23PD	P22PD	P21PD	P20PD
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	0	0	0	0	0	0

BIT[5:0] **P2nPD** – P2 口下拉电阻控制位（n=5-0）

0： 端口内部下拉电阻无效；

1： 端口内部下拉电阻有效；

6.3 端口模式控制

部分端口可以作为数字 I/O 端口，也可复用为模拟信号 I/O 端口，ADIOS 寄存器可以设置这些端口的工作模式。当设置为数字 I/O 时，端口的模拟 I/O 功能被屏蔽；设置为模拟 I/O 模式时，则端口的数字 I/O 功能被屏蔽。

模拟端口使能寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADIOS0	AN7EN	AN6EN	AN5EN	AN4EN	AN3EN	AN2EN	AN1EN	AN0EN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **ANnEN** – 模拟输入通道 ANn 使能位（n=7-0）

0： 端口用作数字 I/O；

1： 端口用作模拟 I/O；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADIOS1	-	-	AN13EN	AN12EN	AN11EN	AN10EN	AN9EN	AN8EN
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	0	0	0	0	0	0

BIT[5:0] **ANnEN** – 模拟输入通道 ANn 使能位（n=13-8）

0： 端口用作数字 I/O；

1： 端口用作模拟 I/O；

7 定时器 TIMER

7.1 看门狗定时器 WDT

看门狗定时器 WDT 的时钟源为内部低频 RC 振荡器，可由用户配置字的 WDTM 位设置工作模式。

若选择始终开启，则在休眠/HOLD 模式下 WDT 依然运行，WDT 溢出时将唤醒 CPU，CPU 恢复运行；若在 CPU 运行时产生 WDT 溢出，WDT 溢出时将复位芯片。

若选择低功耗模式下关闭，则在低功耗模式下 WDT 被硬件自动关闭，在其他方式唤醒休眠模式后 WDT 自动继续运行。

执行 WDT 清 0 指令可使 WDT 计数器重新开始计数。

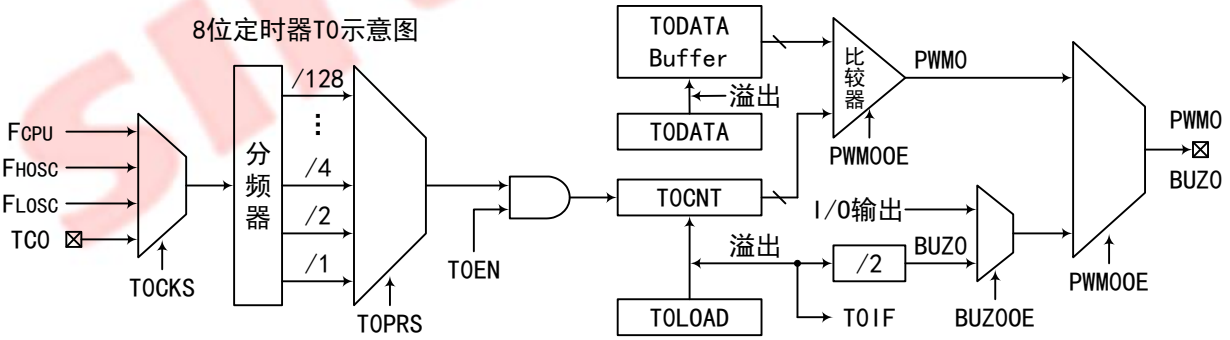
WDT 溢出时间可配置为 64ms/2048ms。

注：WDT 溢出时间为典型值，实际值偏差大，必须保证清 WDT 时间小于典型值的 1/4。

7.2 定时器 T0

定时器 T0 为 8 位定时/计数器，包含 1 个 8 位递减计数器、可编程预分频器、控制寄存器、8 位重载寄存器及比较寄存器。

- ✧ 可通过预分频比设置计数频率，可通过重载寄存器控制计数周期；
- ✧ 支持 PWM 输出，可通过比较寄存器设置 PWM 占空比；
- ✧ 支持 BUZ 输出；
- ✧ 支持溢出中断和溢出唤醒功能；



注：

1. ***F_{HOSC} 指系统高频时钟，F_{LOSC} 指系统低频时钟；***
2. ***当定时器选择高频时钟 F_{HOSC} 且工作模式寄存器 OSCM 的 HFEN=1 时，定时器在低速运行或休眠模式下可继续工作，溢出中断可唤醒休眠模式；若 HFEN=0，则定时器在低速或休眠模式下将停止工作；***
3. ***当定时器选择低频时钟 F_{LOSC} 且工作模式寄存器 OSCM 的 LFEN=1 时，定时器在休眠模式下可继续工作，溢出中断可唤醒休眠模式；若 LFEN=0，则定时器在休眠模式下将停止工作；***
4. ***使能或关闭 PWM、BUZ 输出时，应注意当前的端口状态及操作时机，以免输出尖峰脉冲；***
5. ***定时器重载寄存器的值禁止为 0，否则定时器将无法正常工作；***

定时器 T0 控制寄存器 T0CR 中，T0CKS 位可选择 T0 的时钟源，T0PRS 可选择 T0 的预分频比，所选中的时钟源通过预分频器后产生 T0 计数器 T0CNT 的计数时钟。时钟分频比可选择 1~128 分频，对 T0CNT 的写操作将使预分频器清零，分频比保持不变。

当 T0EN=0 时，T0CNT 保持不变，写重载寄存器 T0LOAD 将自动加载到 T0CNT 中；当 T0EN=1 时，T0CNT 递减计数，此时写 T0LOAD 不会立即加载，T0CNT 计数到 0 时产生 T0 溢出中断，标志位 T0IF 置 1，下一个时钟 T0LOAD 值自动载入 T0CNT 重新开始计数。

定时器 T0 可实现 BUZ 功能，当 BUZ0OE=1 且 PWM0OE=0 时，BUZ0 输出蜂鸣器驱动信号，频率为 T0 溢出频率的 2 分频。

定时器 T0 可实现 PWM 功能，PWM0OE 置 1 将使能 PWM 功能且端口 PWM0 将输出 PWM 波形。每个 PWM 周期内，计数器 T0CNT 从重载值开始递减计数：当计数到与比较寄存器 T0DATA 相等时，PWM0 信号变为高电平；当计数溢出时，PWM0 信号变为低电平。T0DATA 配有 1 个 8 位缓冲器用于与 T0CNT 比较，PWM0OE=0 时写 T0DATA 将立即加载到缓冲器中，而 PWM0OE=1 时写 T0DATA 则将在 T0 溢出时才载入缓冲器中。若要首个 PWM 周期准确，需先写重载寄存器和比较寄存器（设置周期和占空比），再开启定时器和 PWM 功能。

PWM0 信号的占空比计算如下：

- ✧ PWM0 高电平时间 =（T0DATA）× T0CNT 计数时钟周期
- ✧ PWM0 周期（T0 的溢出周期）=（T0LOAD+1）×T0CNT 计数时钟周期
- ✧ PWM0 占空比 =（T0DATA）/（T0LOAD+1）

定时器 T0 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0CR	T0EN	PWM0OE	BUZ0OE	T0CKS1	T0CKS0	T0PRS2	T0PRS1	T0PRS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] **T0EN** – T0 使能控制位

0： 关闭 T0；

1： 开启 T0；

BIT[6] **PWM0OE** – PWM0 功能控制位

0： 关闭 PWM0 功能，并禁止端口输出 PWM 波形；

1： 使能 PWM0 功能，并允许端口输出 PWM 波形；

BIT[5] **BUZ0OE** – BUZ0 输出使能位

0： 关闭 BUZ0 输出；

1： 使能 BUZ0 输出（仅 PWM0OE=0 时有效）；

BIT[4:3] **T0CKS[1:0]** – T0 时钟源选择位

T0CKS[1:0]	T0 时钟源
00	FCPU
01	FHOSC

10	FLOSC
11	TC0 上升沿

BIT[2:0] **T0PRS[2:0]** – T0 预分频比选择位

T0PRS[2:0]	T0 时钟预分频比
000	1:1
001	1:2
010	1:4
011	1:8
100	1:16
101	1:32
110	1:64
111	1:128

定时器 T0 计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0CNT	T0CNT7	T0CNT6	T0CNT5	T0CNT4	T0CNT3	T0CNT2	T0CNT1	T0CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T0CNT[7:0]** – T0 计数器，为可读写的递减计数器

定时器 T0 重载寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0LOAD	T0LOAD7	T0LOAD6	T0LOAD5	T0LOAD4	T0LOAD3	T0LOAD2	T0LOAD1	T0LOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T0LOAD[7:0]** – T0 重载寄存器，用于设置 T0 的计数周期

定时器 T0 比较寄存器

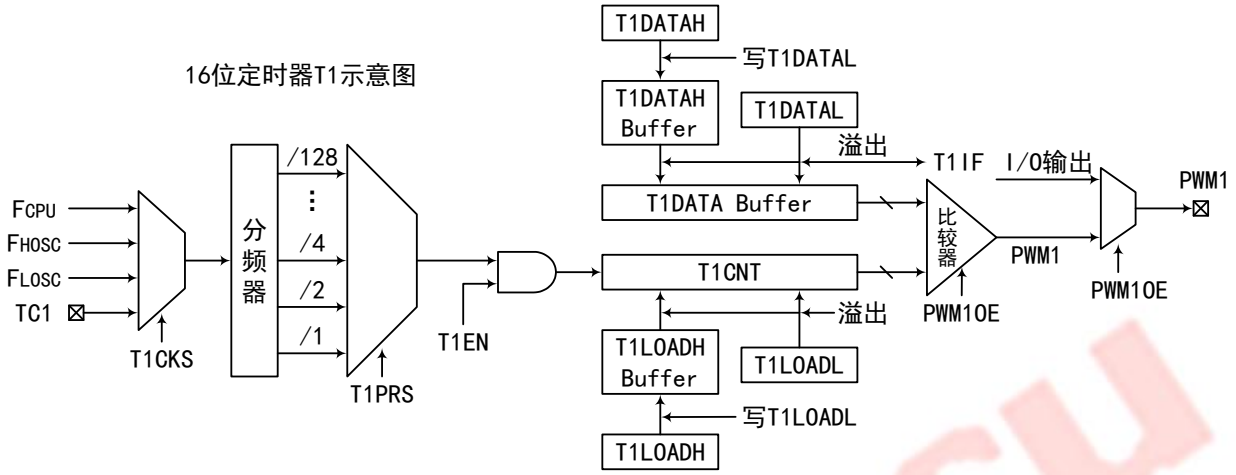
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0DATA	T0DATA7	T0DATA6	T0DATA5	T0DATA4	T0DATA3	T0DATA2	T0DATA1	T0DATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **T0DATA[7:0]** – T0 比较寄存器，用于设置 PWM0 的占空比

7.3 定时器 T1

定时器 T1 为 16 位定时/计数器，包含 1 个 16 位递减计数器、可编程预分频器、控制寄存器、16 位重载寄存器及 16 位比较寄存器。

- ✧ 可通过预分频比设置计数频率，可通过重载寄存器控制计数周期；
- ✧ 支持 16 位 PWM 输出，可通过比较寄存器设置 PWM 占空比；
- ✧ 支持溢出中断和溢出唤醒功能；



定时器 T1 除位数为 16 位外，其余功能及操作模式均与定时器 T0 相同，不同的是 T1 内部配有 2 个 8 位缓冲器和 1 个 16 位缓冲器。

T1LOADH 配有 1 个 8 位缓冲器，写 T1LOADL 时会同时将 T1LOADH 值写入该缓冲器中。若此时 T1EN=0，则会同时将[T1LOADH:T1LOADL]的值写入 T1CNT；若 T1EN=1，则需在 T1CNT 溢出后，才将[缓冲器:T1LOADL]的值写入 T1CNT。所以调整 T1LOAD 值时需先写 T1LOADH，再写 T1LOADL。

T1DATA 配有 1 个 16 位缓冲器和 1 个 8 位缓冲器，写 T1DATAL 时会同时将 T1DATAH 的值写入 8 位缓冲器中；16 位缓冲器 T1DATABUF 用于与 T1CNT 比较，当 T1EN=0 时，写 T1DATAL 的同时会将[8 位缓冲器:T0DATAL]的值写入 T1DATABUF 中，当 T1EN=1 时，只在 T1CNT 溢出时将[8 位缓冲器:T0DATAL]的值写入 T1DATABUF 中；在配置占空比时，需先写 T1DATAH，再写 T1DATAL。

定时器 T1 可实现 16 位 PWM 功能，操作模式与 T0 相同。PWM1 占空比的计算如下：

- ✧ PWM1 高电平时间 =（T1DATA）× T1CNT 计数时钟周期
- ✧ PWM1 周期（T1 的溢出周期）=（T1LOAD+1）×T1CNT 计数时钟周期
- ✧ PWM1 占空比 =（T1DATA）/（T1LOAD+1）

定时器 T1 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CR	T1EN	PWM1OE	-	T1CKS1	T1CKS0	T1PRS2	T1PRS1	T1PRS0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
初始值	0	0	-	0	0	0	0	0

BIT[7] **T1EN** – T1 使能控制位

0： 关闭 T1；

1： 开启 T1；

BIT[6] **PWM1OE** – PWM1 功能控制位

0： 关闭 PWM1 功能，并禁止端口输出 PWM 波形；

1： 使能 PWM1 功能，并允许端口输出 PWM 波形；

BIT[4:3] **T1CKS[1:0]** – T1 时钟源选择位

T1CKS[1:0]	T1 时钟源
00	FCPU
01	FHOSC
10	FLOSC
11	TC1 上升沿

BIT[2:0] **T1PRS[2:0]** – T1 预分频比选择位

T1PRS[2:0]	T1 时钟预分频比
000	1:1
001	1:2
010	1:4
011	1:8
100	1:16
101	1:32
110	1:64
111	1:128

定时器 T1 计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CNTH	T1CNT15	T1CNT14	T1CNT13	T1CNT12	T1CNT11	T1CNT10	T1CNT9	T1CNT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T1CNT[15:8]** – T1 计数器高 8 位，为可读写的递减计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CNTL	T1CNT7	T1CNT6	T1CNT5	T1CNT4	T1CNT3	T1CNT2	T1CNT1	T1CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T1CNT[7:0]** – T1 计数器低 8 位，为可读写的递减计数器

定时器 T1 重载寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1LOADH	T1LOAD15	T1LOAD14	T1LOAD13	T1LOAD12	T1LOAD11	T1LOAD10	T1LOAD9	T1LOAD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T1LOAD[15:8]** – T1 重载寄存器高 8 位，用于设置 T1 的计数周期

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1LOADL	T1LOAD7	T1LOAD6	T1LOAD5	T1LOAD4	T1LOAD3	T1LOAD2	T1LOAD1	T1LOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T1LOAD[7:0]** – T1 重载寄存器低 8 位，用于设置 T1 的计数周期

定时器 T1 比较寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1DATAH	T1DATA15	T1DATA14	T1DATA13	T1DATA12	T1DATA11	T1DATA10	T1DATA9	T1DATA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **T1DATA[15:8]** – T1 比较寄存器高 8 位，用于设置 PWM1 的占空比

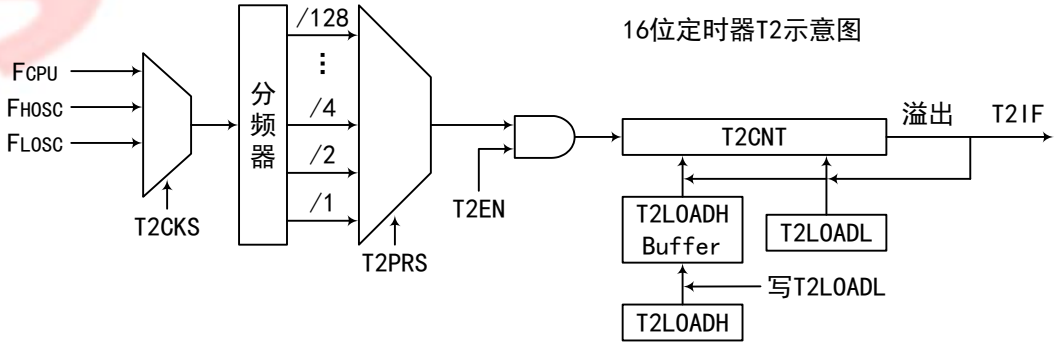
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1DATAL	T1DATA7	T1DATA6	T1DATA5	T1DATA4	T1DATA3	T1DATA2	T1DATA1	T1DATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **T1DATA[7:0]** – T1 比较寄存器低 8 位，用于设置 PWM1 的占空比

