可编程光电式烟雾检测电路

本资料适用范围：CS2132ALO、CS2132BLO、CS2132CLO、CS2132DLO、CS2132MLO

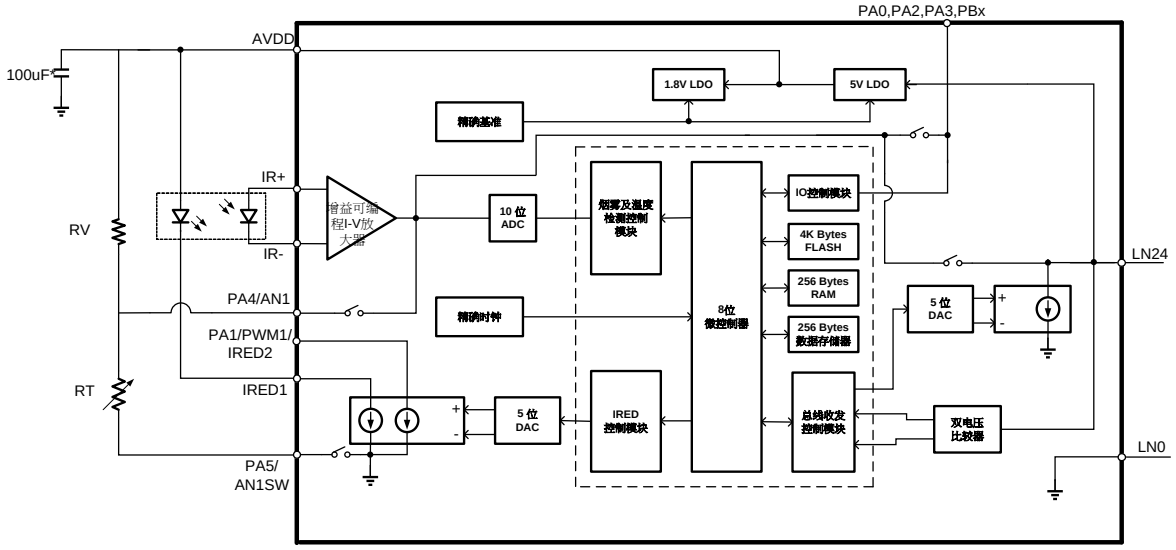
1、概述

CS2132 是一款只需要最少的外部器件，便可以建立起可寻址烟雾检测器的可编程平台，支持光电式烟雾探测器，用于公共场所，商业场所以及居民住宅区中可寻址烟雾检测器的建立。其特点如下：

- 用于联网式可寻址的光电式烟雾检测系统
- 采用 2-wire 总线操作，8~32V 的工作电压范围
- 低静态电流
- 内置 8 位微控制器
- 256B 的 RAM 空间
- 4KB 的程序存储空间
- 256B 的数据存储空间
- 系统时钟 8/4/2/1/MHz、400KHz 可选
- 总线采样时钟 8/4/2/1MHz、400KHz 可选
- 两个 16 位通用定时器
- 总线双电压比较器（绝对电压比较）
- 高精度内置时钟振荡器（8MHz、400KHz）
- 内置增益可编程的光电放大器
- 最大值 115mA 的可编程总线回码电流
- 多路 10 位的 ADC
- 温度检测功能、总线电压检测功能
- 支持双路红外发射驱动，驱动电流软件可编程
- 支持最多 4 路 PWM 输出
- 支持 UART 接口通讯
- 封装形式：SOP16（CS2132ALO、CS2132BLO、CS2132MLO）
  SOP20（CS2132CLO）
  SOP14（CS2132DLO）

2、功能框图与引脚说明

2. 1、功能框图



2. 2、功能描述

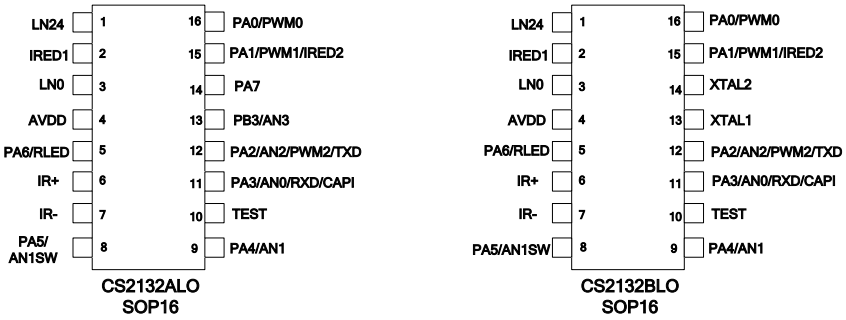
对于外接光电式烟雾传感器的 CS2132 电路，电路可输出强度和持续时间可编程的电流脉冲直接驱动红外发射管，同时具有增益可编程的光电积分电路用以接收光电二极管产生的光电流。光电积分电路中的放大器具有误差校准和带通滤波器特性，并通过高精度 ADC 得到很好的信号处理精度。通过内置的用户可编程的微处理器，可以控制发射脉冲的强度及宽度、接收信号时的增益等，可实现对光电信号的灵活配置。

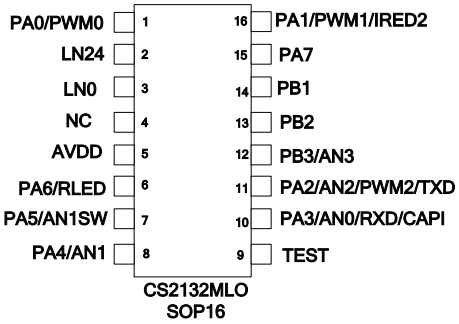
电路包含 8 位 MCU 核作为控制核心，包含 256Byte RAM，4KByte 的程序存储空间和 256Byte 的数据存储空间。

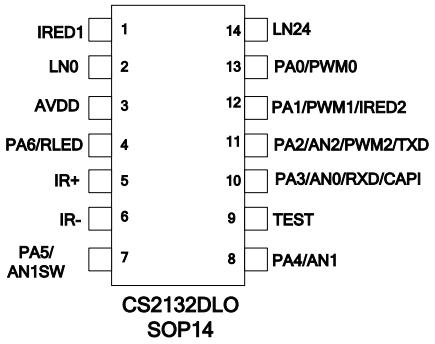
电路实现完全两线式总线的供电和通讯功能的集成，电路一根总线上可支持多个检测器相连。内部集成两个电压比较器，可通过寄存器对总线电压设定最多两个阈值比较电平，从而实现对总线上最多三个电平状态的监控。可通过寄存器设定最大 115mA 的总线回码电流。

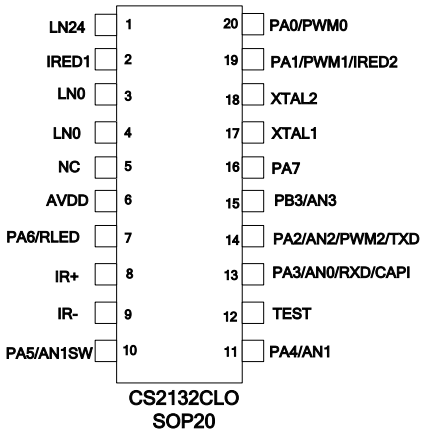
电路可同时实现温感检测功能、总线电压检测功能、支持最多 4 路 PWM 输出。

2. 3、引脚排列图









2. 4、引脚说明

序 号	引 脚 名 称	复 用 功 能	说 明
1	IRED1		红外发光二极管驱动的驱动端口 1
2	LN0		总线，地
3	AVDD		模拟电源，外接 100μF 电容，用于储能和稳压滤波。
4	PA6/RLED	RLED	PWM3 输出口
		PA6	数字输出端口
5	IR+		连接光电二极管的阳极
6	IR-		连接光电二极管的阴极
7	PA5/AN1SW	PA5	通用 IO 口
		AN1SW	外部温度检测信号的驱动口
8	PA4/AN1	PA4	通用 IO 口
		AN1	ADC 通道 1
9	TEST		测试/调试使能端口

转下页

接上页

序 号	引 脚 名 称	复 用 功 能	说 明
10	PA3/AN0/RXD/CAPI	PA3	通用 IO 口
		AN0	ADC 通道 0
		RXD	串口通讯口
		CAPI	捕获功能输入口
11	PA2/AN2/PWM2/TXD	PA2	通用 IO 口
		AN2	ADC 通道 2
		PWM2	PWM2 输出口（开漏输出）
		TXD	串口通讯口
12	PA7		通用 IO 口
13	PB1		通用 IO 口
14	PB2		通用 IO 口
15	PB3/AN3	PB3	通用 IO 口
		AN3	ADC 通道 3
16	XTAL1		外接晶振
17	XTAL2		外接晶振
18	PA1/PWM1/IRED2	PA1	通用 IO 口
		PWM1	PWM1 输出口
		IRED2	红外发光二极管驱动的驱动端口 2
19	PA0/PWM0	PA0	通用 IO 口
		PWM0	PWM0 输出口（开漏输出）
20	LN24		总线，电源

3、电特性

3. 1、极限参数

除非另有规定，T_{amb}=25℃

参 数 名 称	符 号	额 定 值	单 位
电源	V_{LN24}	-0.3~48	V
存储温度	T_{stg}	-55~90	℃
工作环境温度	T_{amb}	-25~75	℃
数字 IO 电压	V_{DPIN}	-0.3~ V_{AVDD} +0.3	V
模拟 IO 电压	V_{APIN}	-0.3~ V_{AVDD} +0.3	V

3. 2、电特性

除非另有规定，T_{amb}=25℃

参数名称	符号	测试条件	规范值			单位
			最小	典型	最大	
电源电压	V_{LN24}	正常工作	8	24	32	V
AVDD 端口输出电压	V_{AVDD}	正常工作		5.2		V
内部振荡器 1 时钟频率	F_{OSC8M}		7.92	8	8.08	MHz
内部振荡器 2 时钟频率	$F_{OSC400k}$		396	400	404	KHz
可编程光电放大器增益	F_{CONV}	通过寄存器 AMP_GAIN[3：0]设置	0.022		0.67	V/nA
ADC 基准	V_{REFADC}			2.4		V
ADC 转换时间	T_{CONV}	ADC 时钟周期为 T_{ADC}		32		T_{ADC}
可编程 IRED1 端口红外发射驱动电流	I_{IRED1}	V_{IRED1}≥1.5V，通过寄存器 IRED1_DAC 设置	16		500	mA
IRED1 端口红外发射驱动电流编程步进	I_{IRED1_STEP}			32		mA
可编程 IRED2 端口红外发射驱动电流	I_{IRED2}	V_{IRED2}≥1.5V，通过寄存器 IRED2_DAC 设置	0		200	mA
IRED2 端口红外发射驱动电流编程步进	I_{IRED2_STEP}			6.45		mA
可编程回码电流	I_{MOD}	V_{LN24}≥3V，通过寄存器 TXD_CUR 设置	0		115	mA
回码电流编程步进	I_{MOD_STEP}	V_{LN24}≥3V		3.7		mA

转下页

接上页

参 数 名 称	符 号	测 试 条 件	规 范 值			单位
			最小	典型	最大	
总线发码比较器 1 阈值	V_{TH1}	通过寄存器 PROT_THR1[5 ： 0] 设置	2		32	V
总线发码比较器 2 阈值	V_{TH2}	通过寄存器 PROT_THR2[5 ： 0] 设置	2		32	V
数据总线宽度	B_{BUS}	-		8		Bit
系统时钟频率	F_{SYS}	寄存器 Sys_clk_sw=0 OSC1SYS_DIV=0000 0000 时		4		MHz
RAM 容量	N_{RAM}			256		Byte
数据存储器容量	$N_{数据存储器}$			256		Byte
数据存储器保持时间	T_{ret}		10			年
数据存储器读写次数	N_{END}		100K			次

4、工作原理

4. 1、中央控制器

CPU 通过内部的系统总线连接到存储器和外围模块。存储器用来存放程序代码和数据。存储器和寄存器的地址访问都要基于 CPU 的指令集来实现。

所有的中断控制请求都会被收集起来发给 CPU，同时会记录下来所有发出中断控制信号，中断控制信号是专门有中断控制模块来产生并保持的。

4. 1. 1、CPU（中央处理器）

- 1 个 CPU 核心
- 兼容 8051 的指令集
- 2 个中断矢量信号
- 1 个复位矢量信号
- 系统时钟 8/4/2/1MHz、400KHz 可选

当电路复位的时候，内部所有的寄存器将会复位到初始状态，堆栈指针强制归 0，中断控制信号置为 1，屏蔽中断信号触发。

CPU 的指令集

算法操作：

Mnemonic	Description	Code	Bytes	Cycles
ADD A,Rn	Add register to accumulator	0x28-0x2F	1	1
ADD A,direct	Add direct byte to accumulator	0x25	2	2
ADD A,@Ri	Add indirect RAM to accumulator	0x26-0x27	1	2
ADD A,#data	Add immediate data to accumulator	0x24	2	2
ADDC A,Rn	Add register to accumulator with carry flag	0x38-0x3F	1	1
ADDC A,direct	Add direct byte to A with carry flag	0x35	2	2
ADDC A,@Ri	Add indirect RAM to A with carry flag	0x36-0x37	1	2
ADDC A,#data	Add immediate data to A with carry flag	0x34	2	2
SUBB A,Rn	Subtract register from A with borrow	0x98-0x9F	1	1
SUBB A,direct	Subtract direct byte from A with borrow	0x95	2	2
SUBB A,@Ri	Subtract indirect RAM from A with borrow	0x96-0x97	1	2
SUBB A,#data	Subtract immediate data from A with borrow	0x94	2	2
INC A	Increment accumulator	0x04	1	1
INC Rn	Increment register	0x08-0x0F	1	2
INC direct	Increment direct byte	0x05	2	3
INC @Ri	Increment indirect RAM	0x06-0x07	1	3
DEC A	Decrement accumulator	0x14	1	1
DEC Rn	Decrement register	0x18-0x1F	1	2
DEC direct	Decrement direct byte	0x15	1	3
DEC @Ri	Decrement indirect RAM	0x16-0x17	2	3
INC DPTR	Increment data pointer	0xA3	1	1
MUL A,B	Multiply A and B	0xA4	1	2
DIV A,B	Divide A by B	0x84	1	6
DA A	Decimal adjust accumulator	0xD4	1	3

逻辑操作：

Mnemonic	Description	Code	Bytes	Cycles
ANL A,Rn	AND register to accumulator	0x58-0x5F	1	1
ANL A,direct	AND direct byte to accumulator	0x55	2	2
ANL A,@Ri	AND indirect RAM to accumulator	0x56-0x57	1	2
ANL A,#data	AND immediate data to accumulator	0x54	2	2
ANL direct,A	AND accumulator to direct byte	0x52	2	3
ANL direct,#data	AND immediate data to direct byte	0x53	3	3
ORL A,Rn	OR register to accumulator	0x48-0x4F	1	1
ORL A,direct	OR direct byte to accumulator	0x45	2	2
ORL A,@Ri	OR indirect RAM to accumulator	0x46-0x47	1	2
ORL A,#data	OR immediate data to accumulator	0x44	2	2
ORL direct,A	OR accumulator to direct byte	0x42	2	3
ORL direct,#data	OR immediate data to direct byte	0x43	3	3
XRL A,Rn	Exclusive OR register to accumulator	0x68-0x6F	1	1
XRL A,direct	Exclusive OR direct byte to accumulator	0x65	2	2
XRL A,@Ri	Exclusive OR indirect RAM to accumulator	0x66-0x67	1	2
XRL A,#data	Exclusive OR immediate data to accumulator	0x64	2	2
XRL direct,A	Exclusive OR accumulator to direct byte	0x62	2	3
XRL direct,#data	Exclusive OR immediate data to direct byte	0x63	3	3
CLR A	Clear accumulator	0xE4	1	1
CPL A	Complement accumulator	0xF4	1	1
RL A	Rotate accumulator left	0x23	1	1
RLC A	Rotate accumulator left through carry	0x33	1	1
RR A	Rotate accumulator right	0x03	1	1
RRC A	Rotate accumulator right through carry	0x13	1	1
SWAP A	Swap nibbles within the accumulator	0xC4	1	1

布尔操作：

Mnemonic	Description	Code	Bytes	Cycles
CLR C	Clear carry flag	0xC3	1	1
CLR bit	Clear direct bit	0xC2	2	3
SETB C	Set carry flag	0xD3	1	1
SETB bit	Set direct bit	0xD2	2	3
CPL C	Complement carry flag	0xB3	1	1
CPL bit	Complement direct bit	0xB2	2	3
ANL C,bit	AND direct bit to carry flag	0x82	2	2
ANL C,/bit	AND complement of direct bit to carry	0xB0	2	2
ORL C,bit	OR direct bit to carry flag	0x72	2	2
ORL C,/bit	OR complement of direct bit to carry	0xA0	2	2
MOV C,bit	Move direct bit to carry flag	0xA2	2	2
MOV bit,C	Move carry flag to direct bit	0x92	2	3

数据传输：

Mnemonic	Description	Code	Bytes	Cycles
MOV A,Rn	Move register to accumulator	0xE8-0xEF	1	1
MOV A,direct	Move direct byte to accumulator	0xE5	2	2
MOV A,@Ri	Move indirect RAM to accumulator	0xE6-0xE7	1	2
MOV A,#data	Move immediate data to accumulator	0x74	2	2
MOV Rn,A	Move accumulator to register	0xF8-0xFF	1	1
MOV Rn,direct	Move direct byte to register	0xA8-0xAF	2	3
MOV Rn,#data	Move immediate data to register	0x78-0x7F	2	2
MOV direct,A	Move accumulator to direct byte	0xF5	2	2
MOV direct,Rn	Move register to direct byte	0x88-8F	2	2
MOV direct1,direct2	Move direct byte to direct byte	85	3	3
MOV direct,@Ri	Move indirect RAM to direct byte	86-87	2	3
MOV direct,#data	Move immediate data to direct byte	75	3	3
MOV @Ri,A	Move accumulator to indirect RAM	F6-F7	1	2
MOV @Ri,direct	Move direct byte to indirect RAM	A6-A7	2	3
MOV @Ri,#data	Move immediate data to indirect RAM	76-77	2	2

转下页

接上页

Mnemonic	Description		Code	Bytes	Cycles
MOV DPTR,#data 16	Load 16-bit constant into active DPH and DPL in LARGE mode		90	3	3
MOV DPTR,#data 24	Load 24-bit constant into active DPX,DPH and DPL in FLAT mode		90	4	4
MOVC A,@A+DPTR	Move code byte relative to DPTR to accumulator		93	1	5
MOVC A,@A+PC	Move code byte relative to PC to accumulator		83	1	4
MOVX A,@Ri	Move external RAM (8-bit address) to A		E2-E3	1	3*
MOVX A,@DPTR	Move external RAM (16-bit address) to A		E0	1	2*
MOVX @Ri,A	Move A to external RAM(8-bit address)	CODE inside ROM/RAM destination XRAM data	F2-F3	1	4*
		all other cases			5*
MOVX @DPTR,A	Move A to external RAM(16-bit address)	CODE inside ROM/RAM destination XRAM data	F0	1	3*
		all other cases			4*
PUSH direct	Push direct byte onto stack	LARGE	C0	2	3
		FLAT			3
POP direct	Pop direct byte from stack	LARGE	D0	2	2
		FLAT			2
XCH A,Rn	Exchange register with accumulator		C8-CF	1	2
XCH A,direct	Exchange direct byte with accumulator		C5	2	3
XCH A,@Ri	Exchange indirect RAM with accumulator		C6-C7	1	3
XCHD A,@Ri	Exchange low-order nibble indirect RAM with A		D6-D7	1	3

程序分支：

Mnemonic	Description		Code	Bytes	Cycles
ACALL addr11	Absolute subroutine call	LARGE	0x11-0xF1	2	4
ACALL addr19		FLAT		3	5
LCALL addr16	Long subroutine call	LARGE	03	3	4
LCALL addr24		FLAT		4	6
RET	Return from subroutine	LARGE	22	1	4
		FLAT			5
RETI	Return from interrupt	LARGE	32	1	4
		FLAT			5
AJMP addr11	Absolute jump	LARGE	01-E1	2	3
AJMP addr19		FLAT		3	4
LJMP addr16	Long jump	LARGE	02	3	4
LJMP addr24		FLAT		4	5
SJMP rel	Short jump (relative address)		80	2	3
JMP @A+DPTR	Jump indrect relative to the DPTR		73	1	5
JZ rel	Jump if accumulator is zero		60	2	4
JNZ rel	Jump if accumulator is not zero		70	2	4
JC rel	Jump if carry flag is set		40	2	3
JNC rel	Jump if carry flag is not set		50	2	3
JB bit,rel	Jump if direct bit is set		20	3	5
JNB bit,rel	Jump if direct bit is not set		30	3	5
JBC bit,direct rel	Jump if direct bit is set and clear bit		10	3	5
CJNE A,direct rel	Compare direct byte to A and jump if not equal		B5	3	5
CJNE A,#data rel	Compare immediate to A and jump if not equal		B4	3	4
CJNE Rn,#data rel	Compare immediate to reg. and jump if not equal		B8-BF	3	4
CJNE @Ri,#data rel	Compare immediate to ind. and jump if not equal		B6-B7	3	5
DJNZ Rn,rel	Decrement register and jump if not zero		D8-DF	2	4
DJNZ direct,rel	Decrement direct byte and jump if not zero		D5	3	5
NOP	NO operation		00	1	1