7.4 定时器 T2

定时器 T2 为 16 位定时器，包含 1 个 16 位递减计数器、可编程预分频器、控制寄存器、16 位重载寄存器。

- ✧ 可通过预分频比设置计数频率，可通过重载寄存器控制计数周期；
- ✧ 支持溢出中断和溢出唤醒功能；
- ✧ 可作为 UART 的波特率发生器；



定时器 T2 的定时功能与定时器 T1 完全相同。

定时器 T2 还可用作内部 UART 模块的波特率发生器，当 UART 工作在方式 1 或方式 3 时，其波特率为 T2 溢出率的 1/16。

定时器 T2 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CR	T2EN	SMOD	SSTAT	T2CKS1	T2CKS0	T2PRS2	T2PRS1	T2PRS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] **T2EN** – T2 使能控制位

0： 关闭 T2；

1： 开启 T2；

BIT[6] **SMOD** – 串口 UART 波特率选择位（仅 UART 方式 2 有效）

0： 方式 2 的波特率为 FCPU/64；

1： 方式 2 的波特率为 FCPU/32；

BIT[5] **SSTAT** – 串口 UART 寄存器 SCON 功能选择位

0： 寄存器 SCON[7:5]功能为 SM[0:2]；

1： 寄存器 SCON[7:5]功能为 FE，RXOV，TXCOL；

BIT[4:3] **T2CKS[1:0]** – T2 时钟源选择位

T2CKS[1:0]	T2 时钟源
00	FCPU
01	FHOSC
10	FLOSC
11	保留

BIT[2:0] **T2PRS[2:0]** – T2 预分频比选择位

T2PRS[2:0]	T2 时钟预分频比
000	1:1
001	1:2
010	1:4
011	1:8
100	1:16
101	1:32
110	1:64
111	1:128

定时器 T2 计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CNTH	T2CNT15	T2CNT14	T2CNT13	T2CNT12	T2CNT11	T2CNT10	T2CNT9	T2CNT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T2CNT[15:8]** – T2 计数器高 8 位，为可读写的递减计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CNTL	T2CNT7	T2CNT6	T2CNT5	T2CNT4	T2CNT3	T2CNT2	T2CNT1	T2CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T2CNT[7:0]** – T2 计数器低 8 位，为可读写的递减计数器

定时器 T2 重载寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2LOADH	T2LOAD15	T2LOAD14	T2LOAD13	T2LOAD12	T2LOAD11	T2LOAD10	T2LOAD9	T2LOAD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T2LOAD[15:8]** – T2 重载寄存器高 8 位，用于设置 T2 的计数周期

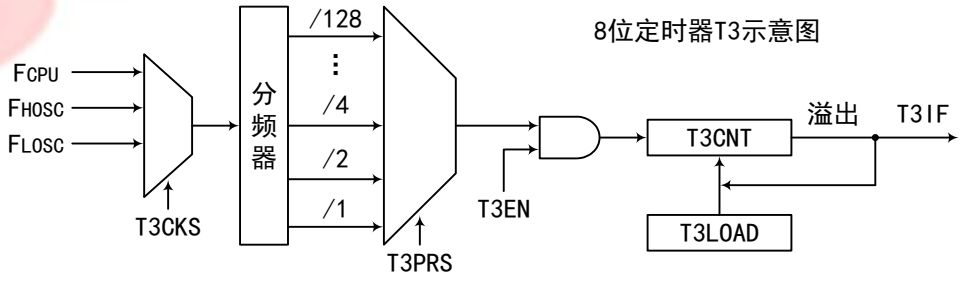
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2LOADL	T2LOAD7	T2LOAD6	T2LOAD5	T2LOAD4	T2LOAD3	T2LOAD2	T2LOAD1	T2LOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T2LOAD[7:0]** – T2 重载寄存器低 8 位，用于设置 T2 的计数周期

7.5 定时器 T3

定时器 T3 为 8 位定时器，包含 1 个 8 位递减计数器、可编程预分频器、控制寄存器、8 位重载寄存器。

- ✧ 可通过预分频比设置计数频率，可通过重载寄存器控制计数周期；
- ✧ 支持溢出中断和溢出唤醒功能；



定时器 T3 的定时功能与定时器 T0 完全相同。

定时器 T3 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T3CR	T3EN	-	-	T3CKS1	T3CKS0	T3PRS2	T3PRS1	T3PRS0

R/W	R/W	-	-	R/W	R/W	R/W	R/W	R/W
初始值	0	-	-	0	0	0	0	0

BIT[7] **T3EN** – T3 使能控制位

0： 关闭 T3；

1： 开启 T3；

BIT[4:3] **T3CKS[1:0]** – T3 时钟源选择位

T3CKS[1:0]	T3 时钟源
00	FCPU
01	FHOSC
10	FLOSC
11	保留

BIT[2:0] **T3PRS[2:0]** – T3 预分频比选择位

T3PRS[2:0]	T3 时钟预分频比
000	1:1
001	1:2
010	1:4
011	1:8
100	1:16
101	1:32
110	1:64
111	1:128

定时器 T3 计数器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T3CNT	T3CNT7	T3CNT6	T3CNT5	T3CNT4	T3CNT3	T3CNT2	T3CNT1	T3CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T3CNT[7:0]** – T3 计数器，为可读写的递减计数器

定时器 T3 重载寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T3LOAD	T3LOAD7	T3LOAD6	T3LOAD5	T3LOAD4	T3LOAD3	T3LOAD2	T3LOAD1	T3LOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T3LOAD[7:0]** – T3 重载寄存器，用于设置 T3 的计数周期

8 模数转换器 ADC

8.1 ADC 概述

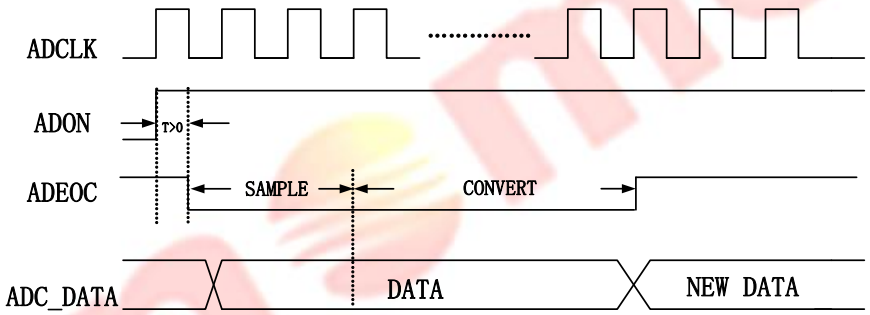
芯片内置的 ADC 模块为 12 位高精度逐次逼近型 ADC。

- ✧ 14 路外部通道：AN0~AN13；2 路内部通道：GND、VDD/4；
- ✧ 参考电压可选：VDD、内部参考电压 VIR（2V）；
- ✧ ADC 时钟：F_{HIRC} 的 8/16/32/64 分频；
- ✧ 支持零点校准；

ADC 模块通过 ADEN 位使能，通过 ADCHS 位选择转换的模拟通道，ADCKS 位选择转换速度，ADEOC 为 ADC 启动位及转换结束标志位。当 ADEOC 标志为 1 时，对该位写 0 将启动模数转换，转换完成后结果放在 ADRH/ADRL 中，ADEOC 将自动置 1，同时中断标志位 ADIF 置 1，提出中断请求。

采样（SAMPLE）时间可选择 4/8/15 个 ADCLK（即 ADC 时钟周期），转换（CONVERT）时间固定为 12 个 ADCLK，一次 ADC 转换为 16/20/27 个 ADCLK。

ADC 转换时序如下图所示：



注：

1、 AD 转换过程中或者 ADEN 未使能时，ADRH/ADRL 中的数据未知，应在 AD 转换结束且 ADEN 使能的情况下读取 AD 转换数据；

2、 若选择内部参考电压 VIR，则需保证 VDD> （VIR+0.5V），否则 VIR 将随之下降；

3、 使能 ADC 模块、切换参考电压等操作后，需待电路稳定（时间 > 200us）后才能启动 AD 转换；切换输入通道后，受外部输入影响，前两次转换的结果会有误差，建议舍弃；

4、 AD 转换精度受参考电压精度的影响，且内部参考电压下的转换精度，比外部参考电压略低 2 个 LSB 左右；

5、 转换时钟越慢、采样时间越长，则越能过滤外部输入的波动，越能保证 AD 转换的精度；

8.2 ADC 操作步骤

模数转换设置步骤：

(1) 设置相应端口为输入端口，关闭上下拉电阻；

(2) 通过 ADIOS，设置相应端口为模拟端口；

(3) 若转换时钟可选，则设置 ADCKS，选取适当的 ADC 转换时钟；

(4) 若采样时间可选，则设置 ADSPS，选取适当的 ADC 采样时间；

(5) 若参考电压可选，则设置 ADVRS，选择适当的参考电压；
(6) 若数据格式可选，则设置 ADRSEL，选择 ADC 结果的数据格式；
(7) ADEN 置 1，使能 ADC 模块；
(8) 设置 ADCHS，选择 ADC 转换通道；
(9) ADEOC 写 0，启动 AD 转换；
(10) 等待 ADEOC 硬件置 1（或利用 ADC 中断）；
(11) 读取 ADC 转换结果（ADRH、ADRL）；
(12) 重复（8）~（11），对不同的通道进行转换或对同一通道进行多次转换；

8.3 ADC 相关寄存器

ADC 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCR0	ADCHS3	ADCHS2	ADCHS1	ADCHS0	VRS1	VRS0	ADEOC	ADEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	0	0	1	0

BIT[7:4] **ADCHS[3:0]** – ADC 模拟通道选择位

ADCHS[3:0]	ADC 模拟通道选择
0000	AN0
0001	AN1
0010	AN2
0011	AN3
0100	AN4
0101	AN5
0110	AN6
0111	AN7
1000	AN8
1001	AN9
1010	AN10
1011	AN11
1100	AN12
1101	AN13
1110	片内 GND
1111	VDD/4

BIT[3:2] **ADVRS[1:0]** – ADC 参考电压选择位

ADVRS[1:0]	ADC 参考电压选择
00	内部 2.0V
11	VDD
其他	保留

BIT[1] **ADEOC** – ADC 启动位及转换结束标志位

0： AD 转换过程中，转换结束后自动置 1；

1： AD 转换结束，对 ADEOC 写入 0 启动 AD 转换；

BIT[0] **ADEN** – ADC 功能使能位

0： 关闭 ADC 功能；

1： 使能 ADC 功能；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCR1	ADRSEL	-	ADCKS1	ADCKS0	-	-	ADSPS1	ADSPS0
R/W	R/W	-	R/W	R/W	-	-	R/W	R/W
初始值	0	-	0	0	-	-	1	1

BIT[7] **ADRSEL** – ADC 转换数据格式选择位

0： ADC 转换结果 12 位数据，高 8 位存入 ADRH[7:0]、低 4 位存入 ADRL[3:0]；

1： ADC 转换结果 12 位数据，高 4 位存入 ADRH[3:0]、低 8 位存入 ADRL[7:0]；

BIT[5:4] **ADCKS[1:0]** – ADC 转换时钟选择位

ADCKS[1:0]	ADC 转换时钟 FADC
00	FHIRC/8
01	FHIRC/16
10	FHIRC/32
11	FHIRC/64

BIT[1:0] **ADSPS[1:0]** – ADC 采样时间选择位

ADSPS[1:0]	ADC 采样时间
00	保留
01	4 个 ADCLK
10	8 个 ADCLK
11	15 个 ADCLK

ADC 转换结果寄存器

ADRSEL=0 时：

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADRH	ADR11	ADR10	ADR9	ADR8	ADR7	ADR6	ADR5	ADR4
R/W	R	R	R	R	R	R	R	R
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **ADR[11:4]** – ADC 转换结果高 8 位

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADRL	-	-	-	-	ADR3	ADR2	ADR1	ADR0
R/W	-	-	-	-	R	R	R	R
初始值	-	-	-	-	X	X	X	X

BIT[3:0] **ADR[3:0]** – ADC 转换结果低 4 位

ADRSEL=1 时：

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADRH	-	-	-	-	ADR11	ADR10	ADR9	ADR8
R/W	-	-	-	-	R	R	R	R
初始值	-	-	-	-	X	X	X	X

BIT[3:0] **ADR[11:8]** – ADC 转换结果高 4 位

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADRL	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0
R/W	R	R	R	R	R	R	R	R
初始值	X	X	X	X	X	X	X	X

BIT[7:0] **ADR[7:0]** – ADC 转换结果低 8 位

零点偏移修调控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OSADJCR	OSADJEN	-	OSADJTR5	OSADJTR4	OSADJTR3	OSADJTR2	OSADJTR1	OSADJTR0
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	-	X	X	X	X	X	X

BIT[7] **OSADJEN** – ADC 零点偏移修调模式使能位（复位初始值为出厂设定值）

0： 关闭 ADC 零点偏移修调模式；

1： 使能 ADC 零点偏移修调模式；

BIT[5] **OSADJTR[5]** – ADC 零点偏移修调方向选择位（复位初始值为出厂设定值）

0： 负向修调，即根据修调电压减小转换值（转换结果大于理论值时应选择负向修调）；

1： 正向修调，即根据修调电压增加转换值（转换结果小于理论值时应选择正向修调）；

BIT[4:0] **OSADJTR[4:0]** – ADC 零点偏移修调电压选择位（复位初始值为出厂设定值）

OSADJTR[4:0]	修调电压（典型值）
00000	0mV
00001	0.5mV
00010	1.0mV
00011	1.5mV
00100	2.0mV
00101	2.5mV
00110	3.0mV
00111	3.5mV
01000	4.0mV
01001	4.5mV
01010	5.0mV
01011	5.5mV
01100	6.0mV
01101	6.5mV

01110	7.0mV
01111	7.5mV
10000	8.0mV
10001	8.5mV
10010	9.0mV
10011	9.5mV
10100	10.0mV
10101	10.5mV
10110	11.0mV
10111	11.5mV
11000	12.0mV
11001	12.5mV
11010	13.0mV
11011	13.5mV
11100	14.0mV
11101	14.5mV
11110	15.0mV
11111	15.5mV

8.4 ADC 零点偏移修调流程

(1) 将 ADC 的输入通道选择为内部接地，设置 OSADJEN=1；设置 ADC 时钟、采样时间等；
(2) 设置 OSADJTR[5:0]=00H；进行 ADC 转换；
(3) 如果 ADC 结果为 0，则执行（6）；
如果 ADC 结果不为 0，则执行（4）；
(4) OSADJTR[4:0]加 1 后进行 ADC 转换；
(5) 如果 ADC 结果为 0，则跳至（10）；
如果 ADC 结果不为 0，则执行（4），直到 ADC 结果为 0 或 OSADJTR[4:0]=1FH 后跳至（10）；
(6) 设置 OSADJTR[5:0]=3FH，进行 ADC 转换；
(7) 如果 ADC 结果为 0，则跳至（10）；
如果 ADC 结果不为 0，则执行（8）；
(8) OSADJTR[4:0]减 1 后进行 ADC 转换；
(9) 如果 ADC 结果为 0，则跳至（10）；
如果 ADC 结果不为 0，则执行（8），直到 ADC 结果为 0 或 OSADJTR[4:0]=00H 后跳至（10）；
(10) OSADJTR[5:0]中的值即为零点偏移最佳修调电压，修调流程结束，后续 ADC 工作时直接使用，不需要再次修调。

9 IIC 通讯接口

芯片内置 IIC 通讯模块，支持 IIC 总线接口。IIC 总线是双向两线结构，数据线 SDA 和时钟线 SCL 与 IO 端口复用，当 IIC 模块使能时，端口用作 SDA/SCL，此时为开漏输出，可选择内置或外接合适的上拉电阻，以匹配选定的通讯速率。

芯片 IIC 通讯模块仅支持 7 位地址编码的单主机模式，总线上的时钟信号始终由芯片 SCL 端口输出，时钟频率可选择 FHIRC/16、FHIRC/20、FHIRC/40、FHIRC/160。