指令集简介 1：

Opcode	Mnemonic	Opcode	Mnemonic
00 H	NOP	30 H	JNB bit.rel
01 H	AJMP addr11/addr19	31 H	ACALL addr11/addr19
02 H	LJMP addr16/addr24	32 H	RETI
03 H	RR A	33 H	RLC A
04 H	INC A	34 H	ADDC A,#data
05 H	INC direct	35 H	ADDC A,direct
06 H	INC @R0	36 H	ADDC A,@R0
07 H	INC @R1	37 H	ADDC A,@R1
08 H	INC R0	38 H	ADDC A,R0
09 H	INC R1	39 H	ADDC A,R1
0A H	INC R2	3A H	ADDC A,R2
0B H	INC R3	3B H	ADDC A,R3
0C H	INC R4	3C H	ADDC A,R4
0D H	INC R5	3D H	ADDC A,R5
0E H	INC R6	3E H	ADDC A,R6
0F H	INC R7	3F H	ADDC A,R7
10 H	JBC bit.rel	40 H	JC rel
11 H	ACALL addr11/addr19	41 H	AJMP addr11/addr19
12 H	LCALL addr16/addr24	42 H	ORL direct,A
13 H	RRC A	43 H	ORL direct,#data
14 H	DEC A	44 H	ORL A,#data
15 H	DEC direct	45 H	ORL A,direct
16 H	DEC @R0	46 H	ORL A,@R0
17 H	DEC @R1	47 H	ORL A,@R1
18 H	DEC R0	48 H	ORL A,R0
19 H	DEC R1	49 H	ORL A,R1
1A H	DEC R2	4A H	ORL A,R2
1B H	DEC R3	4B H	ORL A,R3
1C H	DEC R4	4C H	ORL A,R4
1D H	DEC R5	4D H	ORL A,R5
1E H	DEC R6	4E H	ORL A,R6
1F H	DEC R7	4F H	ORL A,R7
20 H	JB bit.rel	50 H	JNC rel
21 H	AJMP addr11/addr19	51 H	ACALL addr11/addr19
22 H	RET	52 H	ANL direct,A
23 H	RLA	53 H	ANL direct,#data
24 H	ADD A,#data	54 H	ANL A,#data
25 H	ADD A,direct	55 H	ANL A,direct

转下页

接上页

Opcode	Mnemonic	Opcode	Mnemonic
26 H	ADD A,@R0	56 H	ANL A,@R0
27 H	ADD A,@R1	57 H	ANL A,@R1
28 H	ADD A,R0	58 H	ANL A,R0
29 H	ADD A,R1	59 H	ANL A,R1
2A H	ADD A,R2	5A H	ANL A,R2
2B H	ADD A,R3	5B H	ANL A,R3
2C H	ADD A,R4	5C H	ANL A,R4
2D H	ADD A,R5	5D H	ANL A,R5
2E H	ADD A,R6	5E H	ANL A,R6
2F H	ADD A,R7	5F H	ANL A,R7

指令集简介 2：

Opcode	Mnemonic	Opcode	Mnemonic
60 H	JZ rel	90 H	MOV DPTR,#data16 MOV DPTR,#data24
61 H	AJMP addr11	91 H	ACALL aaddr11
62 H	XRL direct,A	92 H	MOV bit,C
63 H	XRL direct,#data	93 H	MOVC A,@A+DPTR
64 H	XRL A,#data	94 H	SUBB A,#data
65 H	XRL A,direct	95 H	SUBB A,direct
66 H	XRL A,@R0	96 H	SUBB A,@R0
67 H	XRL A,@R1	97 H	SUBB A,@R1
68 H	XRL A,R0	98 H	SUBB A,R0
69 H	XRL A,R1	99 H	SUBB A,R1
6A H	XRL A,R2	9A H	SUBB A,R2
6B H	XRL A,R3	9B H	SUBB A,R3
6C H	XRL A,R4	9C H	SUBB A,R4
6D H	XRL A,R5	9D H	SUBB A,R5
6E H	XRL A,R6	9E H	SUBB A,R6
6F H	XRL A,R7	9F H	SUBB A,R7
70 H	JNZ,rel	A0 H	ORL C,bit
71 H	ACALL addr11/addr19	A1 H	AJMP addr11/addr19
72 H	ORL C,direct	A2 H	MOV C,bit
73 H	JMP @A+DPTR	A3 H	INC DPTR
74 H	MOV A,#data	A4 H	MUL AB
75 H	MOV direct,#data	A5 H	
76 H	MOV @R0,#data	A6 H	MOV @R0,direct
77 H	MOV @R1,#data	A7 H	MOV @R1,direct
78 H	MOV R0.#data	A8 H	MOV R0,direct

转下页

接上页

Opcode	Mnemonic	Opcode	Mnemonic
79 H	MOV R1.#data	A9 H	MOV R1,direct
7A H	MOV R2.#data	AA H	MOV R2,direct
7B H	MOV R3.#data	AB H	MOV R3,direct
7C H	MOV R4.#data	AC H	MOV R4,direct
7D H	MOV R5.#data	AD H	MOV R5,direct
7E H	MOV R6.#data	AE H	MOV R6,direct
7F H	MOV R7.#data	AF H	MOV R7,direct
80 H	SJMP rel	B0 H	ANL C,bit
81 H	AJMP addr11/addr19	B1 H	ACALL addr11/addr19
82 H	ANL C,bit	B2 H	CPL bit
83 H	MOVC A,@A+PC	B3 H	CPL C
84 H	DIV AB	B4 H	CJNE A,#data,rel
85 H	MOV direct,direct	B5 H	CJNE A,direct,rel
86 H	MOV direct,@R0	B6 H	CJNE @R0,#data,rel
87 H	MOV direct,@R1	B7 H	CJNE @R1,#data,rel
88 H	MOV direct,R0	B8 H	CJNE R0,#data,rel
89 H	MOV direct,R1	B9 H	CJNE R1,#data,rel
8A H	MOV direct,R2	BA H	CJNE R2,#data,rel
8B H	MOV direct,R3	BB H	CJNE R3,#data,rel
8C H	MOV direct,R4	BC H	CJNE R4,#data,rel
8D H	MOV direct,R5	BD H	CJNE R5,#data,rel
8E H	MOV direct,R6	BE H	CJNE R6,#data,rel
8F H	MOV direct,R7	BF H	CJNE R7,#data,rel

指令集简介 3：

Opcode	Mnemonic	Opcode	Mnemonic
C0 H	PUSH direct	E0 H	MOVX A,@DPTR
C1 H	AJMP addr11/addr19	E1 H	AJMP addr11/addr19
C2 H	CLR bit	E2 H	MOVX A,@R0
C3 H	CLR C	E3 H	MOVX A,@R1
C4 H	SWAP A	E4 H	CLR A
C5 H	XCH A,direct	E5 H	MOV A,direct
C6 H	XCH A,@R0	E6 H	MOV A,@R0
C7 H	XCH A,@R1	E7 H	MOV A,@R1
C8 H	XCH A,R0	E8 H	MOV A,R0
C9 H	XCH A,R1	E9 H	MOV A,R1
CA H	XCH A,R2	EA H	MOV A,R2

转下页

接上页

Opcode	Mnemonic	Opcode	Mnemonic
CB H	XCH A,R3	EB H	MOV A,R3
CC H	XCH A,R4	EC H	MOV A,R4
CD H	XCH A,R5	ED H	MOV A,R5
CE H	XCH A,R6	EE H	MOV A,R6
CF H	XCH A,R7	EF H	MOV A,R7
D0 H	POP direct	F0 H	MOVX @DPTR,A
D1 H	ACALL addr11/addr19	F1 H	ACALL addr11/addr19
D2 H	SETB bit	F2 H	MOVX @R0,A
D3 H	SETB C	F3 H	MOVX @R1,A
D4 H	DA A	F4 H	CPL A
D5 H	DJNZ direct,rel	F5 H	MOV direct,A
D6 H	XCHD A,@R0	F6 H	MOV @R0,A
D7 H	XCHD A,@R1	F7 H	MOV @R1,A
D8 H	DJNZ R0,rel	F8 H	MOV R0,A
D9 H	DJNZ R1,rel	F9 H	MOV R1,A
DA H	DJNZ R2,rel	FA H	MOV R2,A
DB H	DJNZ R3,rel	FB H	MOV R3,A
DC H	DJNZ R4,rel	FC H	MOV R4,A
DD H	DJNZ R5,rel	FD H	MOV R5,A
DE H	DJNZ R6,rel	FE H	MOV R6,A
DF H	DJNZ R7,rel	FF H	MOV R7,A

4. 2、ADDRESS MAP 控制模块

ADDRESS MAP 模块可以产生除 MCU 外的各模块片选信号，管理除特殊功能寄存器（SFR）外的所有寄存器和存储器地址，且 MCU 可以通过该模块的控制对这些寄存器和存储器进行读写。

注：无效地址访问会导致系统复位。

寄存器描述：

无效访问寄存器（ERREN）

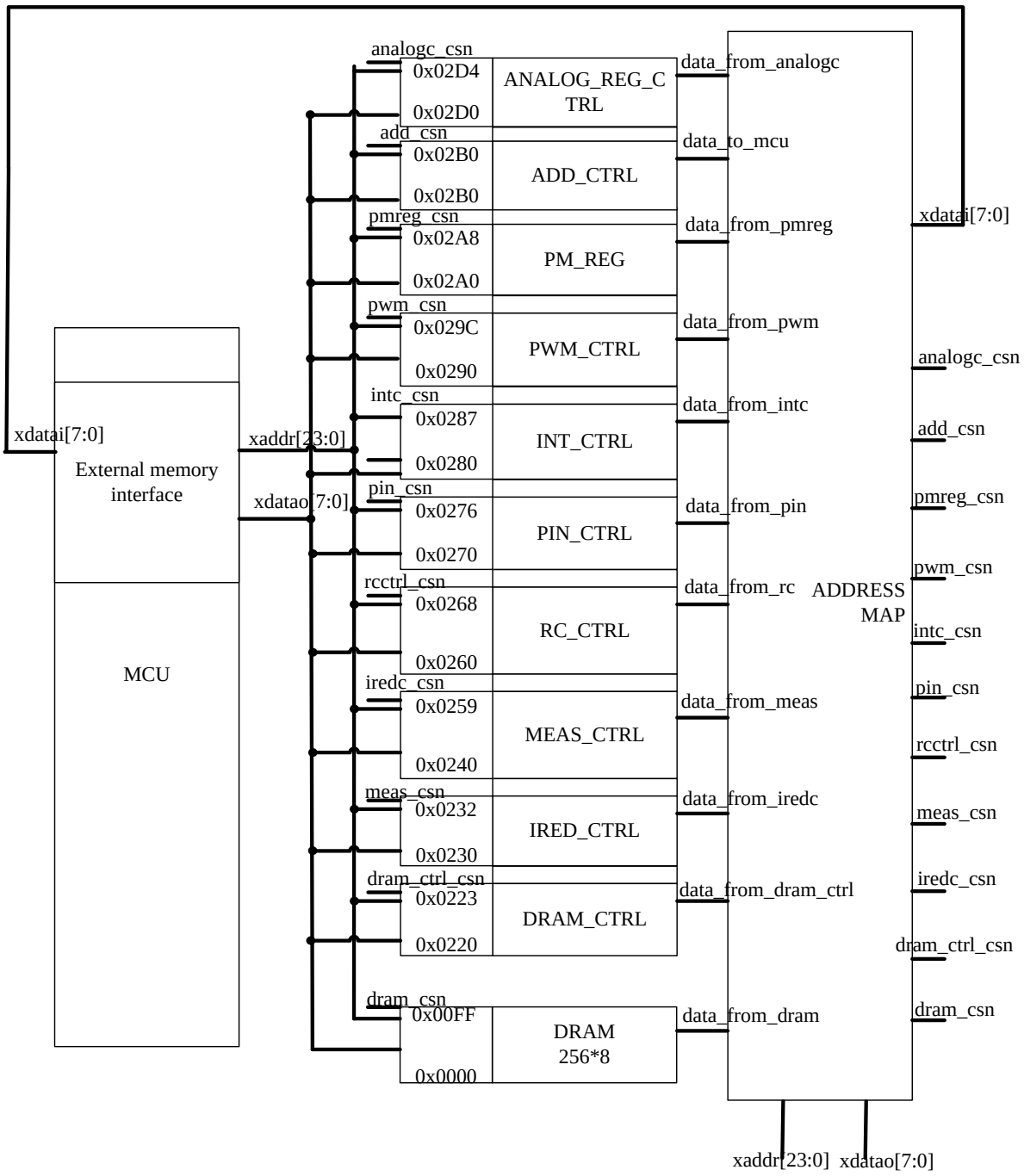
地址 ：0x02B0

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：4]			保留
[3]	Fint_en	R/W	Flash 地址错误中断信号输出使能： 1：禁止 Flash_int 从 address_int 端口输出； 0：使能 Flash_int 从 address_int 端口输出
[2]	Mint_en	R/W	无效地址访问中断信号输出使能： 1：禁止 map_int 从 address_int 口输出； 0：使能 map_int 从 address_int 口输出
[1]	Flash_int	R/W	Flash 编程地址错误中断信号： 无效地址访问时该信号置“1”； 由 MCU 写“0”清除该中断信号
[0]	Map_int	R/W	无效地址访问（R/W）中断信号： 无效地址访问时该信号置“1”； 由 MCU 写“0”清除该中断信号

ADDRESS MAP 地址分配图：



4. 3、时钟与复位控制单元

芯片内部有两个时钟源：

（1）一个振荡频率为 400kHz 的低速振荡器 OSC400K，该振荡器可以连续不断的工作。所有需要连续执行的任务和时序都被控制在低速时钟域内，目的是保持所有的电流消耗最低。

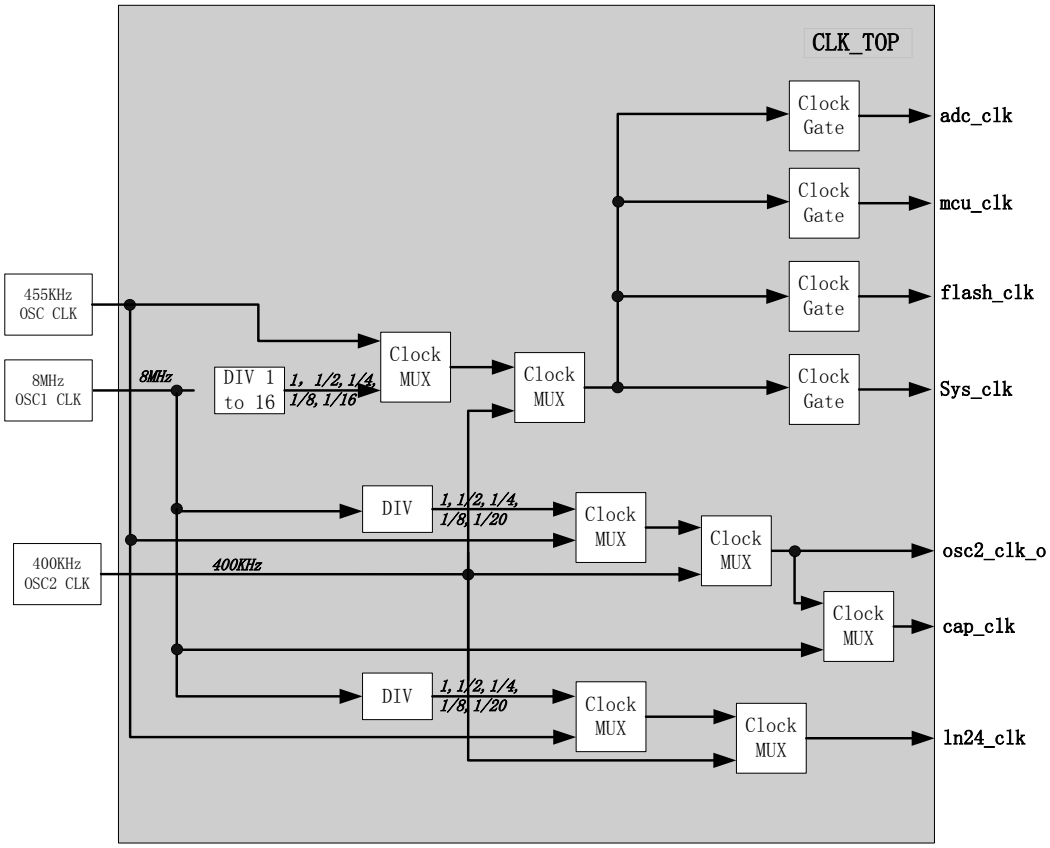
（2）一个振荡频率为 8MHz 的高速振荡器 OSC8M。振荡器可以由用户分频输出作为系统时钟。为了省电，用户可以在 CPU 空闲时进入休眠状态，通过向 PCON 寄存器 stop 位写 1 就可使进入休眠并自动将高速振荡器停止工作。CPU 保持休眠状态，同时 FLASH 变为睡眠模式。当有中断信号时，OSC8M 首先启动，然后是 FLASH 启动，最后是 CPU 启动。

芯片只可以通过配置相应寄存器，使能外置晶振作为系统时钟。

在以下状况可以使系统复位：

- 当系统上电时复位（POR）；
- 无效的地址读写；
- 看门狗溢出复位；
- 错误的数据存储器锁定状态下擦除/编写程序导致复位；
- 调试模式下强制进行调试复位。

具体时钟树如下图所示：



时钟控制寄存器：

地址	寄存器名称	有效位	访问	复位值
0x02A0	SYS_REG1	8	R/W	0x1B
0x02A1	RST_FLAG	2	R/W	0x00
0x02A2	SOFTRST_REG	2	R/W	0x00
0x02A3	CLK8M_DIV	3	R/W	0x00
0x02A4	LN24_DIV	3	R/W	0x00
0x02A5	SYS_REG_MASK	8	R/W	0x00
0x02A6	OSC1SYS_DIV	3	R/W	0x00
0x02A7	SYS_REG2	8	R/W	0x99

寄存器描述

1）系统寄存器 1（SYS_REG1）

地址 ：0x02A0

复位值 ：0x1B

访问 ：R/W

Bit	Name	R/W	Description
[7]	Test_ mode	R	测试模式选择： 0：正常工作模式； 1：测试模式
[6]	Wdt_ rst_ en	R/W	看门狗复位使能： 0：关闭； 1：打开
[5]	Prg_ err_ en	R/W	程序出错复位使能： 0：关闭； 1：打开
[4]	Hard_ off_ en	R/W	休眠模式下关闭 OSC8M： 0：休眠模式关闭 8MHz 振荡器时钟； 1：休眠模式不关闭 8MHz 振荡器时钟
[3]	Osc1en_reg	R/W	OSC8M 开关寄存器： 0：关闭 8MHz 振荡器时钟； 1：打开 8MHz 振荡器时钟
[2]	Sys_ clk_ sw	R/W	系统时钟源选择： 0：OSC8M 分频（设置分频器 OSC1SYS_DIV）； 1：OSC400K
[1]	Ln24_sel	R/W	LN24 时钟源选择： 0：OSC8M 分频（设置分频器 LN24_DIV）； 1：OSC400K
[0]	Clk400_sel	R/W	400K 时钟源选择： 0：OSC8M 分频（设置分频器 CLK8M_DIV）； 1：OSC400K

2）复位标志寄存器（RST_FLAG）

地址 ：0x02A1

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：3]			保留
[2]	Prg_ err_ flag	R/W	程序出错复位标志，高有效 再次写“1”可清除该标志
[1]	Wdt_ rst_ flag	R/W	看门狗复位标志，高有效 再次写“1”可清除该标志
[0]			保留

3）软复位寄存器（SOFTRST_REG）

地址 ：0x02A2

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：2]			保留
[1]	ln24_soft_rst	R/W	LN24 软复位信号，高有效
[0]	osc2_soft_rst	R/W	400K 的软复位信号，高有效

4）8MHz 时钟分频寄存器（CLK8M_DIV）

地址 ：0x02A3

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：3]			保留
[2]	Clk_ sel	R/W	时钟输出选择： 1：选择 8MHz 时钟； 0：选择 8MHz 分频时钟
[1：0]	Div_osc8m [1：0]	R/W	8MHz 时钟分频选择： 00：1/2； 01：1/4； 10：1/8； 11：1/20； 占空比均为 50：50