9.1 数据传输

总线空闲时，SDA 线、SCL 线均为高电平。SCL 线为高电平期间，SDA 线由高电平变为低电平的下降沿表示起始信号 START；SCL 线为高电平期间，SDA 线由低电平变为高电平的上升沿表示停止信号 STOP。数据传输时，SCL 线为高电平期间，SDA 线上的电平必须保持稳定，高电平表示数据“1”、低电平表示数据“0”，只有在 SCL 线低电平期间，SDA 线的数据才允许变化

一帧数据传输以一个起始信号 START 开始，以一个停止信号 STOP 或重复起始信号 START 终止，一个重复 START 信号也是下一帧数据传输的开始（需从机支持重复 START 信号），总线不被释放。每一帧数据传输时需先由主机发送包括 7 位从机地址和 1 位读/写命令的控制字节，然后再由主机向从机发送或接收从机数据，每个字节传输的时钟为 9 位，前 8 位为控制字节或数据字节（最高位最先传输），第 9 位为应答位，应答位时 SDA 线的信号为应答信号，为 0 表示有应答（ACK），为 1 表示非应答（NACK）。

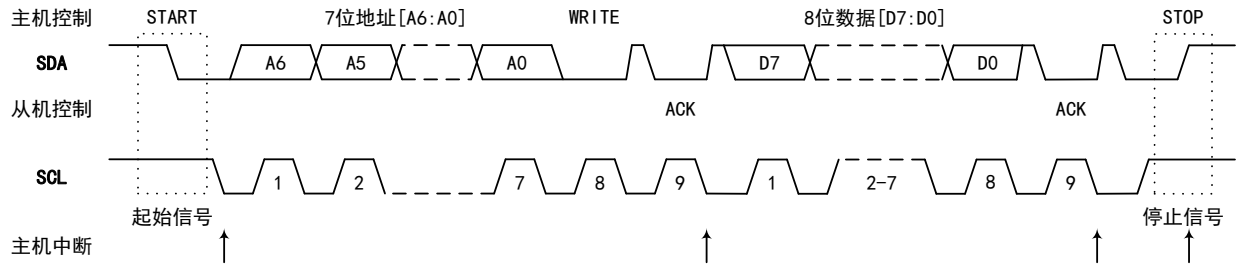
芯片在通讯时可选择两种中断触发模式，触发条件达成后将产生中断，中断标志将被置 1，并需要软件清 0 以正确响应下一次中断：

IICIMS=0 时，当主机发送完 8 位数据并接收完从机应答信号 ACK/NACK、主机接收完从机 8 位数据并发送完应答信号 ACK/NACK、主机发送完 STOP 位信号后，将产生中断：

IICIMS=1 时，当主机发送完 START 或 STOP 位信号、主机发送完 8 位数据并接收完从机应答信号 ACK/NACK、主机接收完从机 8 位数据、主机发送完应答信号 ACK/NACK 后，将产生中断。

主机到从机的数据传输

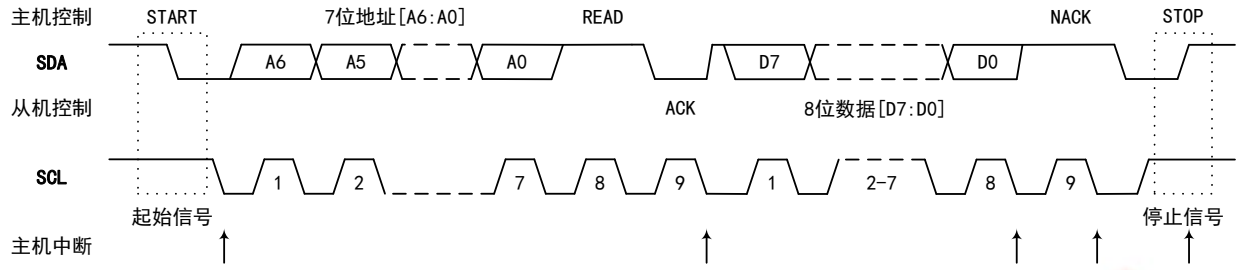
主机首先发送起始信号 START，再发送一个包含写命令的控制字节，从机返回一个 ACK，接着主机发送数据字节，从机在每一个字节接收完毕后返回一个 ACK，主机在接收到最后字节的从机应答 ACK 后，发送停止信号 STOP 结束本次传输。



主机发送数据的时序

从机到主机的数据传输

主机首先发送起始信号 START，再发送一个包含读命令的控制字节，从机返回一个 ACK，接着主机接收从机发送的数据字节，主机在每一个字节接收完毕后返回一个 ACK，但主机在最后一个字节接收完毕后返回一个非应答 NACK，并发送停止信号 STOP 结束本次数据传输。



主机接收数据的时序

9.2 IIC 相关寄存器

IIC 通讯控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IICCR	IICEN	IICRS	IICCKS1	IICCKS0	IICTYP1	IICTYP0	IICRDTP	IICIMS
R/W	R/W	R/W	R/W	R/W	R	R	R	R/W
初始值	0	0	0	0	0	1	0	0

BIT[7] **IICEN** – IIC 通讯使能位

0： 关闭 IIC 功能，端口用作通用 I/O；

1： 使能 IIC 功能，端口用作 SCL/SDA；

BIT[6] **IICRS** – 内置上拉电阻选择位

0： 内置上拉电阻为 4.7K；

1： 内置上拉电阻为 1.3K；

BIT[5:4] **IICCKS[1:0]** – 时钟频率（通讯速率）选择位

IICCKS[1:0]	时钟频率	通讯速率
00	FHIRC/160	100Kbps
01	FHIRC/40	400Kbps
10	FHIRC/20	800Kbps
11	FHIRC/16	1Mbps

注：实际通讯速率受电路寄生阻容及上拉电阻影响，并受中断响应速度和 CPU 处理速度影响。

BIT[3:2] **IICTYP[1:0]** – IIC 主机传输状态位

IICTYP[1:0]	主机传输状态
00	START 信号已发送
01	STOP 信号已发送

10	已发送 8 位数据，并已接收 1 位从机应答信号
11	已接收 8 位数据（IICRDTP=0）， 或已发送 1 位主机应答信号（IICRDTP=1）

注：IICTYP 标志用于判断上一步完成的传输过程，由此决定下一步将要进行的传输过程。

BIT[1] **IICRDTP** – IIC 主机读操作状态位（IICTYP=11 时有效）

0： 主机已接收从机的 8 位数据；

1： 主机已向从机发送应答信号；

BIT[0] **IICIMS** – IIC 中断触发模式选择位

0： 主机发完 START 或收到从机数据后，不产生中断；

1： 主机发完 START 或收到从机数据后，将产生中断，中断标志将置 1；

IIC 通讯状态寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IICSR	IICSTR	IICSTP	IICRD	IICWR	IICACK	SACK	RACK	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	-
初始值	0	0	0	0	0	0	0	-

BIT[7] **IICSTR** – 起始信号发送控制位

0： 不发送 START 起始信号；

1： 写 1 将发送 START 起始信号，发送完成后自动清 0；

BIT[6] **IICSTP** – 停止信号发送控制位

0： 不发送 STOP 停止信号；

1： 写 1 将发送 STOP 停止信号，发送完成后自动清 0；

BIT[5] **IICRD** – 主机接收数据使能位

0： 主机不启动接收从机数据的操作；

1： 写 1 将启动主机接收从机 8 位数据的操作，接收完成后自动清 0；

BIT[4] **IICWR** – 主机发送数据使能位

0： 主机不启动向从机发送数据的操作；

1： 写 1 将启动主机向从机发送 8 位数据再接收从机应答信号的操作，接收完后自动清 0；

BIT[3] **IICACK** – 主机发送应答信号使能位

0： 主机不启动向从机发送应答信号的操作；

1： 写 1 将启动主机向从机发送 1 位应答信号（SACK 内容）的操作，发送完后自动清 0；

BIT[2] **SACK** – 发送应答信号时的数据

0： 应答信号为应答（ACK）；

1： 应答信号为非应答（NACK）；

BIT[1] **RACK** – 接收从机应答信号的数据

0： 从机应答信号为低电平，从机有应答；
1： 从机应答信号为高电平，从机无应答；

IIC 通讯数据寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IICDR	IICDR7	IICDR6	IICDR5	IICDR4	IICDR3	IICDR2	IICDR1	IICDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **IICDR[7:0]** – IIC 通讯数据
IICDR：主机发送时的控制字节或发送数据；主机接收时发送的控制字节或接收到的数据。

9.3 应用流程说明

主机发送流程

主机发送时，数据由主机发送到从机，流程如下：

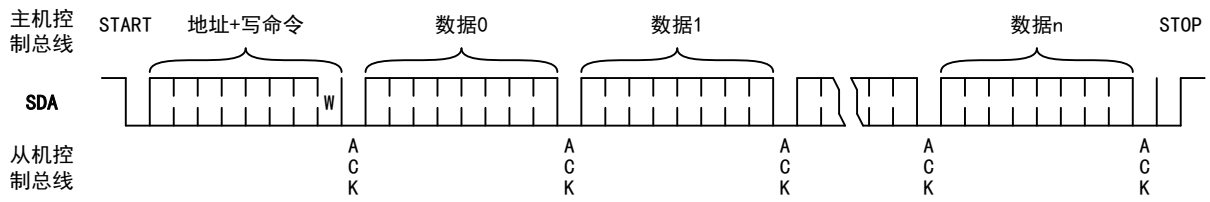
(1) 初始化 IIC 通讯模块：
- ✓ 通过 IICIMS，设置 IIC 的中断触发模式（本流程中 IICIMS 设置为 1）；
- ✓ 通过 IICCKS[1:0]，设置 IIC 的通讯速率；
- ✓ 中断标志位 IICIF 清 0，中断使能位 IICIE 置 1，使能 IIC 中断；
- ✓ IICEN 位置 1，使能 IIC 通讯模块；
- ✓ IICSTR 位置 1，主机发送一个 START 信号，启动 IIC 通讯；

(2) START 发送完后，主机产生第 1 个中断，软件将中断标志位清 0，并将控制字（7 位地址+写命令位）写入 IICDR 中，IICWR 位置 1，硬件自动发送 8 位数据及应答位时钟；

(3) 从机收到控制字节后返回 ACK/NACK 信号，主机接收后产生第 2 个中断：
- ✓ 如果 RACK 位=0，表明通讯成功，软件将下一个发送数据写入 IICDR 中并 IICWR 位置 1，硬件自动发送数据及应答位时钟；
- ✓ 如果 RACK 位=1，表明通讯错误，软件可将 IICSTP 位置 1 以中止通讯，或将 IICSTR 位置 1 以重新开始第二次通讯；

(4) 重复（3）直到所有字节都发送成功，最后将 IICSTP 位置 1 发送 STOP 信号以终止通讯；

(5) STOP 发送完后，主机产生最后 1 个中断，可以借此停止 IIC 模块释放 IO 端口；



主机发送数据的格式

注：IICIMS 设置为 0 时，主机发送 START 位后不产生中断，将紧接着发送 7 位地址和写命令，发送完后将进入中断，此时 START 信号传输状态 IICTYP 无意义。

主机接收流程

主机接收时，数据由从机发送到主机，流程如下：

(1) 初始化 IIC 通讯模块：
- ✓ 通过 IICIMS，设置 IIC 的中断触发模式（本流程中 IICIMS 设置为 1）；
- ✓ 通过 IICCKS[1:0]，设置 IIC 的通讯速率；
- ✓ 中断标志位 IICIF 清 0，中断使能位 IICIE 置 1，使能 IIC 中断；
- ✓ IICEN 位置 1，使能 IIC 通讯模块；
- ✓ START 位置 1，产生一个起始信号，启动 IIC 通讯；

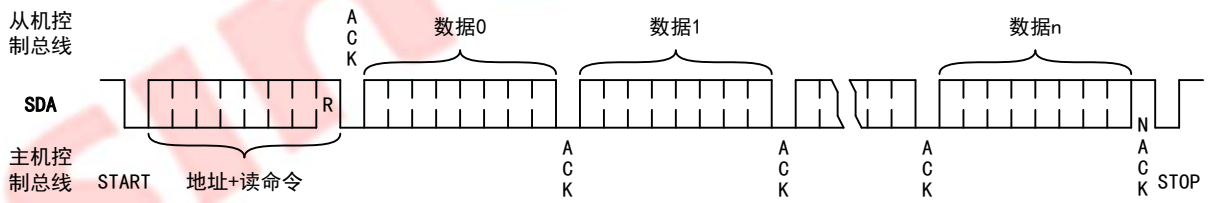
(2) 当 START 发送完后，主机产生第 1 个中断，软件将中断标志位清 0，并将控制字（7 位地址+读命令位）写入 IICDR 中，IICWR 位置 1，硬件自动发送 8 位数据及应答位时钟；

(3) 从机收到控制字节后返回 ACK/NACK 信号，主机接收后产生第 2 个中断：
- ✓ 如果 RACK 位=0，表明通讯成功，软件清 0 中断标志位，IICRD 位置 1 启动数据接收，硬件自动发送时钟信号，等待从机发送数据；
- ✓ 如果 RACK 位=1，表明通讯错误，软件可将 IICSTP 位置 1 以中止通讯，或将 IICSTR 位置 1 以重新开始第二次通讯。

(4) 当接收完从机发送的数据时，主机产生第 3 个中断：
- ✓ 如果继续接收，则 SACK 位清 0 并将 IICACK 位置 1，给予从机应答；
- ✓ 如果停止接收，则 SACK 位置 1 并将 IICACK 位置 1，不给予从机应答；

(5) 发送完应答信号 ACK/NACK 后，主机产生第 4 个中断：
- ✓ 如果 SACK 位=0 则表示将继续接收，将 IICRD 位置 1 启动下一个数据接收；重复（4）（5）直到停止接收；
- ✓ 如果 SACK 位=1 则表示停止接收，主机可将 IICSTP 位置 1 发送 STOP 信号以终止通讯，或将 IICSTR 位置 1 发送 RESTART 信号重新开始。

(6) STOP 发送完后，主机产生最后 1 个中断，可以借此停止 IIC 模块释放 IO 端口。



主机接收数据的格式

注：IICIMS 设置为 0 时，主机发送 START 位后不产生中断，将紧接着发送 7 位地址和写命令，发送完后将进入中断，此时 START 信号传输状态 IICTYP 无意义；此种情况下，主机接收完 8 位数据后也不产生中断，将紧接着发送 ACK/NOACK 信号，此后产生中断，因此在接收最后一个字节时，SACK 位需提前设置好。此时 IICRDTP 信号无意义。

10 增强型异步通讯 EUART

10.1 概述

芯片包含 1 个增强型通用异步收发器 EUART，波特率可选择为系统时钟分频或者定时器 T2 的时钟溢出率；EUART 的增强功能包括帧出错检测以及自动地址识别。

EUART 支持 8 位同步、8 位异步、9 位异步固定波特率、9 位异步可变波特率等 4 种工作方式。

10.2 工作方式

EUART 有 4 种工作方式。在通讯之前必须先初始化串口控制寄存器 SCON，选择 EUART 的工作方式和波特率。如果使用方式 1 或方式 3 还应先初始化定时器 T2。

在所有四种方式中，任何将串口缓冲寄存器 SBUF 作为目标寄存器的写操作都会启动发送。在方式 0 中由条件 RI=0 和 REN=1 初始化接收，将在 TxD 引脚上产生 1 个时钟信号，然后在 RxD 引脚上串行移入/移出 8 位数据。在其他方式中则利用外部输入的起始位来初始化接收（如果 REN=1），外部发送器通过发送起始位开始通信。

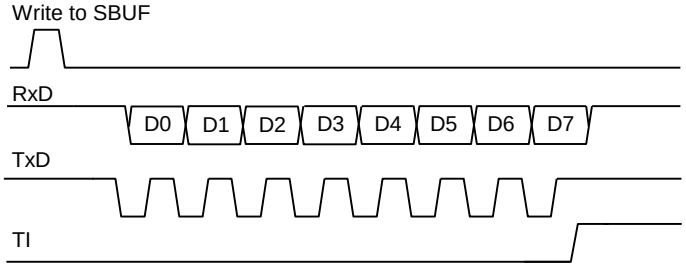
EUART 方式列表

SM[0:1]	方式	类型	波特率	帧长度	起始位	停止位	第 9 位
00	0	同步	FCPU/4 或 FCPU/12	8 位	无	无	无
01	1	异步	T2 溢出率/16	10 位	1	1	无
10	2	异步	FCPU/32 或 FCPU/64	11 位	1	1	0/1
11	3	异步	T2 溢出率/16	11 位	1	1	0/1

方式 0：8 位同步半双工

方式 0 支持与外部设备的同步通信。EUART 通过 TxD 引脚发送移位时钟，在 RxD 引脚上收发串行数据。因此这个方式是串行通信的半双工方式，在此方式中，每帧收发 8 位，低位先接收/发送。

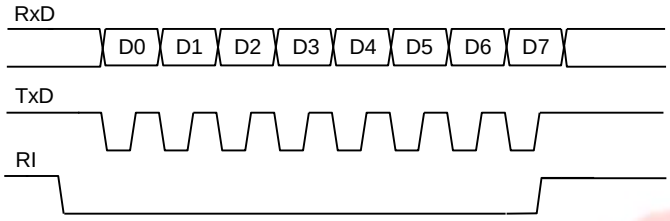
通过设置寄存器 SCON 中的 SM2 位为 0 或 1，波特率固定为系统时钟的 1/12 或 1/4。当 SM2 位为 0 时，串行端口以系统时钟的 1/12 运行；当 SM2 位置 1 时，串行端口以系统时钟的 1/4 运行。



Send Timing of Mode 0

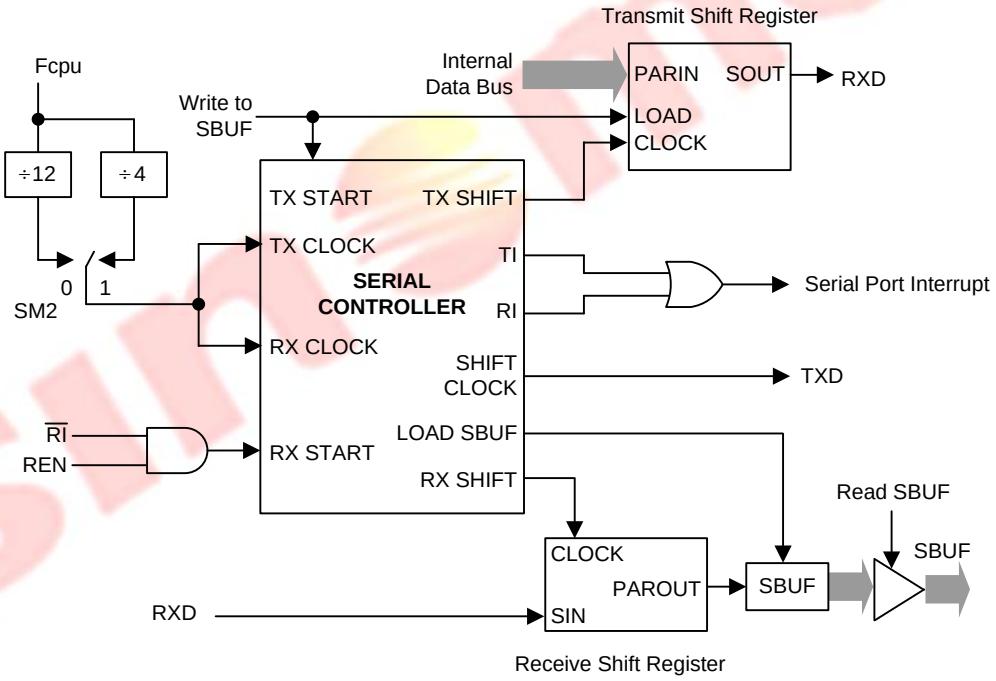
EUART 模块通过 TxD 引脚输出同步时钟，通过 RxD 引脚将数据读入或移出串行端口。任何将 SBUF 作为目标寄存器的写操作都会启动发送。下一个系统时钟 Tx 控制块开始发送。数据转换发生在移位时钟的下降沿，移位寄存器的内容逐次从左往右移位，空位置 0。当移位寄存器中的所有 8 位都发送后，TX 控制模块停止发送操作，然后在下一个系统时钟的上升沿将 TI 置 1。

REN 位置 1 和 RI 位清 0 将初始化接收。下一个系统时钟启动接收，在移位时钟的上升沿锁存数据，接收转换寄存器的内容逐次向左移位。当所有 8 位都接收到接收移位寄存器中后，RX 控制模块停止接收，然后在下一个系统时钟的上升沿上 RI 置 1，直到被软件清 0 才允许接收。



Receive Timing of Mode 0

方式 0 功能块框图如下图所示：

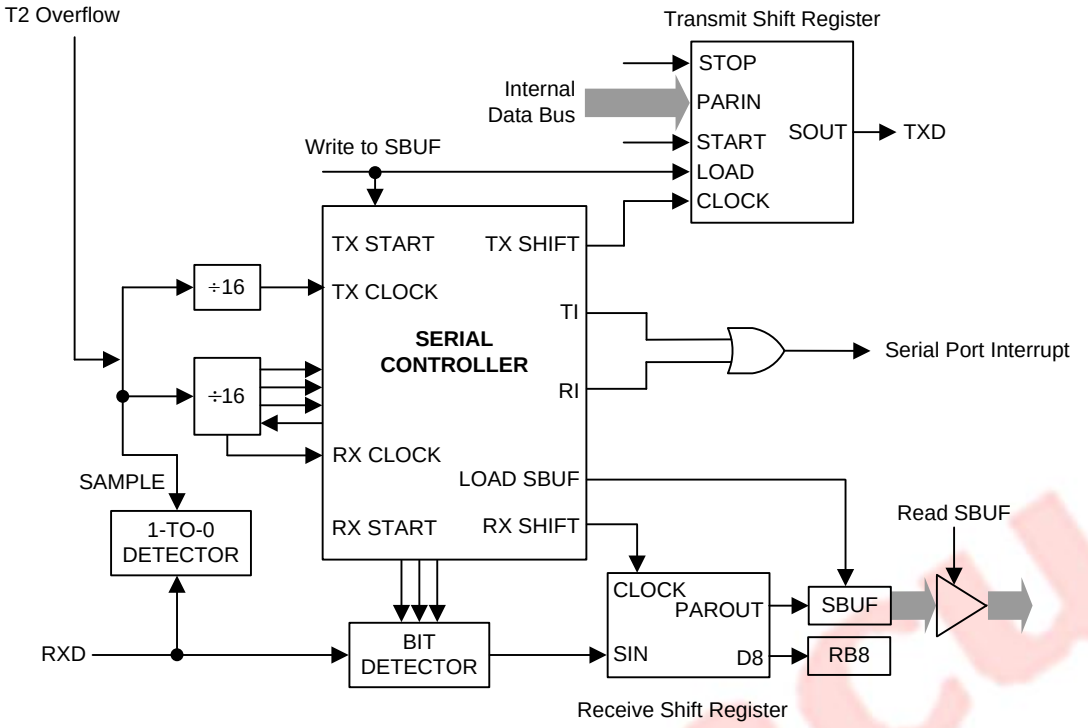


Receive Shift Register

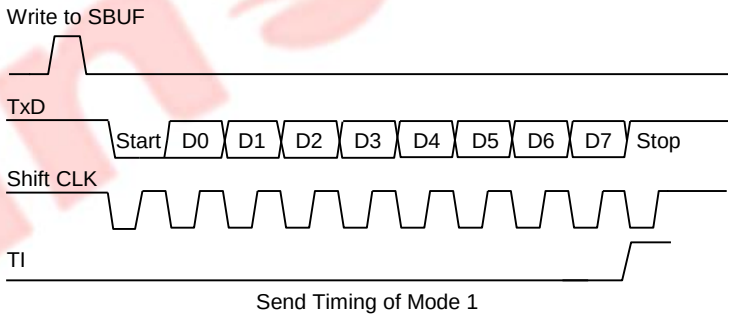
方式 1：8 位异步全双工，可变波特率

方式 1 提供 10 位全双工异步通信，10 位由一个起始位（逻辑 0）、8 个数据位（低位在前）和一个停止位（逻辑 1）组成。在接收时，这 8 个数据位存储在 SBUF 中而停止位储存在 RB8 位中。方式 1 中的波特率是可变的，串行收发波特率为定时器 T2 溢出率的 1/16。

方式 1 功能块框图如下图所示：



任何将 SBUF 作为目标寄存器的写操作都会启动发送，实际上发送是从 16 分频计数器中的下一次跳变之后的系统时钟开始的，因此位时间与 16 分频计数器是同步的，与对 SBUF 的写操作不同步。起始位首先在 TxD 引脚上移出，然后是 8 位数据位。在发送移位寄存器中的所有 8 位数据都发送完后，停止位在 TxD 引脚上移出，在停止位发出的同时 TI 标志置 1。



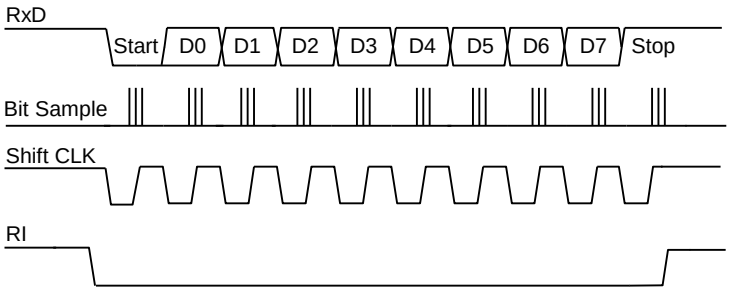
Send Timing of Mode 1

只有 REN 位置 1 时才允许接收。当 RxD 引脚检测到下降沿时串行口开始接收串行数据。为此，CPU 对 RxD 不断采样，采样速率为波特率的 16 倍。当检测下降沿时，16 分频计数器立即复位，这有助于 16 分频计数器与 RxD 引脚上的串行数据位同步。16 分频计数器把每一位的时间分为 16 个状态，在第 7、8、9 状态时，位检测器对 RXD 端的电平进行采样。为抑制噪声，在这 3 个状态采样中至少有 2 次采样值一致数据才被接收。如果所接收的第一位不是 0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待 RxD 引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。8 个数据位和 1 个停止位移入之后，移位寄存器的内容被分别装入 SBUF 和 RB8 中，RI 置 1，但必须满足下列条件：

1、RI=0；

2、SM2=0，或者接收的停止位=1；

如果这些条件被满足，那么停止位装入 RB8，8 个数据位装入 SBUF，RI 被置 1。否则接收的帧会丢失。这时，接收器将重新去探测 RxD 端是否另一个下降沿。用户必须用软件清除 RI，然后才能再次接收。

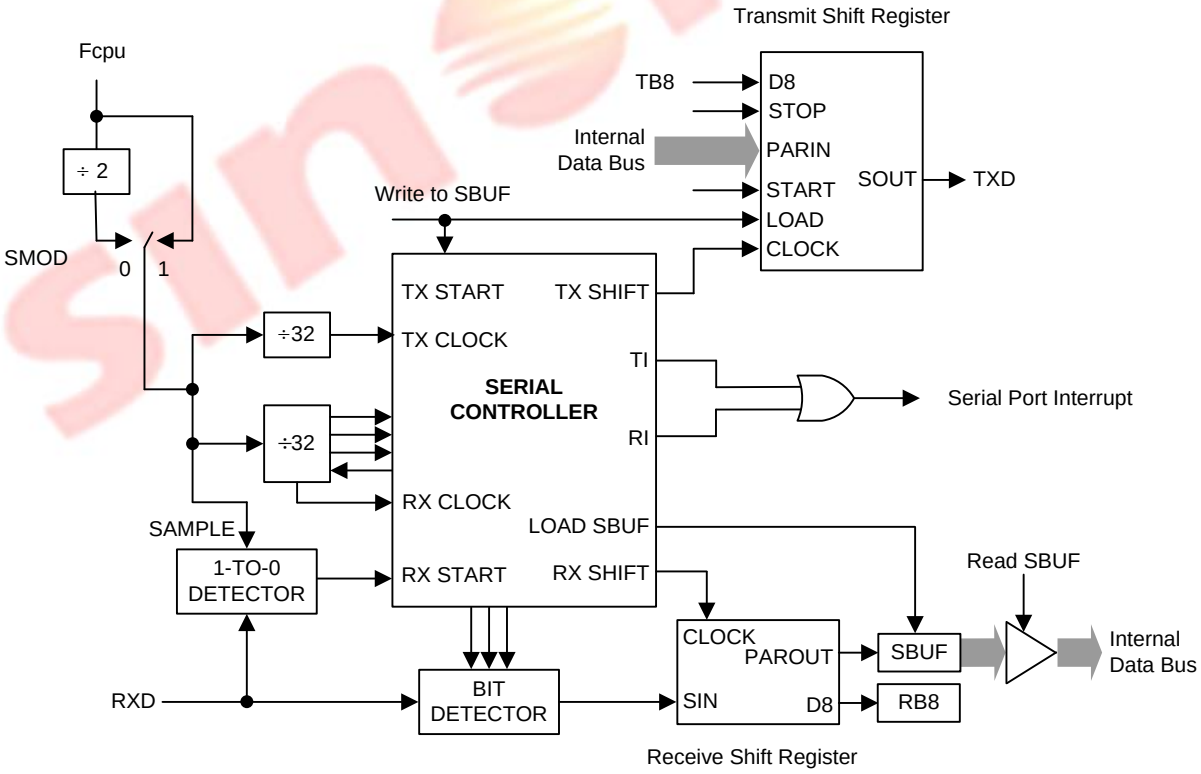


Receive Timing of Mode 1

方式 2：9 位异步全双工，固定波特率

这个方式使用异步全双工通信中的 11 位。一帧由一个起始位（逻辑 0），8 个数据位（低位在前），一个可编程的第 9 数据位和一个停止位（逻辑 1）组成。方式 2 支持多机通信和硬件地址识别（详见多机通讯章节）。在数据传送时，第 9 数据位（SCON 中的 TB8）可以写 0 或 1，例如，可写入 PSW 中的奇偶位 P，或用作多机通信中的数据/地址标志位。当接收到数据时，第 9 数据位进入 RB8 而停止位不保存。T2CR 中的 SMOD 位选择波特率为系统工作频率的 1/32 或 1/64。

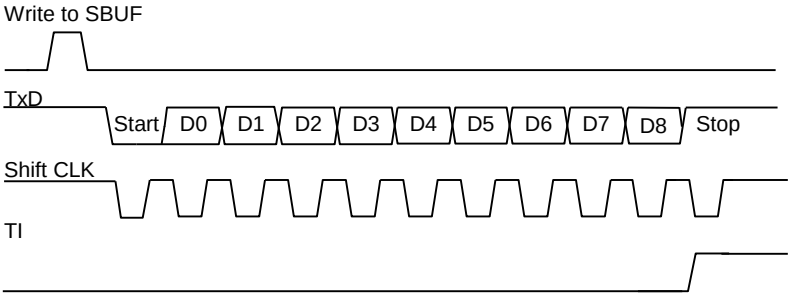
方式 2 功能块框图如下所示：



Receive Shift Register

任何将 SBUF 作为目标寄存器的写操作都会启动发送，同时也将 TB8 载入到发送移位寄存器的第 9 位中。实际上发送是从 16 分频计数器中的下一次跳变之后的系统时钟开始的，因此位时间与 16 分频计

数器是同步的，与对 SBUF 的写操作不同步。起始位首先在 TxD 引脚上移出，然后是第 9 位数据。在发送转换寄存器中的所有 9 位数据都发送完后，停止位在 TxD 引脚上移出，在停止位开始发送时 Tl 标志置 1。



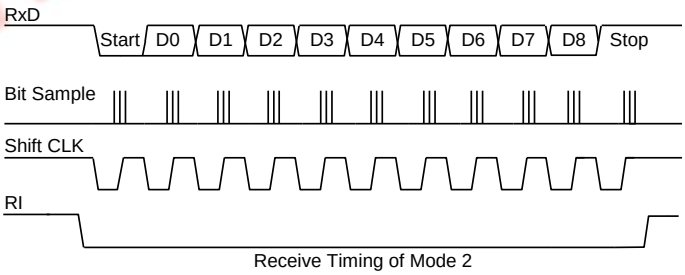
Send Timing of Mode 2

只有 REN 位置 1 时才允许接收。当 RxD 引脚检测到下降沿时串行口开始接收串行数据。为此，CPU 对 RxD 不断采样，采样速率为波特率的 16 倍。当检测下降沿时，16 分频计数器立即复位。这有助于 16 分频计数器与 RxD 引脚上的串行数据位同步。16 分频计数器把每一位的时间分为 16 个状态，在第 7、8、9 状态时，位检测器对 RXD 端的电平进行采样。为抑制噪声，在这 3 个状态采样中至少有 2 次采样值一致数据才被接收。如果所接收的第一位不是 0，说明这位不是一帧数据的起始位，该位被忽略，接收电路被复位，等待 RxD 引脚上另一个下降沿的到来。若起始位有效，则移入移位寄存器，并接着移入其它位到移位寄存器。9 个数据位和 1 个停止位移入之后，移位寄存器的内容被分别装入 SBUF 和 RB8 中，RI 置 1，但必须满足下列条件：