5）LN24 时钟分频寄存器（LN24_DIV）

地址 ：0x02A4

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：3]			保留
[2]	Clk_ sel	R/W	时钟输出选择： 1：选择 8MHz 时钟； 0：选择分频时钟
[1：0]	Div_ln24 [1：0]	R/W	LN24 时钟分频选择： 00：1/2； 01：1/4； 10：1/8； 11：1/20； 占空比均为 50：50

6）系统寄存器屏蔽寄存器（SYS_REG_MASK）

地址 ：0x02A5

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	Sys_ reg_ mask	R/W	系统寄存器屏蔽寄存器 只有向此寄存器写入 0xA5 方可操作寄存器 SYS_REG1、SYS_REG 2

7）OSC1SYS 时钟（源自 OSC8M）分频寄存器（OSC1SYS_DIV）

地址 ：0x02A6

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：3]			保留
[2]	Clk_ sel	R/W	时钟输出选择： 1：选择 8MHz 时钟； 0：选择 8MHz 分频时钟
[1：0]	Div_osc1sys[1：0]	R/W	OSC8M 时钟分频选择（输出时钟提供给系统时钟）： 00：1/2； 01：1/4； 10：1/8； 11：1/16； 占空比均为 50：50

8）系统寄存器 2（SYS_REG2）

地址 ：0x02A7

复位值 ：0x99

访问 ：R/W

Bit	Name	R/W	Description
[7]	Ldo_ sel	R/W	休眠模式下 LDO 模式选择： 0：休眠模式进入弱模式； 1：休眠模式保持强模式
[6]	Tidle_ sel	R/W	Flash Sleep 到 Idle 时间选择（如系统时间切换成 400K，该时间固定为 10μs）： 0：16μs（考虑到上电后 OSC8M 时钟不稳，时间设的比较长）； 1：8μs
[5]	Osc455_en	R/W	外部晶振 OSC455K 使能信号： 0：关闭； 1：打开
[4]	Osc455_mode	R/W	外部晶振 OSC455K 模式选择： 0：低功耗； 1：高功耗
[3]	Osc8_455_sw	R/W	振荡器 8M 和外部晶振 455K 切换选择： 0：选择外部晶振 455K； 1：选择振荡器 8M
[2：0]	Osc1_lock_sw	R/W	振荡器 OSC8M 稳定计数时间选择： 000：20μs； 001：17.5μs（默认）； 010：15μs； 011：12.5μs； 100：10μs； 101：7.5μs； 110：5μs； 111：2.5μs

4. 4、特殊功能寄存器

SFR 寄存器地址映射

特殊功能寄存器为 128Bytes，CPU 本身需要占用一部分，其余部分为外设 SFR（Timer0/1、Watchdog、PAPORT、PBPORT、UART、LN24_CTRL、CAP_CTRL）。

特殊功能寄存器分配表

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
0xF8	*LN24MIRQ*								0xF8
0xF0	**B**	*CAPCON*	*CAPCON1*	*CAPCNTL*	*CAPCNTH*				0xF0
0xE8	*LN24STAT1*	**STATUS**	**MXAX**	**TA**					0xE8
0xE0	**ACC**	*LN24CTRL*	*LN24STAT2*	*RXFILTER*	*DISCHAR_CTRL*	*DISCHAR_THR*	*RXBLK2_L*	*RXBLK2_H*	0xE0
0xD8	*WDCON*	*WDCNTM*	*WDCNTH*	*XDATA_CTRL*					0xD8
0xD0	**PSW**	*RXREAD_L*	*RXREAD_H*	*RXCAP1_L*	*RXCAP1_H*	*RXCAP2_L*	*RXCAP2_H*		0xD0
0xC8	*TTMOD2*	*RXCOMP1_L*	*RXCOMP1_H*	*RXCOMP2_L*	*RXCOMP2_H*	*RXBLK1_L*	*RXBLK1_H*		0xC8
0xC0	*POWCONF*	*LN24CONF1*	*LN24CONF2*	*LN24CONF3*	*TXDEL_L*	*TXDEL_H*	*TXDUR_L*	*TXDUR_H*	0xC0
0xB8	**IP**								0xB8
0xB0									0xB0
0xA8	**IE**	*PBREG*	*PBDIR*	*PBIE*	*PBIES*	*PBIFG*			0xA8
0xA0	**P2**	*PAREG*	*PADIR*	*PAIE*	*PAIES*	*PAIFG*			0xA0
0x98	*SCON*	*SBUF*	*SMOD*		**AP**	**ACON**			0x98
0x90			**WTST**	**DPX0**					0x90
0x88	**TCON**	*TMOD*	*TL0*	*TL1*	*TH0*	*TH1*	**CKCON**		0x98
0x80		**SP**	**DPL0**	**DPH0**			*TTCON*	**PCON**	0x80
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

另外，PCON 的 bit1 是 STOP 位，CPU 写 STOP 位为 1 将进入休眠模式，在进入休眠模式之前需将相关中断控制器及接口模块使能，以保证 CPU 能被唤醒，在 CPU 唤醒之后，STOP 自动清 0。

注意：在 STOP 指令之后需要至少 4 条 NOP 指令，以保证程序唤醒后正常运行。

在下述使用 SFR 模块中提到的 sys_ clk 均为系统时钟。

CPU 所占用的特殊功能寄存器描述

1）累加器（ACC）

SFR 地址 ：0xE0

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	ACC.7~ACC.0	R/W	

2）累加器（B）

SFR 地址 ：0xF0

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	B.7~B.0	R/W	

3）寄存器（PSW）

SFR 地址 ：0xD0

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	CY	R/W	进位标志
[6]	AC	R/W	辅助进位
[5]	F0	R/W	通用标志 0
[4：3]	RS[1：0]	R/W	寄存器 R0~R7 bank 选择位： 00：Bank0，地址：0x00~0x07； 01：Bank1，地址：0x08~0x0F； 10：Bank2，地址：0x10~0x17； 11：Bank3，地址：0x18~0x1F
[2]	OV	R/W	溢出标志
[1]	F1	R/W	通用标志 1
[0]	P	R/W	奇偶标志

4）FLAT/LARGE 模式切换寄存器（ACON）

SFR 地址 ：0x9D

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：2]			保留
[1]	AM	R/W	地址模式选择： 0：16 位程序地址模式– LARGE； 1：23 位扩展程序地址模式– FLAT 本项目存储空间仅适用于 LARGE 模式
[0]			保留

5）程序写使能寄存器（PCON）

SFR 地址 ： 0x87

复位值 ： 0x00

访问 ： R/W

Bit	Name	R/W	Description
[7：5]			保留
[4]	PWE	R/W	xdata 写操作控制： 0：允许 xdata 写操作； 1：禁止 xdata 写操作
[3]			保留
[2]	SWB		保留，不能使用
[1]	STOP	R/W	暂停 CPU 程序执行： MCU 外部电路收到外围设备的指令或内部低频模块的处理完成中断后，将会唤醒 MCU
[0]	PMM		保留，不能使用

6）程序等待周期控制寄存器（WTST）

SFR 地址 ： 0x92

复位值 ： 0x00

访问 ： R/W

Bit	Name	R/W	Description
[7：3]			保留
[2：0]	WTST	R/W	在本电路中，WTST 保留，不需要使用

7）寄存器（DPX0）

SFR 地址 ： 0x93

复位值 ： 0x00

访问 ： R/W

Bit	Name	R/W	Description
[7：0]	DPX.7~ DPX.0	R/W	

8）寄存器（MXAX）

SFR 地址 ： 0xEA

复位值 ： 0x00

访问 ： R/W

Bit	Name	R/W	Description
[7：0]	MXAX.7~ MXAX.0	R/W	

9）寄存器（P2）

SFR 地址 ：0xA0

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	P2.7~P2.0	R/W	

10）数据指针低八位寄存器（DPL0）

SFR 地址 ：0x82

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	DPL0	R/W	

11）数据指针高八位寄存器（DPH0）

SFR 地址 ：0x83

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	DPH0	R/W	

12）时钟控制寄存器（CKCON）

SFR 地址 ：0x8E

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：3]			保留
[2：0]	MD[2：0]	R/W	在本项目中，CKCON 保留，不需要使用

13）堆栈指针寄存器（SP）

SFR 地址 ：0x81

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	SP.7~SP.0	R/W	

14）中断使能寄存器（IE）

SFR 地址 ：0xA8

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	EA	R/W	全局中断使能位
[6：3]			保留
[2]	EX1	R/W	Int1 中断使能位
[1]			保留
[0]	EX0	R/W	Int0 中断使能位

本电路中，CPU共两个中断向量，如下表所示：

Interrupt Flag	Function	Active Level Edge	Flag Resets	Vector	Natural Priority
IE0	Device pin INT0	Low/Falling	Hardware	0x03	1
IE1	Device pin INT1	Low/Falling	Hardware	0x13	3

15）中断优先级控制寄存器（IP）

SFR 地址 ：0xB8

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：3]			保留
[2]	PX1	R/W	Int1 优先级控制位
[1]			保留
[0]	PX0	R/W	Int0 优先级控制位

16）中断配置寄存器（TCON）

SFR 地址 ：0x88

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：4]			保留
[3]	IE1	R/W	Int1 中断标志位： 当 MCU 进入中断处理程序后，硬件自动清 0
[2]	IT1	R/W	Int1 电平和沿触发选择位： 0：电平触发； 1：沿触发
[1]	IE0	R/W	Int0 中断标志位： 当 MCU 进入中断处理程序后，硬件自动清 0

转下页

接上页

Bit	Name	R/W	Description
[0]	IT0	R/W	Int0 电平和沿触发选择位： 0：电平触发； 1：沿触发

17）中断状态寄存器（STATUS）

SFR 地址 ：0xE9

复位值 ：0x00

访问 ：R

Bit	Name	R/W	Description
[7]			保留
[6]	HIP	R	高优先级中断状态
[5]	LIP	R	低优先级中断状态
[4：0]			保留

4. 5、定时器

在本电路集成两个 16bits 的定时器，Timer0 和 Timer1。这两个定时器的控制方式完全相同，可作为普通定时器使用，也可以用作产生 UART 的波特率。

作为普通定时器时，16bits 的定时器模块根据软件配置的定时周期，和选择的 sys_clk 频率的分频系数（1，1/2，1/4 或 1/8）进行计数，输出定时中断信号。作为波特率产生时，可以选择采用两个 8bits 的定时器分别用两个不同的计数时钟进行计数，共同产生一个计时溢出周期信号。其中低 8 位采用普通定时器的计数 clk（sys_ clk 频率的 1 分频，2 分频 4 分频，8 分频）进行计数，高 8 位可选择采用 sys_ clk 频率的 12 分频或 16 分频的计数 clk 进行计数，波特率周期为高低 8 位的计数时间之和。