1、RI=0；

2、SM2=0 或者接收的第 9 位=1，且接收的字节符合实际从机地址；

如果这些条件被满足，那么第 9 位移入 RB8，8 位数据移入 SBUF，RI 被置 1。否则接收的数据帧会丢失。在停止位的当中，接收器回到寻找 RxD 引脚上的另一个下降沿。用户必须用软件清除 RI，然后才能再次接收。

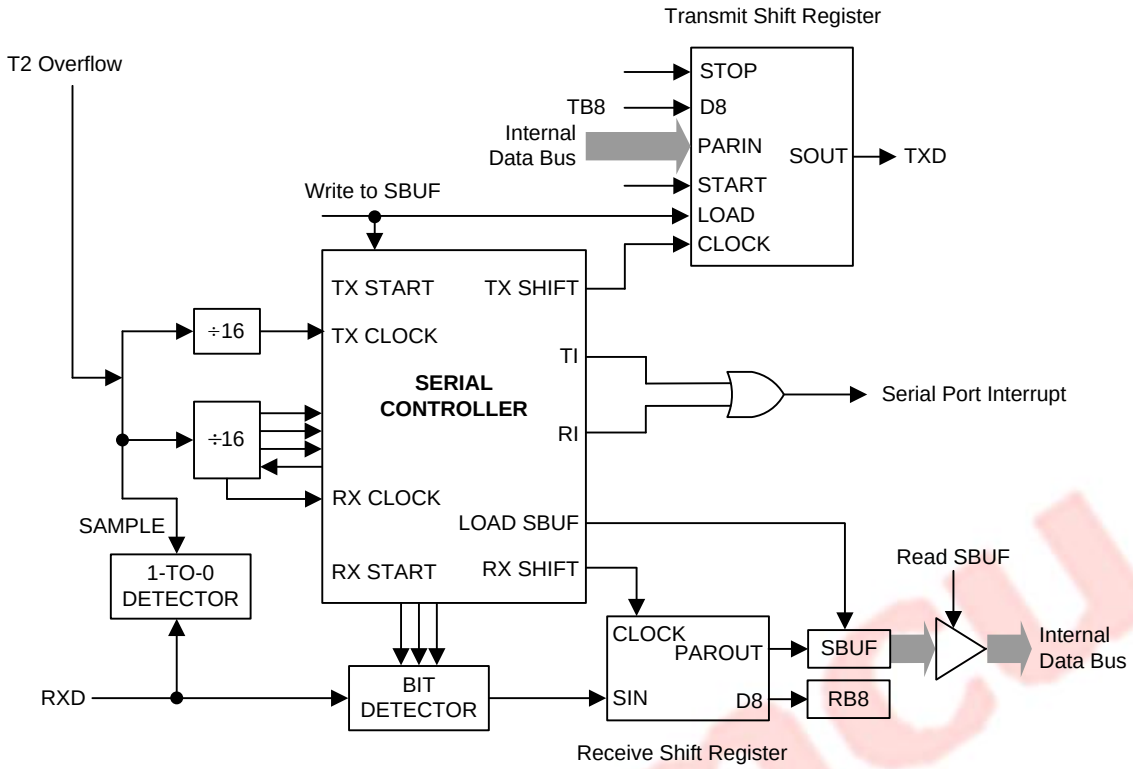


Receive Timing of Mode 2

方式 3：9 位异步全双工，可变波特率

方式 3 使用方式 2 的传输协议以及方式 1 的波特率产生方式。

方式 3 功能块框图如下所示：



10.3 波特率

在方式 0 中，波特率可编程为系统时钟的 1/12 或 1/4，由 SM2 位决定。当 SM2 为 0 时，串行端口在系统时钟的 1/12 下运行；当 SM2 为 1 时，串行端口在系统时钟的 1/4 下运行。

在方式 1 和方式 3 中，波特率为定时器 T2 的溢出率，方式 1 和方式 3 波特率公式：

BaudRate = F_{T2} / [16*(T2LOAD+1)]，F_{T2} 为 T2 的时钟频率；

注：T2 用于 UART 时 T2LODA 不能为 0，否则无法正确产生波特率。

在方式 2 中，波特率固定为系统时钟的 1/32 或 1/64，由 SMOD 位决定。当 SMOD 位为 0 时，EUART 以系统时钟的 1/64 运行；当 SMOD 位为 1 时，EUART 以系统时钟的 1/32 运行。

10.4 多机通讯

软件地址识别

方式 2 和方式 3 有一个专门的适用于多机通讯的功能。在这两个方式下，接收的是 9 位数据，第 9 位移入 RB8 中，然后再来一位停止位。EUART 可以这样设定：当接收到停止位时，只有在 RB8=1 的条件下，串行口中断才会有效（请求标志 RI 置 1）。可通过将 SCON 寄存器的 SM2 位置 1 使 EUART 具有这个功能。

在多机通讯系统中，以如下所述来利用这一功能。当主机要发送一数据块给几个从机中的一个时，先送出一地址字节，以辨认目标从机。地址字节与数据字节可用第 9 数据位来区别，地址字节的第 9 位

为 1，数据字节的第 9 位为 0。

如果从机 SM2 为 1，则不会响应数据字节中断。地址字节可以中断所有从机，这样，每一个从机都检查所接收到的地址字节，以判别自己是不是目标从机。被寻到的从机将 SM2 位清 0，并准备接收即将到来的数据字节，当接收完毕时，从机再一次将 SM2 置 1。没有被寻址的从机，则维持其 SM2 位为 1，忽略到来的数据字节，继续做自己的事情。

注：在方式 0 中，SM2 用来选择波特率加倍。在方式 1 中，SM2 用来检测停止位是否有效，如果 SM2 = 1，接收中断不会响应直到接收到一个有效的停止位。

自动（硬件）地址识别

在方式 2 和方式 3 中，SM2 置 1 将使 EUART 在如下状态运行：当 1 个停止位被接收时，如果载入 RB8 的第 9 数据位为 1（地址字节）并且接收到的数据字节符合 EUART 的从机地址，EUART 产生一个中断。接着，从机应将 SM2 清零，以接收后续的数据字节。

在 9 位方式下要求第 9 位为 1 以表明该字节是地址而非数据。当主机要发送一组数据给几个从机中的一个时，必须先发送目标从机的地址。所有从机在等待接收地址字节时，为了确保仅在接收地址字节时产生中断，SM2 位必须置 1。自动地址识别的特点是只有地址匹配的从机才能产生中断，地址比较通过硬件完成而不是软件。

中断产生后，地址相匹配的从机清零 SM2，继续接收数据字节。地址不匹配的从机不受影响，将继续等待接收和它匹配的地址字节。一旦全部信息接收完毕，地址匹配的从机应该再次把 SM2 置 1，忽略所有传送的非地址字节，直到接收到下一个地址字节。

使用自动地址识别功能时，主机可以通过调用给定的从机地址选择与一个或多个从机通信。使用广播地址可以联系所有的从机。有两个特殊功能寄存器用来定义从机地址（SADDR）和地址屏蔽（SADEN）。从机地址是一个 8 位的字节，存于 SADDR 寄存器中。SADEN 用于定义 SADDR 内位的有效与否，如果 SADEN 中某一位为 0，则 SADDR 中相应位的被忽略，如果 SADEN 中某一位置 1，则 SADDR 中相应位的将用于得到给定的从机地址。这可以使用户在不改变 SADDR 寄存器中的从机地址的情况下灵活地寻址多个从机。使用给定地址可以识别多个从机而排除其他的从机。

	从机 1	从机 2
SADDR	1010 0100	1010 0111
SADEN（为 0 的位被忽略）	1111 1010	1111 1001
实际从机地址	1010 0x0x	1010 0xx1
广播地址（SADDR 或 SADEN）	1111 111x	1111 1111

从机 1 和从机 2 给定地址的最低位是不同的。从机 1 忽略了最低位，而从机 2 的最低位是 1。因此只与从机 1 通讯时，主机必须发送最低位为 0 的地址（10100000）。类似地，从机 1 的第 1 位为 0，从机 2 的第 1 位被忽略。因此，只与从机 2 通讯时，主机必须发送第 1 位为 1 的地址（10100011）。如果主机希望同时与两从机通讯，则第 0 位为 1，第 1 位为 0，第 2 位被两从机都忽略，此时有两个不同的地址用于选定两个从机（1010 0001 和 1010 0101）。

主机可以通过广播地址与所有从机同时通讯。这个地址等于 SADDR 和 SADEN 的逻辑或，结果中的 0 表示该位被忽略。多数情况下，广播地址为 0xFFh，该地址可被所有从机应答。

系统复位后，SADDR 和 SADEN 两个寄存器初始化为 0，这两个结果设定了给定地址和广播地址为 XXXXXXXX（所有位都被忽略）。这有效地去除了多处机通讯的特性，禁止了自动寻址方式。这样的 EUART 将对任何地址都产生应答，兼容了不支持自动地址识别的 8051 控制器。用户可以按照上面提到的方法实现软件识别地址的多机通讯。

10.5 帧出错检测

当寄存器 T2CR 中的 SSTAT 位为逻辑 1 时，帧出错检测功能才有效。2 个错误标志位被置 1 后，只能通过软件清零，尽管后续接收的帧没有任何错误也不会自动清零。

注：SSTAT=1 时是访问状态位 （FE，RXOV），SSTAT=0 时是访问方式选择位 （SM0，SM1 和 SM2） 。

接收溢出

在接收缓冲器中数据未被读取之前，RI 被清 0，此时如果又有新的数据存入接收缓冲器，则接收溢出位 RXOV 位置 1。如果发生了接收溢出，接收缓冲器中原来的数据将丢失。

帧出错

如果检测到一个无效（低）停止位，则帧出错位 FE 置 1。

暂停检测

当连续检测到 11 个位都为低电平位时，则认为检测到一个暂停。由于暂停条件同样满足帧错误条件，因此检测到暂停时也会报告帧错误。一旦检测到暂停条件，UART 将进入空闲状态并一直保持，直至接收到有效停止位（RXD 引脚上出现上升沿）。

10.6 EUART 相关寄存器

串口控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCON	SM0/FE	SM1/RXOV	SM2	REN	TB8	RB8	TI	RI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:6] **SM[0:1]** – EUART 方式控制位（SSTAT=0）

SM[0:1]	EUART 工作方式
00	方式 0：同步方式，固定波特率
01	方式 1：8 位异步方式，可变波特率
10	方式 2：9 位异步方式，固定波特率
11	方式 3：9 位异步方式，可变波特率

BIT[5] **SM2** – EUART 功能设定位（SSTAT=0）

SM2	方式 0	方式 1	方式 2/3
0	波特率=F_{CPU}/12	禁止停止位确认检验，停止位将置 RI 为 1 产生中断	任何字节均会置 RI 为 1 产生中断
1	波特率=F_{CPU}/4	允许停止位确认检验，只有有效的停止位“1”才能置 RI 为 1 产生中断	只有寻址字节（第 9 位=1）能置 RI 为 1 产生中断

BIT[7] **FE** – 帧出错标志位（SSTAT=1）

0： 无帧出错，由软件清 0；

1： 发生帧出错，由硬件置 1；

BIT[6] **RXOV** – 接收溢出标志位（SSTAT=1）
0： 无接收溢出，由软件清 0；
1： 接收溢出，由硬件置 1；

BIT[4] **REN** – 接收器允许位
0： 禁止接收；
1： 允许接收；

BIT[3] **TB8** – 方式 2/3 时发送数据的第 9 位
0： 方式 2/3 发送数据的第 9 位为 0；
1： 方式 2/3 发送数据的第 9 位为 1；

BIT[2] **RB8** – 方式 1/2/3 时接收数据的第 9 位（停止位或数据位）
0： 方式 1/2/3 时接收数据的第 9 位为 0；
1： 方式 1/2/3 时接收数据的第 9 位为 1；

BIT[1] **TI** – 发送中断标志位
0： 由软件清 0；
1： 在方式 0 的第 8 位最后，或在其他方式的停止位开始，由硬件置 1；

BIT[0] **RI** – 接收中断标志位
0： 由软件清 0；
1： 在方式 0 的第 8 位最后，或在其他方式的停止位开始，由硬件置 1；

串口缓存寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SBUF	SBUF7	SBUF6	SBUF5	SBUF4	SBUF3	SBUF2	SBUF1	SBUF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **SBUF[7:0]** – EUART 数据缓存，SBUF 访问两个寄存器：1 个移位寄存器和 1 个接收锁存寄存器。SBUF 的写入将发送字节到移位寄存器中，然后开始从端口发送；SBUF 的读取将返回接收锁存寄存器中的内容

从机地址寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SADDR	SADDR7	SADDR6	SADDR5	SADDR4	SADDR3	SADDR2	SADDR1	SADDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **SADDR[7:0]** – EUART 的从机地址

地址掩码寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SADEN	SADEN7	SADEN6	SADEN5	SADEN4	SADEN3	SADEN2	SADEN1	SADEN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **SADENn** – SADDRn 检验控制位（n=7-0）

0： 忽略 SADDRn 位；

1： 检验 SADDRn 位是否对应接收地址；

11 EEPROM

11.1 EEPROM 概述

芯片内置 64 字节的 EEPROM 数据存储器，支持用户程序在带电工作中实时地读出或写入数据。读写数据时需通过控制寄存器 EECR、数据寄存器 EEDR、地址寄存器 EEAR 和保护寄存器 EEMASK 进行。

EECR 中 EETRIG 位为操作启动标志位，置 1 启动读写操作，完成后自动清 0；EERW 位为读写命令位，为 0 表示读数据、为 1 表示写数据：读数据操作将从 EEAR 对应的 EEPROM 地址中读出数据，保存在 EEDR 中，写数据操作将 EEDR 中的数据写入 EEAR 对应的 EEPROM 地址中。

为防止误触发 EEPROM 读写操作，寄存器 EEMASK 需先写入 5AH 再立即写入 A5H，EETRIG 位才能置 1，否则无法写“1”，且 2 个指令周期后 EEMASK 将自动清除。

芯片仅支持单字节数据的读写操作，不支持连续地址读写功能，每次都必须通过 EEAR 设置将要访问的 EEPROM 数据的 6 位地址后，才能进行读写操作。

当启动 EEPROM 读写操作后，CPU 将暂停在当前指令，只有等 EEPROM 读写操作完成后，才能继续执行下一条指令。在读写 EEPROM 时需屏蔽中断，否则会因系统响应中断而导致读写错误。

11.2 EEPROM 相关寄存器

EEPROM 控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EECR	EETRIG	EERW	-	-	-	-	-	-
R/W	R/W	R/W	-	-	-	-	-	-
初始值	0	0	-	-	-	-	-	-

BIT[7] **EETRIG** – EEPROM 读写启动控制位

0： 未进行 EEPROM 读写操作，或 EEPROM 读写操作完成后硬件自动清 0；

1： 启动 EEPROM 读写操作；

BIT[6] **EERW** – EEPROM 读写命令位

0： 从 EEPROM 中读出数据；

1： 向 EEPROM 中写入数据；

EEPROM 保护寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EEMASK	EEMASK7	EEMASK6	EEMASK5	EEMASK4	EEMASK3	EEMASK2	EEMASK1	EEMASK0
R/W	W	W	W	W	W	W	W	W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **EEMASK[7:0]** – EEPROM 操作保护位，需先写 5AH 再立即写 A5H，EETRIG 才能置 1

EEPROM 地址寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EEAR	-	-	EEA5	EEA4	EEA3	EEA2	EEA1	EEA0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	0	0	0	0	0	0

BIT[5:0] **EEA[5:0]** – EEPROM 读写操作的 6 位地址

EEPROM 数据寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EEDR	EED7	EED6	EED5	EED4	EED3	EED2	EED1	EED0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **EED[7:0]** – EEPROM 读写操作的 8 位数据