寄存器描述：

1）Timer 控制寄存器（TTCON）

SFR 地址 ：0x86

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	TF1	R/W	Timer1 中断标志位： 高有效
[6]	TF0	R/W	Timer0 中断标志位： 高有效
[5]	TR1	R/W	Timer1 计数使能位： 0：除能； 1：使能

转下页

接上页

Bit	Name	R/W	Description
[4]	TR0	R/W	Timer0 计数使能位： 0：除能； 1：使能
[3]	TE1	R/W	Timer1 使能位： 0：除能； 1：使能
[2]	TE0	R/W	Timer0 使能位： 0：除能； 1：使能
[1]	LOAD1	R/W	Timer1 定时周期配置有效位： 软件写 1 有效，硬件自动清零
[0]	LOAD0	R/W	Timer0 定时周期配置有效位： 软件写 1 有效，硬件自动清零

软件在更新定时周期寄存器的值后，需要配置 TTCON 的 LOAD0 或 LOAD1 为 1 后，配置的定时周期才有效。

TE0 和 TE1 为 Timer0 和 Timer1 的使能位，为低时 Timer0 和 Timer1 处于复位状态。

TR0 和 TR1 为 Timer0 和 Timer1 的计数使能位，为低时 Timer0 和 Timer1 处于计数暂停状态（保持当前计数值），定时器作波特率产生时，不支持暂停控制。

TF0 和 TF1 为 Timer0 和 Timer1 的中断标志位，软件检测到 Timer 中断后应立即清除中断标志，即往 TF0 或 TF1 写 0。即使 Timer0 和 Timer1 的中断使能信号为低，当 Timer0 和 Timer1 定时溢出时 TF0 和 TF1 位也会变成 1。

2）Timer 模式寄存器（TMOD）

SFR 地址 ：0x89

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	GATE1_EN	R/W	Timer1 门控使能位： 0：除能； 1：使能
[6]	T1_IE	R/W	Timer1 中断使能位： 0：除能； 1：使能
[5：4]	DIV1	R/W	Timer1 预分频系数： 00：1 分频； 01：2 分频； 10：4 分频； 11：8 分频

转下页

接上页

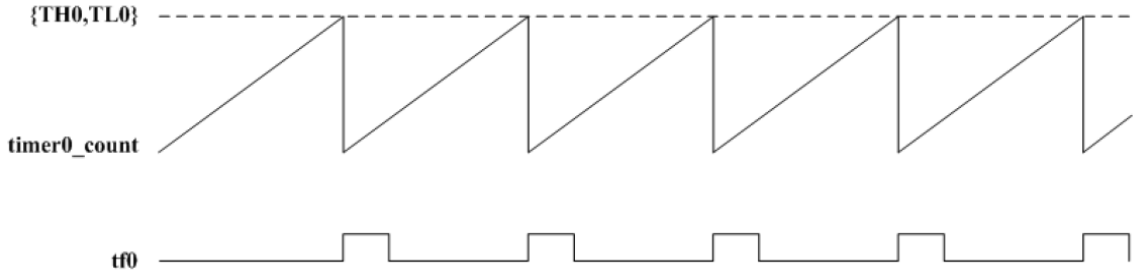
Bit	Name	R/W	Description
[3]	GATE0_EN	R/W	Timer0 门控使能位： 0：除能； 1：使能
[2]	T0_IE	R/W	Timer0 中断使能位： 0：除能； 1：使能
[1：0]	DIV0	R/W	Timer0 预分频系数： 00：1 分频； 01：2 分频； 10：4 分频； 11：8 分频

T0_IE/T1_IE 为 Timer0/Timer1 的中断使能信号。

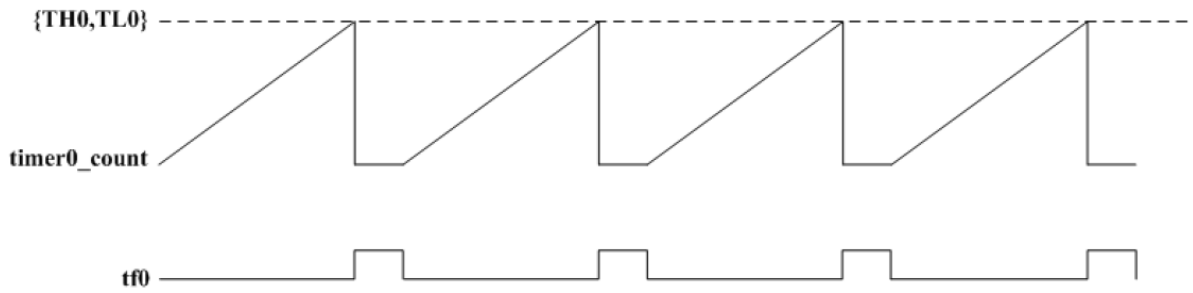
GATE0_EN/GATE1_EN 为 Timer0/Timer1 的门控使能信号。当门控使能为高时，如果 Timer0/Timer1 计数溢出，Timer0/ Timer1 将处于复位状态（计数值保持为零并停止计数）直到标志位 TF0/TF1 被清零后继续开始计数（如果 TE0/TE1 和 TR0/TR1 保持有效的话）；当门控使能为低时，如果 Timer0/Timer1 计数溢出，Timer0/Timer1 将继续进行下一轮计数（如果 TE0/TE1 和 TR0/TR1 保持有效的话），不管标志位 TF0/TF1 是否被清零。若用作产生串口的波特率时，门控使能信号应配置成 0。

DIV0/DIV1 为 Timer0/Timer1 的预分频系数，使得 Timer0/Timer1 可以在比系统时钟频率更低的时钟频率下进行计数。

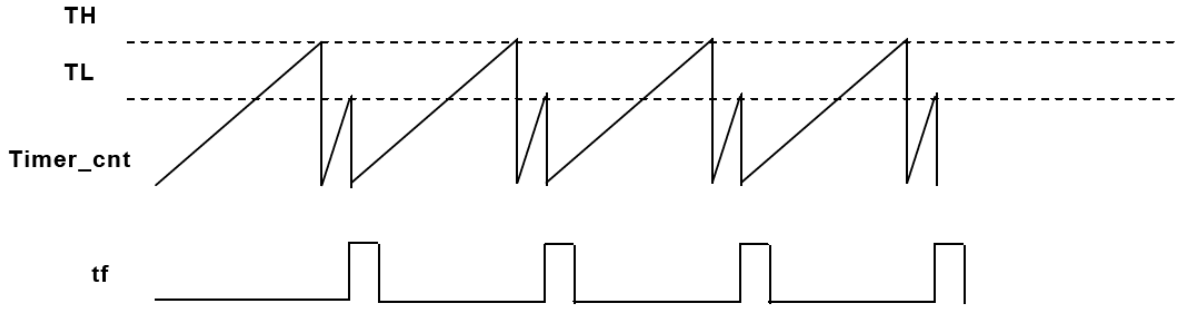
门控使能无效时 Timer0/ Timer1 定时器计数示意图：



门控使能有效时 Timer0/ Timer1 定时器计数示意图：



BAUD_SEL0 为 1 时 Timer0/ Timer1 定时器计数示意图：



3）Timer0 计数周期高八位寄存器（TH0）

SFR 地址 ： 0x8C

复位值 ： 0xff

访问 ： R/W

Bit	Name	R/W	Description
[7：0]	TIMER0_TH[15：8]	R/W	Timer0 计数周期高 8 位

4）Timer0 计数周期低八位寄存器（TL0）

SFR 地址 ： 0x8A

复位值 ： 0xff

访问 ： R/W

Bit	Name	R/W	Description
[7：0]	TIMER0_TH[7：0]	R/W	Timer0 计数周期低 8 位

5）Timer1 计数周期高八位寄存器（TH1）

SFR 地址 ： 0x8D

复位值 ： 0xff

访问 ： R/W

Bit	Name	R/W	Description
[7：0]	TIMER1_TH[15：8]	R/W	Timer1 计数周期高 8 位

6）Timer1 计数周期低八位寄存器（TH1）

SFR 地址 ： 0x8B

复位值 ： 0xff

访问 ： R/W

Bit	Name	R/W	Description
[7：0]	TIMER1_TH[7：0]	R/W	Timer1 计数周期低 8 位

7）Timer 模式寄存器 2（TTMOD2）

SFR 地址 ：0xC8

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：6]			保留
[5]	RCLK	R/W	串口接收部分选择 Time0 或 Time1 波特率的选择位： 0：选择 Time1 产生波特率； 1：选择 Time0 产生波特率
[4]	TCLK	R/W	串口发送部分选择 Time0 或 Time1 波特率的选择位 0：选择 Time1 产生波特率； 1：选择 Time0 产生波特率
[3]	H_CLK_SE1	R/W	BAUD_SEL1 为 1 时，Time1 高八位计数时钟选择位： 0：1/12 sys_ clk； 1：1/16 sys_ clk
[2]	BAUD_SEL1	R/W	Time1 波特率产生方式，高低 8bits 分别计数标志位： 当该位为 0 时，也可用作串口波特率产生，此时为 16bit 的计数模式； 当 RCLK 或 TCLK 为 0 时，串口选择 Time1 产生的波特率
[1]	H_CLK_SEL0	R/W	BAUD_SEL0 为 1 时，Time0 高八位计数时钟选择位： 0: 1/12 sys_ clk； 1: 1/16 sys_ clk
[0]	BAUD_SEL0	R/W	Time0 波特率产生方式，高低 8bits 分别计数标志位： 当该位为 0 时，也可用作串口波特率产生，此时为 16bit 的计数模式； 当 RCLK 或 TCLK 为 1 时，串口接收或发送选择 time0 产生的模特率

如果选择 16 为计数器产生串口的溢出时钟，即 BAND_SEL 为 0，那么计数器 16 位计数器以系统时钟 div 分频产生溢出时钟，如门控使能无效时 Timer0/ Timer1 定时器计数示意图所示，此时 H_CLK_SEL 无作用。

如果选择 8 位计数器产生串口的溢出时钟，即 BAND_SEL 为 1，那么计数器高低八位分别计数，以高 8 位为主，低 8 位为辅，高 8 位的计数时钟是系统时钟的 1/12 或 1/16(由 H_CLK_SEL 选择)，低 8 位的计数时钟是系统时钟的 div 分频，详见 BAUD_SEL0 为 1 时 timer0/ Timer1 定时器计数示意图。

Timer 工作流程：

1、配置 Timer0 和/或 Timer1 计数周期寄存器；

2、向 Timer 控制寄存器 TTCON 中的 bit0（load0）和/或 bit1（load1）写“1”，将 Timer 计数周期值真正配入硬件；

3、配置 Timer 模式寄存器 TMOD；

4、配置 Timer 控制寄存器 TTCON 并使能 Timer0 和/或 Timer1；

5、Timer 正常工作时检测是否有中断或定时器是否溢出，如果检测到中断或定时器溢出，清除中断标志位。

4.6、看门狗

看门狗模块采用异步设计，计数工作时钟为 400K 时钟。模块内部有一个定时器，当看门狗使能时，该定时器自动开始计数，如果定时器溢出，看门狗模块将复位 MCU 或产生定时溢出中断信号。

看门狗模块的寄存器全部分配在特殊功能寄存器中，软件应该通过特殊功能寄存器读写指令在控制看门狗模块。由于看门狗模块特殊功能寄存器为用户自定义特殊功能寄存器，芯片中无法支持 bit 寻址操作。

看门狗模块的计数时钟为 400KHz，在 400KHz 时钟域信号输出复位信号，在 CPU 休眠时仍可以产生 WDT_RST 信号，使得整个系统复位。

看门狗模块的计数器为 24 位计数器，高 16 位可配，模块根据寄存器配置的值，溢出产生溢出中断，或者溢出复位信号。

寄存器描述：

1）Watchdog 控制寄存器（WDCON）

SFR 地址 ：0xD8

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	WDT_IFG	R/W	Watchdog 中断标志位： 高有效，写 0 清中断
[6]	WDT_GATE_EN	R/W	Watchdog 门控使能位： 0：除能； 1：使能
[5]	WDT_EN	R/W	Watchdog 使能位： 0：除能； 1：使能
[4]	WDT_CLR	R/W	Watchdog 清零位： 软件写“1”，硬件自动清零

转下页

接上页

Bit	Name	R/W	Description
[3]	WDT_RE	R/W	Watchdog 复位使能位： 0：除能； 1：使能
[2]	WDT_IE	R/W	Watchdog 中断使能位： 0：除能； 1：使能
[1：0]			保留

补充说明：程序任何时候都能将 WDT_IE/ WDT_RE/ WDT_EN 写 1 操作。如果需要置 0，需要将 XDATA_CTRL 的寄存器 bit7 位写 1 后，才能清 0。清 0 后，须将 XDATA_CTRL 的 bit7 也清 0。

在系统中，WDT 复位的产生还需要配置时钟复位控制模块寄存器 SYS_REG1 的 WDT 的使能信号，在 CPU 休眠时，WDT 复位将复位整个系统，功能跟上电复位相同，在正常运行期间，WDT 复位将复位所有系统相关的接口模块。

2）Watchdog 计数寄存器 1（WDCNTM）

SFR 地址 ：0xD9

复位值 ：0x1A

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	WDT_CNTM	R/W	Watchdog 计数周期的 bit15~bit8

3）Watchdog 计数寄存器 2（WDCNTH）

SFR 地址 ：0xDA

复位值 ：0x06

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	WDT_CNTH	R/W	Watchdog 计数周期的 bit23~bit16

注：计数默认值为 1s

4）XDATA 总线的读写控制寄存器（XDATA_CTRL）

SFR 地址 ：0xDB

复位值 ：0x00

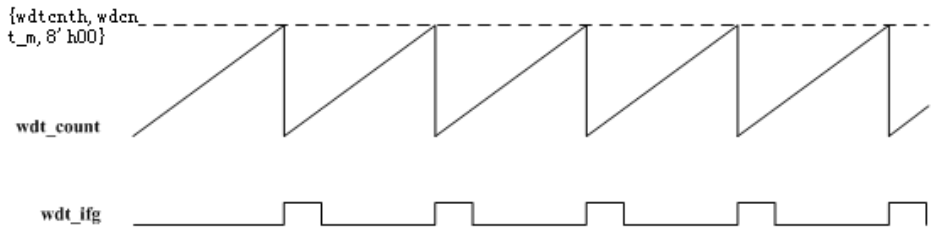
访问 ：R/W

Bit	Name	R/W	Description
[7]	WDT_PRT_CTRL	R/W	WDT_IE/WDT_RE/WDT_EN 寄存器位清零保护位： WDCON 寄存器中的 WDT_IE/WDT_RE/ WDT_EN 位在任何时候都能进行写 1 操作。如果需要置 0，需要将该位写 1 后才能清 0 清 0 后，须将该位也清 0
[6：1]			保留
[0]	XDATA_CTRL	R/W	XDATA 的 Memory 和 Reg 读写使能控制位： 0：允许读写； 1：不允许读写

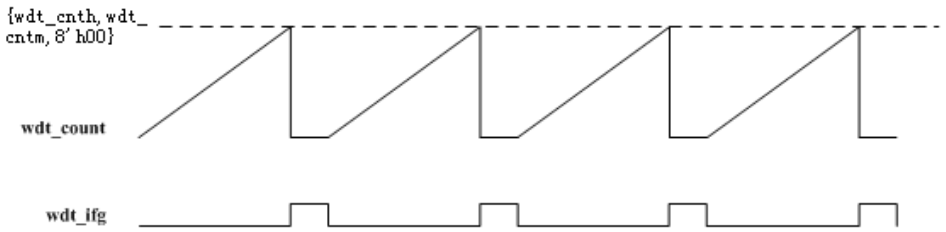
WDT_GATE_EN 为 Watchdog 定时器模式（WDR_RE 无效）下的门控使能位。当门控使能位为 1 时，如果 Watchdog 计数溢出，Watchdog 计数器将处于复位状态（计数值保持为零并停止计数）直到标志位 WDT_IFG 清零后继续开始计数（若 WDT_EN 保持有效）。当门控使能为低时，如果 Watchdog 计数溢出，Watchdog 继续进行下一轮计数（若 WDT_EN 保持有效），不管标志位 WDT_IFG 是否被清零。如果 Watchdog 复位使能有效，当 Watchdog 计数溢出时将不产生中断标志和中断信号，而是复位整个电路。

门控使能信号 WDT_GATE_EN 为低时可以实现精确的定时中断，因为软件在进行中断处理过程时，Watchdog 定时器已经开始下一轮计数。软件应根据需要自行决定是否使用门控使能功能。

门控使能无效时 Watchdog 定时器计数示意图：



门控使能有效时 Watchdog 定时器计数示意图：



工作流程

复位功能 Watchdog 工作流程：

1、配置 WDCON 寄存器并使能 Watchdog；

2、在 SYS_REG1 中配置 Watchdog 复位使能（如果 WDCON 中复位使能的话）；

3、软件定期清除 Watchdog，否则 Watchdog 定时溢出后会复位 MCU。

定时功能 Watchdog 工作流程：

1、配置 WDCON 寄存器并使能 Watchdog；

2、检测 Watchdog 中断或查询 Watchdog 是否溢出，如果检测到中断或定时器溢出，清除中断标志位。

4. 7、UART

UART 是一个全双工的串行口，可以用于网络通信，也可以实现串行异步通信，还可以作为同步移位寄存器使用。软件对串口的操作通过 SFR 寄存器来完成。

寄存器描述

1）寄存器（SCON）

SFR 地址 ：0x98

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	SM0	R/W	串行口方式选择位，两位组合决定了 4 中工作模式；
[6]	SM1	R/W	
[5]	SM2	R/W	多机通讯控制位（主要用于方式 2、3）： 1：接收方式时当第 9（RB9）位为 1 时，将数据送入 SBUF，并置位 RI 并产生中断，否则丢弃； 0：接收方式时不论第 9 位为 0 或 1，都将数据送入 SBUF，并置位 RI，产生中断
[4]	REN	R/W	允许接收位： 1：允许接收数据； 0：禁止接收数据
[3]	TB8	R/W	发送数据的第 9 位： 在方式 2、3 时，由用户通过软件设置，双机通讯，作为齐偶校验位使用，多机通信，用作地址（1）/数据（0）标志
[2]	RB8	R/W	接收数据的第 9 位： 在方式 2、3 时，RB8 为存放接收到的第 9 位数据，它代表接收数据的特征，可能是齐偶校验位，也可能是地址/数据的标志位

转下页

接上页

Bit	Name	R/W	Description
[1]	TI	R/W	发送中断标志位： 在方式 0 时，发送完 8 位后，硬件置位；在其他方式下，发送停止位，硬件置位； 由软件清 0
[0]	RI	R/W	接收中断标志位 在方式 0 时，接收完 8 位后，硬件置位，在其他方式下，收到停止位，硬件置位； 由软件清 0

SM0	SM1	工作方式	功　能	波特率
0	0	方式 0	扩展移位寄存器方式	f_{SYS} /12
0	1	方式 1	8 位 UART	定时器溢出率/n
1	0	方式 2	9 位 UART	f_{SYS}/32 或 f_{SYS}/64
1	1	方式 3	9 位 UART	定时器溢出率/n

2）寄存器（SBUF）

SFR 地址 ：0x99

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	SBUF_REG		发送和接收 8 位数据缓存： 串行发送时，MCU 向 SBUF 写入数据； 串行接收时，MCU 从 SBUF 读出数据

3）寄存器（SMOD）

SFR 地址 ：0x9A

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：1]			保留
[0]	SMOD	R/W	串行口波特率的倍增位： 1：串行口波特率加倍； 0：串行口波特率不加倍