11.3 EEPROM 操作示例

例：向地址为 10H 的 EEPROM 单元写入数据 55H

```
MOVAI   10H
MOVRA   EEAR        ；将地址 10H 写入 EEAR
MOVAI   55H
MOVRA   EEDR        ；将数据 55H 写入 EEDR
BCLR    GIE         ；屏蔽中断
MOVAI   5AH
MOVRA   EEMASK      ；使能 EE 操作，第 1 步
MOVAI   A5H
MOVRA   EEMASK      ；使能 EE 操作，第 2 步
MOVAI   C0H
MOVRA   EECR        ；启动 EE 写操作，将数据 55H 写入 EEPROM 地址 10H 中
NOP                 ；防止时序错误，CPU 必须冗余 1 个指令周期
BSET    GIE         ；允许中断
```

例：从地址为 10H 的 EEPROM 单元读出数据（数据存放在 EEDR 中）

```
MOVAI     10H
MOVRA     EEAR          ；将地址 10H 写入 EEAR
BCLR      GIE           ；屏蔽中断
MOVAI     5AH
MOVRA     EEMASK        ；使能 EE 操作，第 1 步
MOVAI     A5H
MOVRA     EEMASK        ；使能 EE 操作，第 2 步
MOVAI     80H
MOVRA     EECR          ；启动 EE 写操作，从 EEPROM 地址 10H 中读出数据
NOP                     ；防止时序错误，CPU 必须冗余 1 个指令周期
BSET      GIE           ；允许中断
```

12 中断

中断有外部中断（INT0~INT3）、定时器中断（T0~T3）、ADC 转换中断、键盘中断、IIC 通讯中断和 UART 通讯中断等。可通过寄存器控制位 GIE 屏蔽所有中断。

中断响应过程如下：

- ✧ 当发生中断请求时，CPU 将相关下一条要执行的指令的地址压栈保存（累加器 A 和状态寄存器 PFLAG 需要软件保护），对中断屏蔽位 GIE 清 0，禁止中断响应。与复位不同，硬件中断不停止当前指令的执行，而是暂时挂起中断直到当前指令执行完成。
- ✧ CPU 执行中断时，程序跳到中断向量地址开始执行中断服务程序，中断服务程序应先保存累加器 A 和状态寄存器 PFLAG，然后判断是哪一个中断响应。
- ✧ 中断服务程序执行完中断内容后应恢复累加器 A 和状态寄存器 PFLAG，然后执行 RETIE 返回主程序。此时芯片将从堆栈取出 PC 值，然后从中断发生时当前指令的下一条指令继续执行。

本芯片的中断向量地址是 0008H。

12.1 外部中断

本芯片有 4 路外部中断源，INT0/INT1 可设置为上升沿触发、下降沿触发或电平变化触发等触发方式，INT2/INT3 固定为下降沿触发。当外部中断触发时，外部中断标志（INT0IF、INT1IF、INT2IF、INT3IF）将被置 1，若中断总使能位 GIE 为 1 且外部中断使能位（INT0IE、INT1IE、INT2IE、INT3IE）为 1，则产生外部中断。

12.2 定时器中断

定时器 T0、T1、T2、T3 在计数溢出时中断标志（T0IF、T1IF、T2IF、T3IF）将被置 1，若中断总使能位 GIE 为 1 且定时器中断使能位（T0IE、T1IE、T2IE、T3IE）为 1，则产生定时器中断。

12.3 ADC 中断

ADC 转换完成后中断标志（ADIF）将被置 1，若中断总使能位 GIE 为 1 且 ADC 中断使能位（ADIE）为 1，则产生 ADC 中断。

12.4 键盘中断

本芯片有 8 路键盘中断源，可通过 KBCR 寄存器单独屏蔽，任意一路未被屏蔽的中断源的电平发生变化时，触发键盘中断，键盘中断标志（KBIF）将被置 1，若中断总使能位 GIE 为 1 且键盘中断使能位（KBIE）为 1 ，则产生键盘中断。

键盘中断控制寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
KBCR	KBCR7	KBCR6	KBCR5	KBCR4	KBCR3	KBCR2	KBCR1	KBCR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **KBCRn** – P1n 口键盘中断使能位（n=7-0）

0： 屏蔽端口键盘中断功能；

1： 使能端口键盘中断功能；

12.5 IIC 通讯中断

芯片在进行 IIC 通讯时会产生中断，可选择两种中断触发模式，触发后中断标志将被置 1，并需要软件清 0 以正确响应下一次中断：

IICIMS=0 时，当发送完数据并接收完从机应答信号 ACK/NACK、接收完从机数据并发送完应答位信号 ACK/NACK、或发送完 STOP 位信号后，中断标志（IICIF）将被置 1，若中断总使能位 GIE 为 1 且 IIC 中断使能位（IICIE）为 1，则产生 IIC 中断；

IICIMS=1 时，当发送完 START 或 STOP 位信号、发送完数据并接收完从机应答信号 ACK/NACK、接收完从机 8 位数据、或发送完应答位信号 ACK/NACK 后，中断标志（IICIF）将被置 1，若中断总使能位 GIE 为 1 且 IIC 中断使能位（IICIE）为 1，则产生 IIC 中断。

12.6 UART 通讯中断

芯片在进行 UART 通讯时会产生中断，当 TI、RI 被硬件置 1 时将触发中断，中断标志（UARTIF）将被置 1，若中断总使能位 GIE 为 1 且 UART 中断使能位（UARTIE）为 1，则产生 UART 中断。

12.7 中断相关寄存器

中断使能寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTE	UARTIE	ADIE	IICIE	KBIE	INT1IE	INT0IE	T1IE	T0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] **UARTIE** – UART 中断使能位

0： 屏蔽 UART 中断；

1： 使能 UART 中断；

BIT[6] **ADIE** – ADC 中断使能位
0： 屏蔽 ADC 中断；
1： 使能 ADC 中断；

BIT[5] **IICIE** – IIC 中断使能位
0： 屏蔽 IIC 中断；
1： 使能 IIC 中断；

BIT[4] **KBIE** – 键盘中断使能位
0： 屏蔽键盘中断；
1： 使能键盘中断；

BIT[3] **INT1IE** – INT1 中断使能位
0： 屏蔽 INT1 中断；
1： 使能 INT1 中断；

BIT[2] **INT0IE** – INT0 中断使能位
0： 屏蔽 INT0 中断；
1： 使能 INT0 中断；

BIT[1] **T1IE** – 定时器 T1 中断使能位
0： 屏蔽定时器 T1 中断；
1： 使能定时器 T1 中断；

BIT[0] **T0IE** – 定时器 T0 中断使能位
0： 屏蔽定时器 T0 中断；
1： 使能定时器 T0 中断；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTE1	-	-	-	-	INT3IE	INT2IE	T3IE	T2IE
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3] **INT3IE** – INT3 中断使能位
0： 屏蔽 INT3 中断；
1： 使能 INT3 中断；

BIT[2] **INT2IE** – INT2 中断使能位
0： 屏蔽 INT2 中断；
1： 使能 INT2 中断；

BIT[1] **T3IE** – 定时器 T3 中断使能位
0： 屏蔽定时器 T3 中断；
1： 使能定时器 T3 中断；

BIT[0] **T2IE** – 定时器 T2 中断使能位
0： 屏蔽定时器 T2 中断；
1： 使能定时器 T2 中断；

中断标志寄存器

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTF	UARTIF	ADIF	IICIF	KBIF	INT1IF	INT0IF	T1IF	T0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] **UARTIF** –UART 中断标志位
0： 未发生 UART 中断；
1： 发生 UART 中断，需软件清零；

BIT[6] **ADIF** – ADC 中断标志位
0： 未发生 ADC 中断；
1： 发生 ADC 中断，需软件清零；

BIT[5] **IICIF** – IIC 中断标志位
0： 未发生 IIC 中断；
1： 发生 IIC 中断，需软件清零；

BIT[4] **KBIF** – 键盘中断标志位
0： 未发生键盘中断；
1： 发生键盘中断，需软件清零；

BIT[3] **INT1IF** – INT1 中断标志位
0： 未发生 INT1 中断；
1： 发生 INT1 中断，需软件清零；

BIT[2] **INT0IF** – INT0 中断标志位
0： 未发生 INT0 中断；
1： 发生 INT0 中断，需软件清零；

BIT[1] **T1IF** – 定时器 T1 中断标志位
0： 未发生定时器 T1 中断；
1： 发生定时器 T1 中断，需软件清零；

BIT[0] **T0IF** – 定时器 T0 中断标志位
0： 未发生定时器 T0 中断；
1： 发生定时器 T0 中断，需软件清零；

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTF1	-	-	-	-	INT3IF	INT2IF	T3IF	T2IF
R/W	-	-	-	-	R/W	R/W	R/W	R/W
初始值	-	-	-	-	0	0	0	0

BIT[3] **INT3IF** – INT3 中断标志位

0： 未发生 INT3 中断；

1： 发生 INT3 中断，需软件清零；

BIT[2] **INT2IF** – INT2 中断标志位

0： 未发生 INT2 中断；

1： 发生 INT2 中断，需软件清零；

BIT[1] **T3IF** – 定时器 T3 中断标志位

0： 未发生定时器 T3 中断；

1： 发生定时器 T3 中断，需软件清零；

BIT[0] **T2IF** – 定时器 T2 中断标志位

0： 未发生定时器 T2 中断；

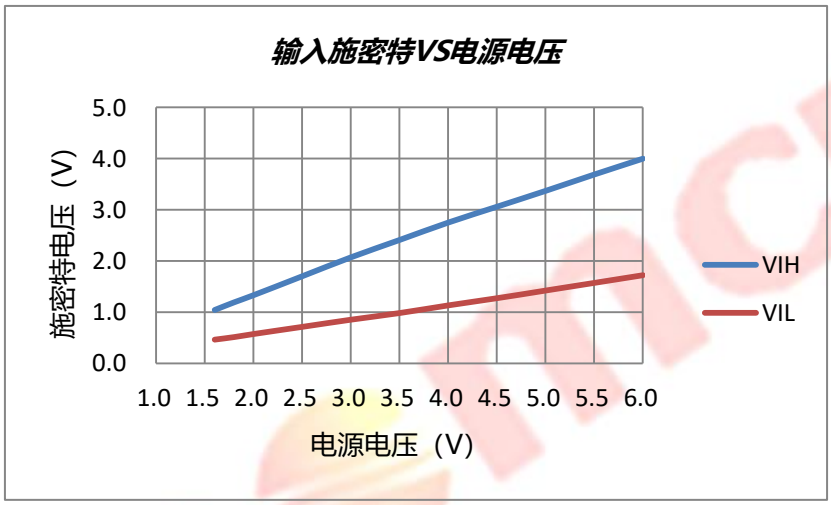
1： 发生定时器 T2 中断，需软件清零；

13 特性曲线

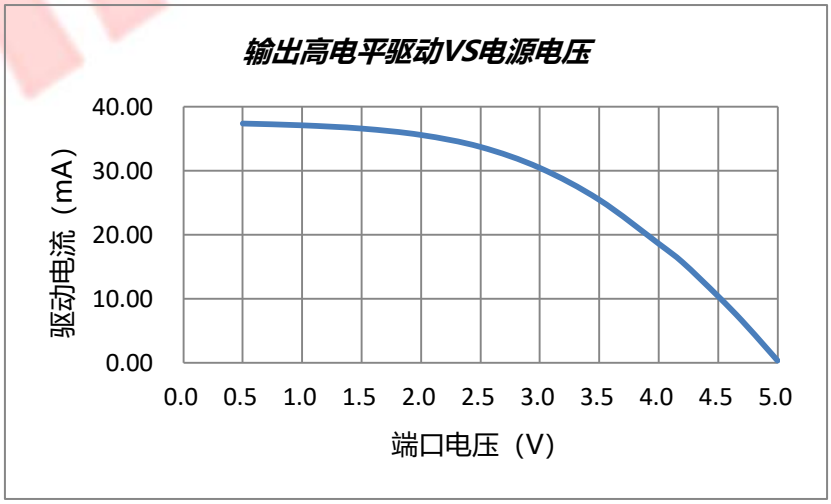
注：本章节所列特性曲线图为抽样实测数据，仅作为应用参考，部分数据因生产工艺偏差，可能与实际芯片不符；为保证芯片能正常工作，请确保其工作条件符合电气特性参数说明。

13.1 I/O 特性

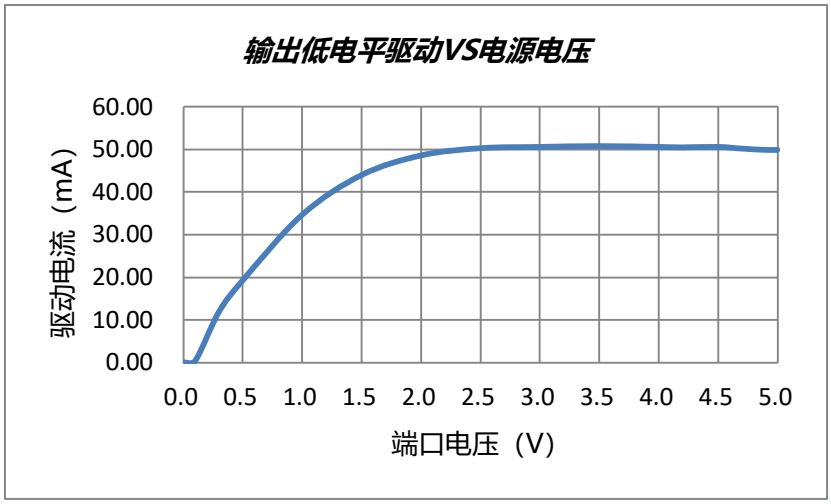
I/O 输入施密特电压 VS 电源电压



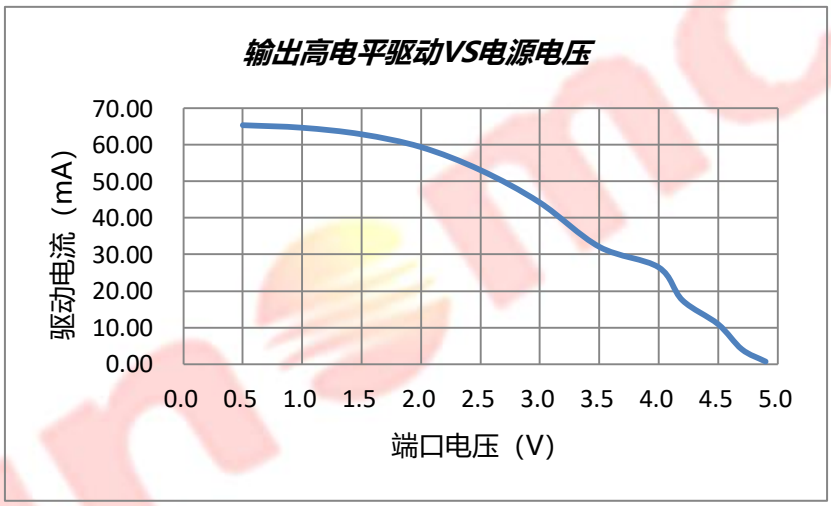
P0 口输出高电平驱动电流 VS 端口电压



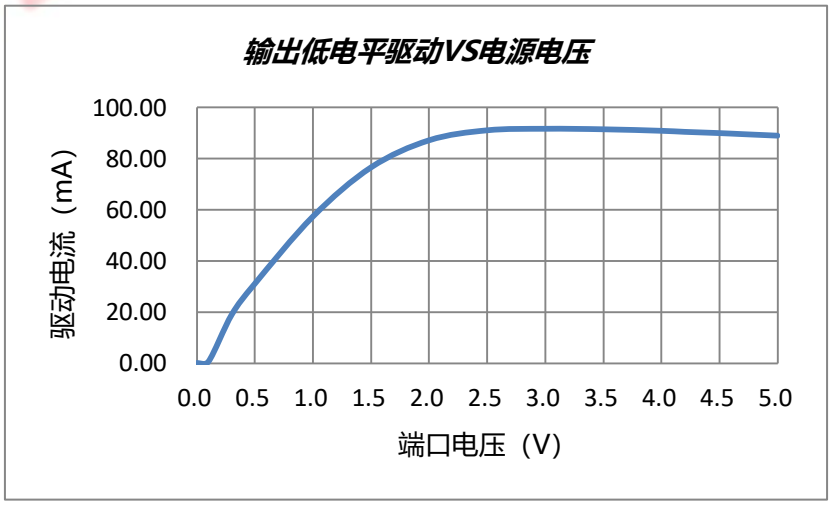
P0 口输出低电平驱动电流 VS 端口电压



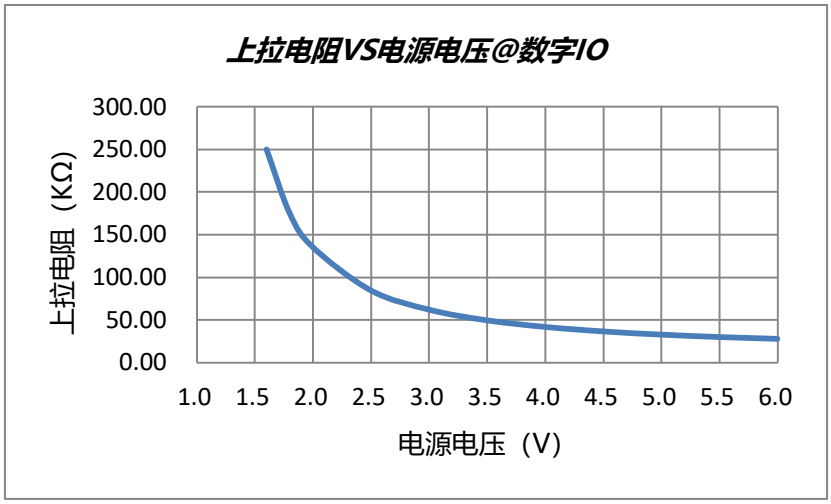
P1 口输出高电平驱动电流 VS 端口电压



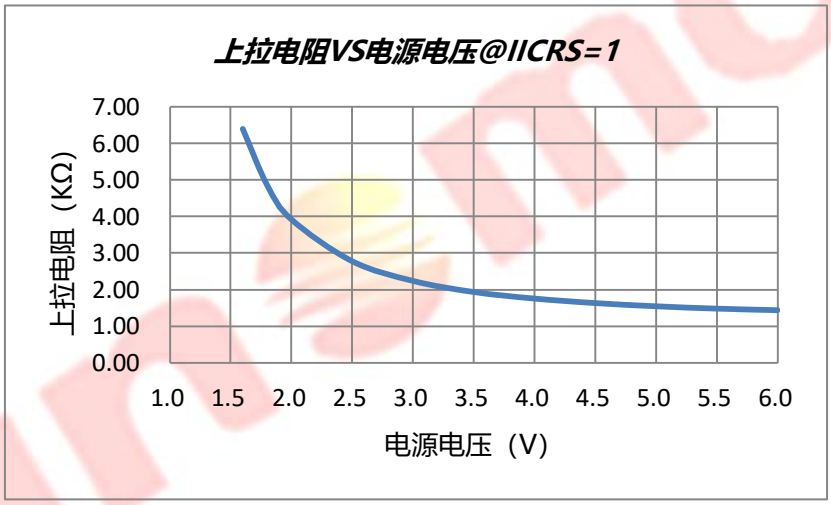
P1 口输出低电平驱动电流 VS 端口电压

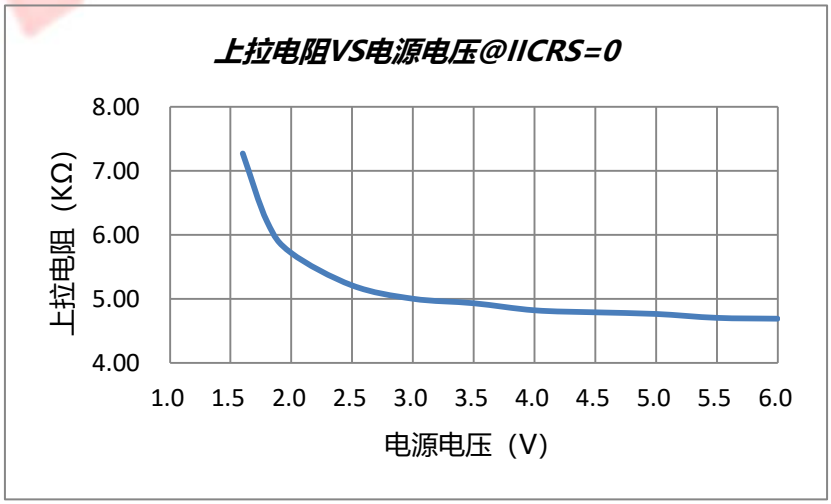


数字 I/O 上拉电阻 VS 电源电压

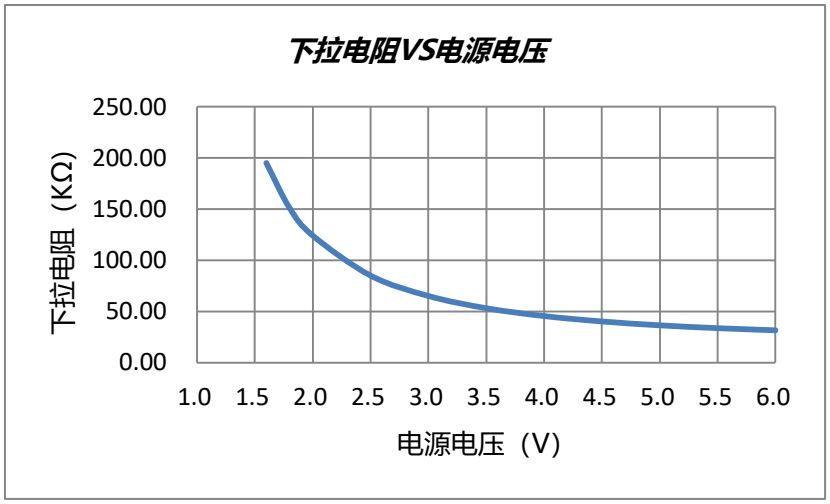


通讯 I/O 上拉电阻 VS 电源电压



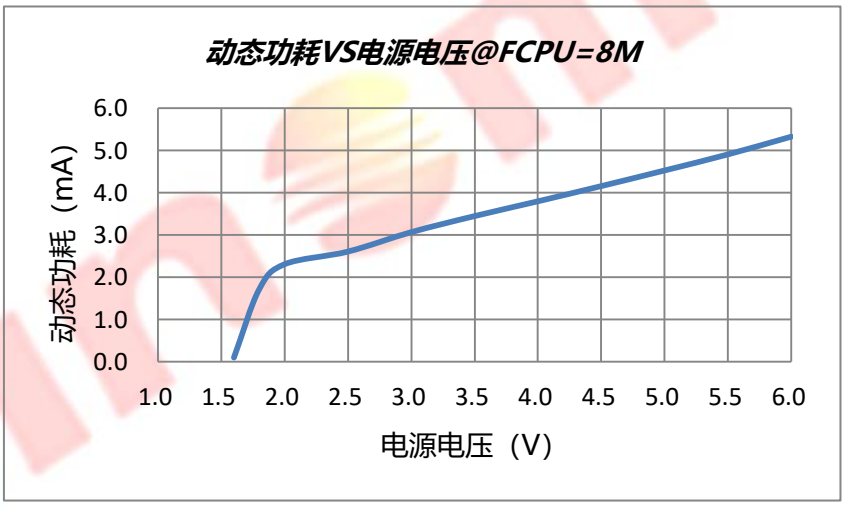


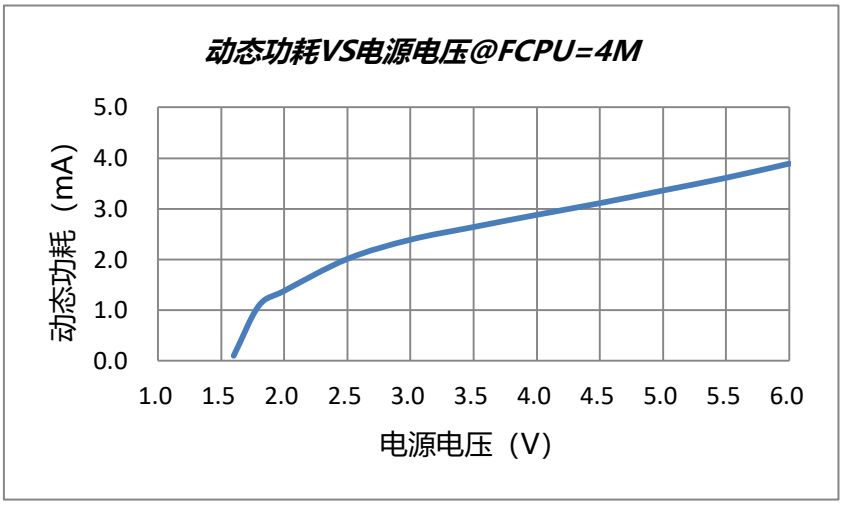
数字 I/O 下拉电阻 VS 电源电压

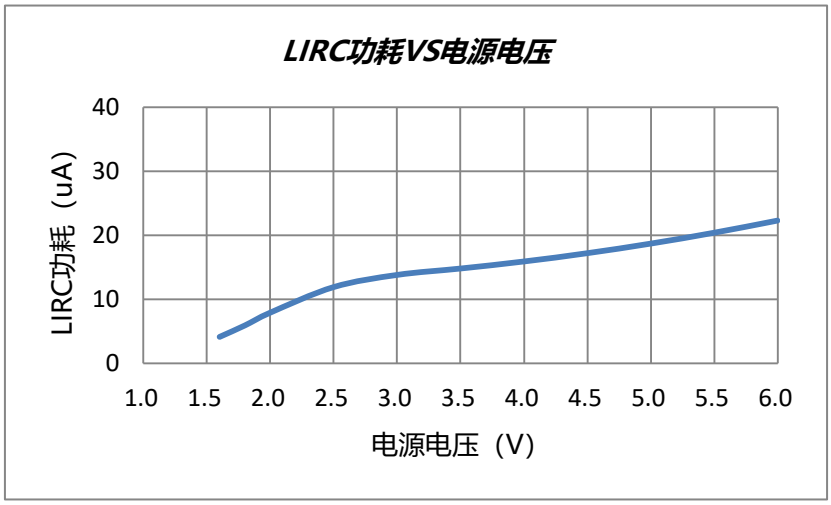


13.2 功耗特性

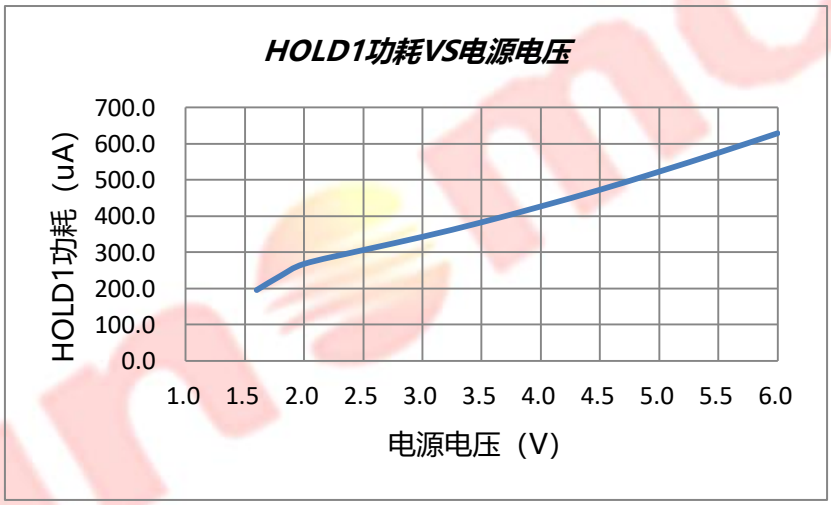
动态功耗 VS 电源电压

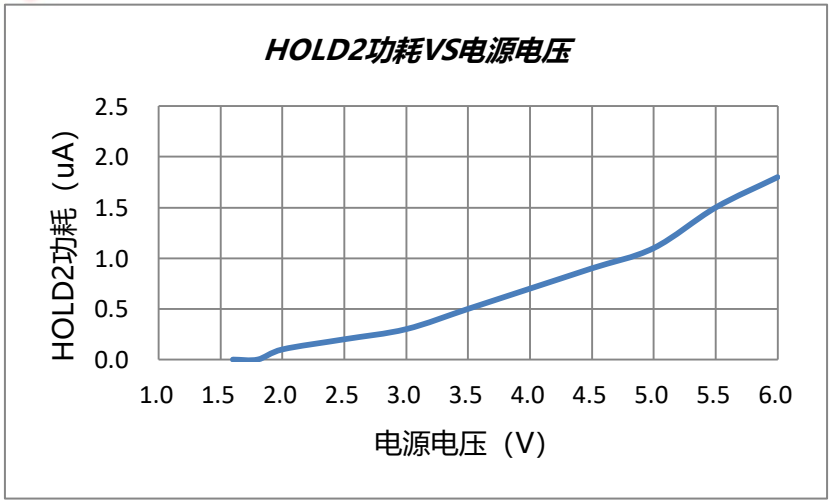




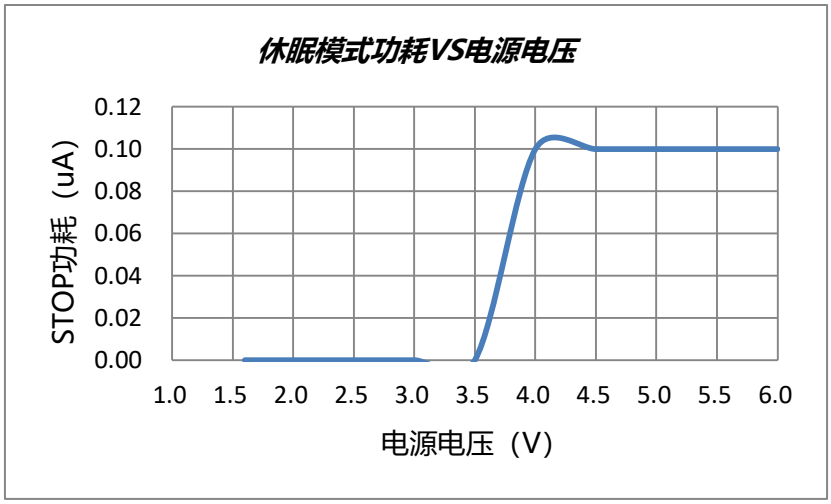


HOLD 模式功耗 VS 电源电压



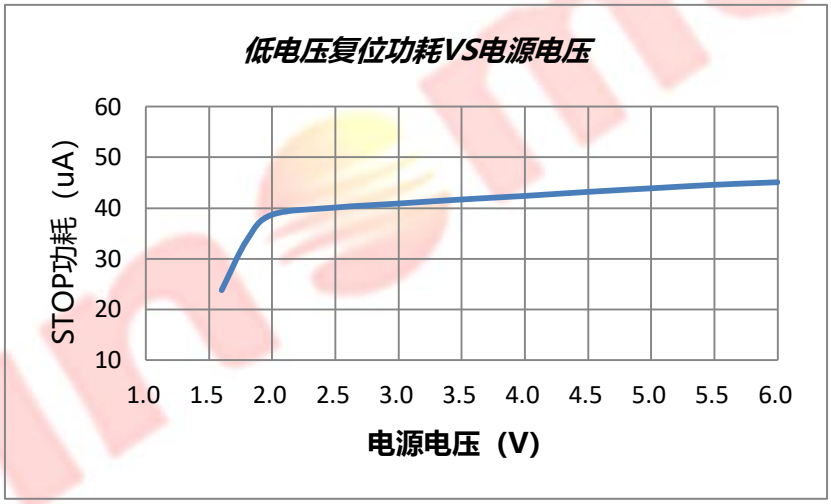


休眠模式功耗 VS 电源电压

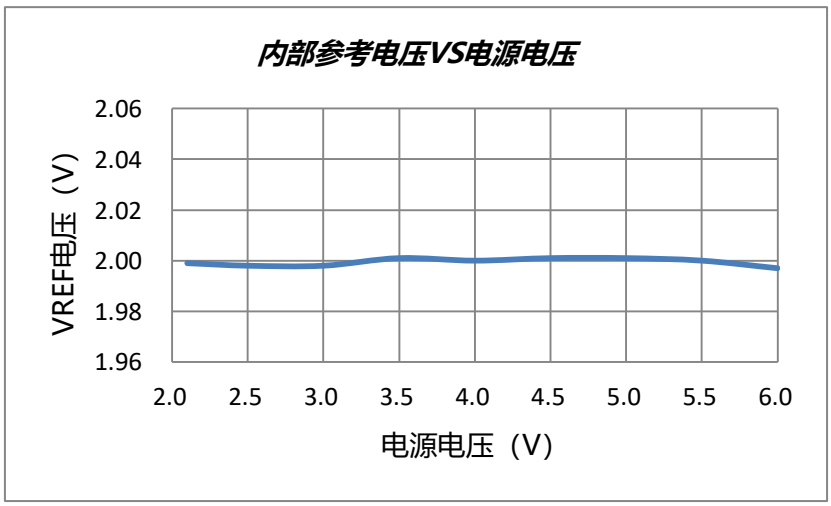


13.3 模拟电路特性

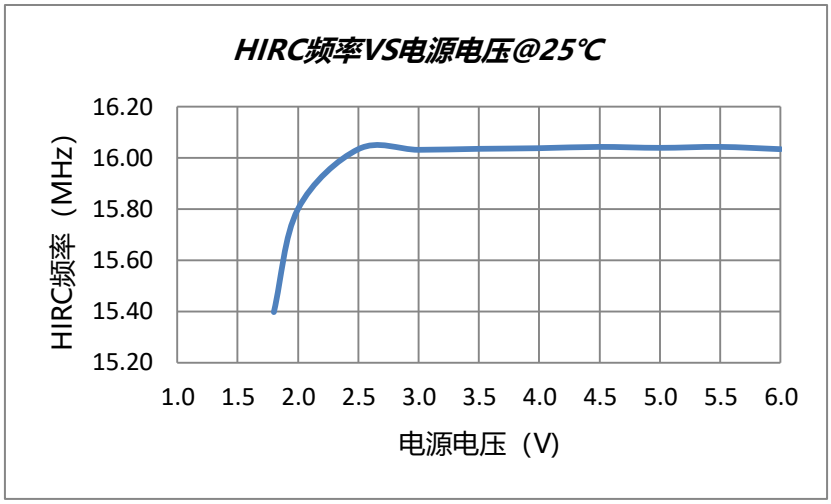
LVR 功耗 VS 电源电压



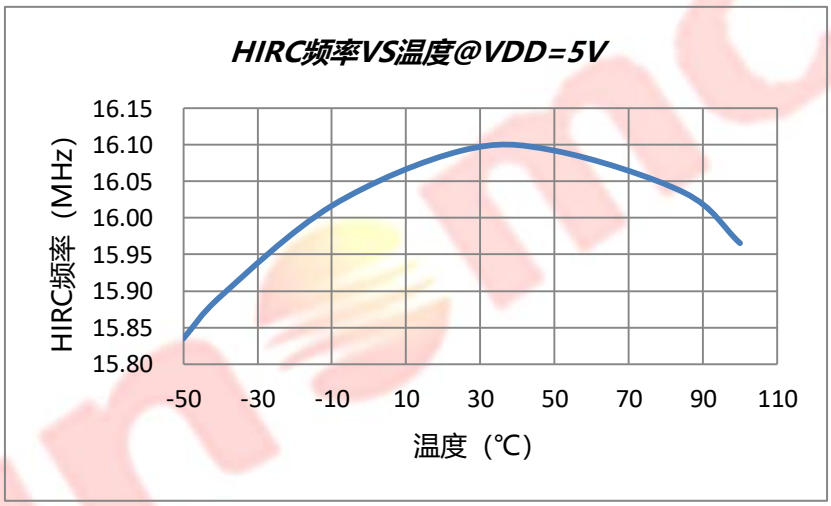
内部参考电压 VS 电源电压



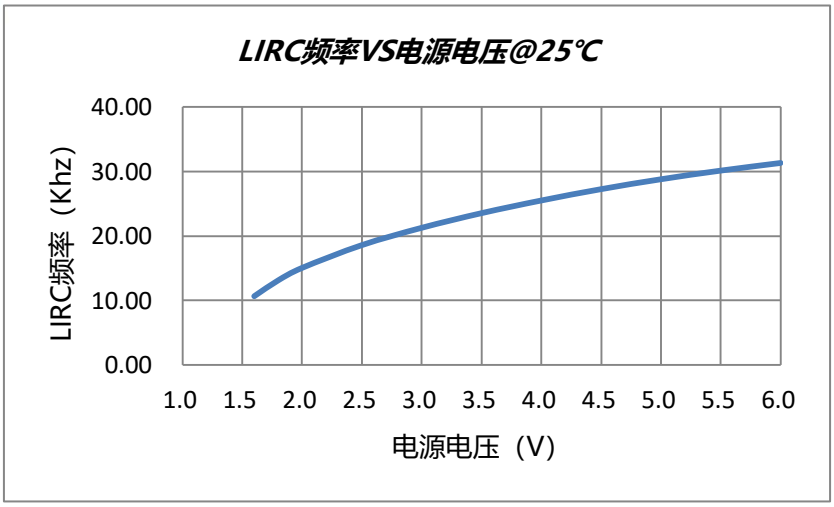
HIRC 频率 VS 电源电压



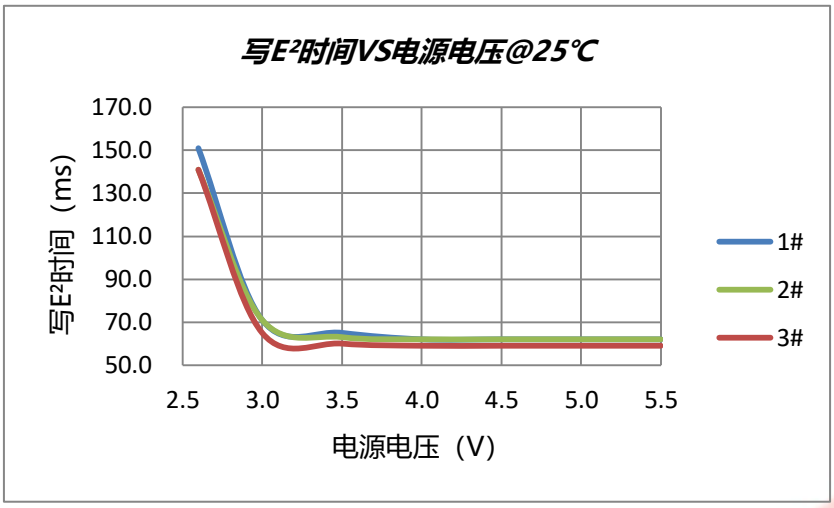
HIRC 频率 VS 温度



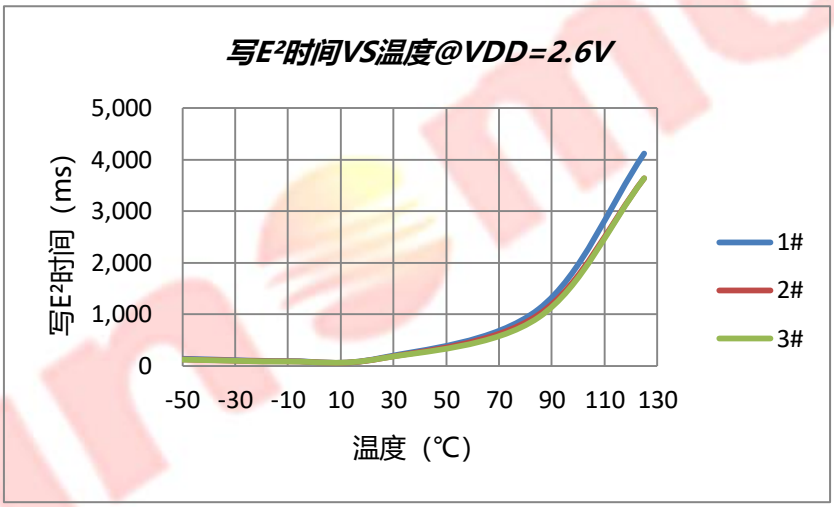
LIRC 频率 VS 电源电压

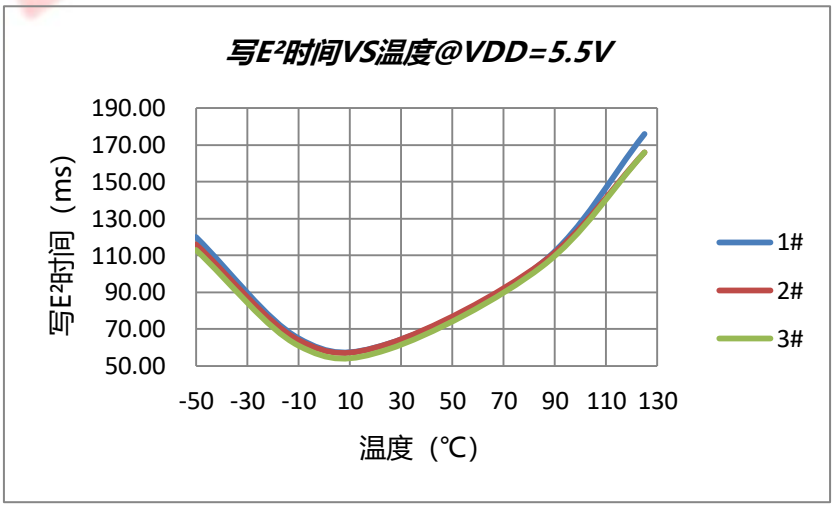


64 字节 EEPROM 擦写时间 VS 电源电压



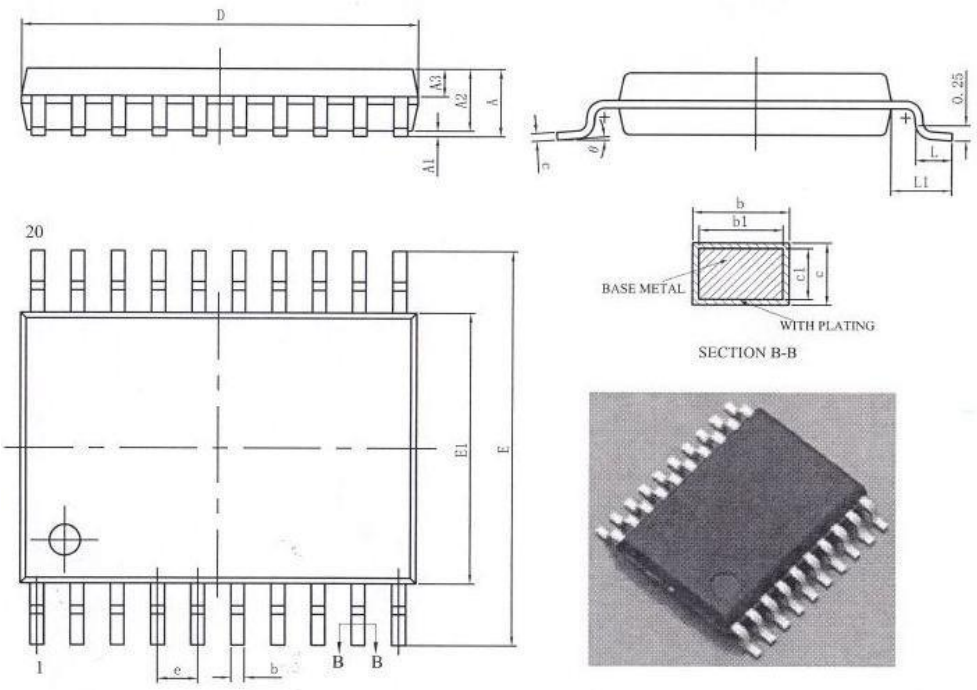
64 字节 EEPROM 擦写时间 VS 温度





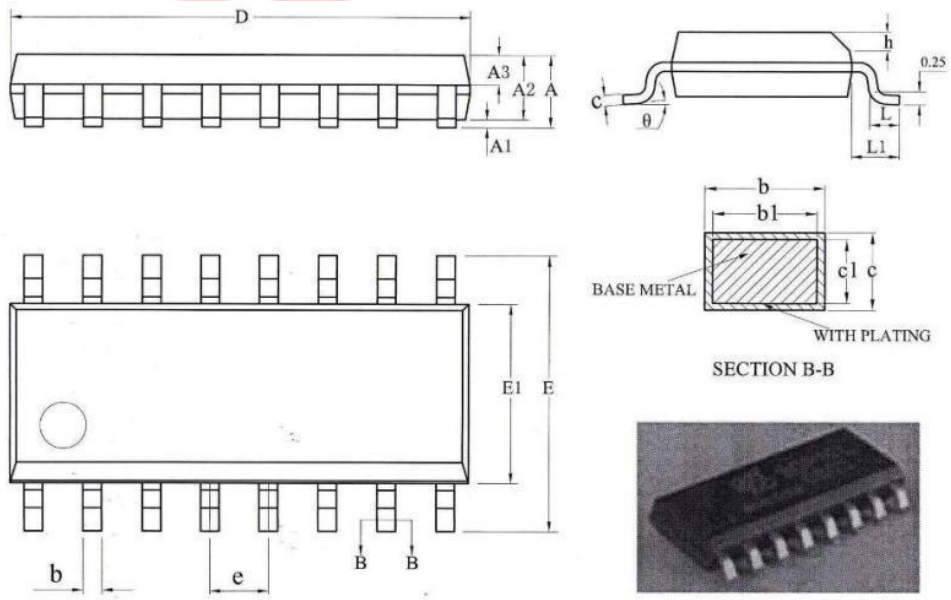
14 封装尺寸

14.1 TSSOP20



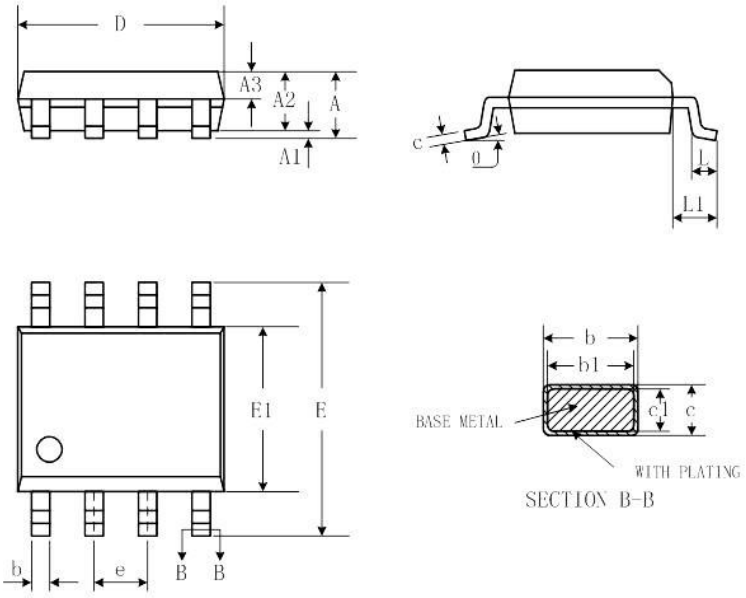
SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.20
A1	0.05	—	0.15
A2	0.80	1.00	1.05
A3	0.39	0.44	0.49
b	0.20	—	0.29
b1	0.19	0.22	0.25
c	0.13	—	0.18
c1	0.12	0.13	0.14
D	6.40	6.50	6.60
E1	4.30	4.40	4.50
E	6.20	6.40	6.60
e	0.65BSC		
L	0.45	0.60	0.75
L1	1.00BSC		
θ	0	—	8°

14.2 SOP16



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.75
A1	0.10	—	0.225
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.39	—	0.47
b1	0.38	0.41	0.44
c	0.20	—	0.24
c1	0.19	0.20	0.21
D	9.80	9.90	10.00
E	5.80	6.00	6.20
E1	3.80	3.90	4.00
e	1.27BSC		
h	0.25	—	0.50
L	0.50	—	0.80
L1	1.05REF		
θ	0	—	8°

14.3 SOP8



SYMBOL	MILLIMETER		
	MIN	TYP	MAX
A	-	-	1.77
A1	0.08	0.18	0.28
A2	1.20	1.40	1.60
A3	0.55	0.65	0.75
b	0.39	-	0.48
b1	0.38	0.41	0.43
c	0.21	-	0.26
c1	0.19	0.20	0.21
D	4.70	4.90	5.10
E	5.80	6.00	6.20
E1	3.70	3.90	4.10
e	1.27BSC		
L	0.50	0.65	0.80
L1	1.05BSC		
θ	0	-	8°

15 修订记录

版本	修订日期	修订内容
V1.0	2019-08-14	初版发布;