串口工作流程：

1、选择工作模式，计算波特率；

2、根据工作模式的选择，配置 SCON；

3、根据波特率的要求，若需要 Timer0/Timer1 产生波特率，则配置 Timer0/Timer1 的工作模式；

4、配置 SMOD；

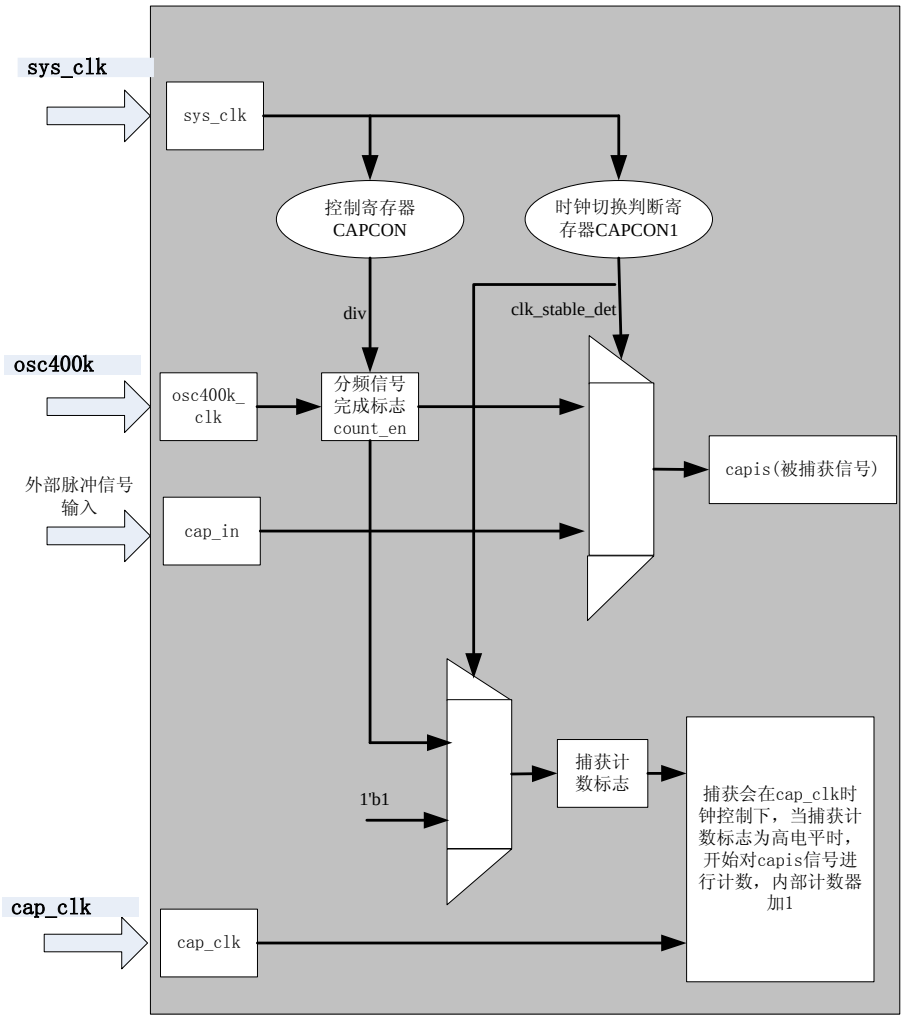
5、若数据接收，等待接收中断，从 SBUF 读取输入数据并清除中断后，执行下一次操作；若发送数据，写数据到 SBUF，等到发送中断达到后，执行下一次操作。

4. 8、捕获

该模块具有时钟稳定检测和外部脉冲信号捕获功能。时钟稳定检测主要用于在时钟切换时判定新打开时钟是否已经稳定。该模块内部有一个 16bits 的计数器，捕获功能分为 4 种模式：无捕获、上升沿捕获、下降沿捕获、上升沿/下降沿捕获。捕获结束后的计数值由 MCU 读出。

通过配置寄存器 CAPCON1 用户可以选择时钟稳定检测或外部脉冲信号捕获功能，配置 CAPCON 寄存器可以对捕获模式、信号分频等进行灵活选择。

该模块功能使用框图如下：



框图中左侧方向箭头所标时钟信号由时钟模块提供，其指向的内部信号 SYS_CLK、OSC400K_CLK 和 CAP_CLK 为 CAP_CTRL 相对应的时钟输入端口。

对外部脉冲输入信号 CAP_IN 进行捕获时，用户需将 CLK_STABLE_DET 配置为 0，被捕获信号 CAPIS 为 CAP_IN，此时 CAP_CLK（为 400KHz）会在 CAPCON 配置寄存器的控制下，在 COUNT_EN 分频完成脉冲标志（持续周期为 1 个 osc400k_clk）为高时对 CAPIS 进行捕获，并产生相应的中断和溢出标志。

使用时钟稳定检测功能时，用户需将 CLK_STABLE_DET 配置为 1，此时 CAP_CLK 为不稳定的 8MHz 时钟，被捕获信号 CAPIS 为 OSC400K_CLK（由时钟模块提供，频率为 400KHz）或其分频信号作为被捕获信号，CAP_CLK 在 CAPCON 配置寄存器控制下，捕获计数标志此时始终为高，即在 CAP_CLK 上升沿到来时便对 CAPIS 进行捕获，并产生相应的中断和溢出标志，MCU 通过多次读取和对比捕获的数值，如果数值稳定，则证明 8MHz 时钟已经稳定，可以切换。

寄存器描述

1）CAP 控制寄存器（CAPCON）

SFR 地址 ：0xF1

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	CAPOVFL	R/W	CAP 计数器溢出标志位： 高有效
[6]	CAPIFG	R/W	CAP 中断标志位： 高有效
[5]	CAPIE	R/W	CAP 中断使能位： 0：除能； 1：使能
[4]	CAP_EN	R/W	CAP 使能位： 0：除能； 1：使能
[3:2]	CAPM	R/W	捕获模式设置： 00：无捕获； 01：上升沿； 10：下降沿； 11：上升沿/下降沿
[1：0]	CLK_DIV	R/W	分频系数设置： 00： /1； 01： /2； 10： /4； 11： /8

2）CAP 控制寄存器（CAPCON1）

SFR 地址 ：0xF2

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：1]			保留
[0]	CLK_STABLE_DET	R/W	0：外部脉冲信号捕获 1：时钟稳定检测

3）CAP 计数寄存器 1（CAPCNTL）

SFR 地址 ： 0xF3

复位值 ： 0x00

访问 ： R/W

Bit	Name	R/W	Description
[7：0]	CAP_CNTL	R/W	Cap 计数周期的 bit7~bit0

4）CAP 计数寄存器 2（CAPCNTH）

SFR 地址 ： 0xF4

复位值 ： 0x00

访问 ： R/W

Bit	Name	R/W	Description
[7：0]	CAP_CNTH	R/W	Cap 计数周期的 bit15~bit8

工作流程

1、外部脉冲信号捕获功能：

（1）配置 PINSEL0 寄存器[6 ： 5]为 2’b11，选择 PA3 作为 CAPI 信号输入；

（2）配置 CAPCON1 和 CAPCON 控制寄存器，主要包括捕获使能、中断使能、模式选择（上升沿/下降沿/双边沿）和分频系数选择；

（3）MCU 检测捕获中断标志和溢出标志，当中断标志产生后可读取 CAPCNTH 和 CAPCNTL 寄存器中的计数值，并随后清除中断标志或溢出标志；

（4）当 MCU 再次检测到中断标志和溢出标志后，重复(3)；

（5）当捕获功能使用完毕，配置 CAPCON 寄存器，关闭捕获使能信号或写入默认值。

2、时钟稳定检测功能：

（1）将 CAPCON1 寄存器中的 CLK_STABLE_DET 配置为高，并配置 CAPCON 控制寄存器选择相应的模式，主要包括捕获使能、中断使能、模式选择（上升沿/下降沿/双边沿）和分频系数选择；

（2）MCU 检测捕获中断标志和溢出标志，当中断标志产生后可读取 CAPCNTH 和 CAPCNTL 寄存器中的计数值，并随后清除中断标志或溢出标志；

（3）当 MCU 再次检测到中断标志和溢出标志后，重复(3)；

（4）当判断 8MHz 时钟稳定时，关闭捕获使能信号或将 CAPCON 和 CAPCON1 寄存器写入默认值。

4. 9、GPIO

GPIO PORT 模块用于实现通用输入输出端口控制。当外部 PA0~PA7、PB0~PB3 端口选择选用 GPIO 的功能时，数据从对应的 IO 端口输入输出，分为 PA、PB 两组输入输出口，软件可以单独控制每组的每一位。

寄存器描述

1）PA 数据寄存器（PAREG）

SFR 地址 ：0xA1

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	PA_7_DATA	R/W	PA_7 输入/输出数据，当 PA7 选用 GPIO 功能时，数据从 PA7 端口输入输出： PA_7_DIR 为 1 时，PA_7_DATA 为 PA_7 输出数据； PA_7_DIR 为 0 时，PA_7_DATA 为 PA_7 输入数据
[6]	PA_6_DATA	R/W	PA_6 输出数据，当 PA6/RLED 端口选用 PA6 功能时，数据从该端口输出： PA_6_DIR 为 1 时，PA_6_DATA 为 PA_6 输出数据（注：该端口只有输出功能）
[5]	PA_5_DATA	R/W	PA_5 输入/输出数据，当 PA5 选用 GPIO 功能时，数据从 PA5 端口输入输出： PA_5_DIR 为 1 时，PA_5_DATA 为 PA_5 输出数据； PA_5_DIR 为 0 时，PA_5_DATA 为 PA_5 输入数据
[4]	PA_4_DATA	R/W	PA_4 输入/输出数据，当 PA4 选用 GPIO 功能时，数据从 PA4 端口输入输出： PA_4_DIR 为 1 时，PA_4_DATA 为 PA_4 输出数据； PA_4_DIR 为 0 时，PA_4_DATA 为 PA_4 输入数据
[3]	PA_3_DATA	R/W	PA_3 输入/输出数据，当 PA3 选用 GPIO 功能时，数据从 PA3 端口输入输出： PA_3_DIR 为 1 时，PA_3_DATA 为 PA_3 输出数据； PA_3_DIR 为 0 时，PA_3_DATA 为 PA_3 输入数据
[2]	PA_2_DATA	R/W	PA_2 输入/输出数据，当 PA2 选用 GPIO 功能时，数据从 PA2 端口输入输出： PA_2_DIR 为 1 时，PA_2_DATA 为 PA_2 输出数据； PA_2_DIR 为 0 时，PA_2_DATA 为 PA_2 输入数据
[1]	PA_1_DATA	R/W	PA_1 输入/输出数据，当 PA1 选用 GPIO 功能时，数据从 PA1 端口输入输出： PA_1_DIR 为 1 时，PA_1_DATA 为 PA1 输出数据； PA_1_DIR 为 0 时，PA_1_DATA 为 PA1 输入数据
[0]	PA_0_DATA	R/W	PA_0 输入/输出数据，当 PA0 选用 GPIO 功能时；数据从 PA0 端口输入输出： PA_0_DIR 为 1 时，PA_0_DATA 为 PA0 输出数据； PA_0_DIR 为 0 时，PA_0_DATA 为 PA0 输入数据

2）PA 数据方向寄存器（PADIR）

SFR 地址 ： 0xA2

复位值 ： 0x00

访问 ： R/W

Bit	Name	R/W	Description
[7]	PA_7_DIR	R/W	PA_7 数据输入输出方向： 1：PA_7 为数据输出口； 0：PA_7 为数据输入口
[6]	PA_6_DIR	R/W	PA_6 数据输入输出方向： 1：PA_6 为数据输出口； 0：无效
[5]	PA_5_DIR	R/W	PA_5 数据输入输出方向： 1：PA_5 为数据输出口； 0：PA_5 为数据输入口
[4]	PA_4_DIR	R/W	PA_4 数据输入输出方向： 1：PA_4 为数据输出口； 0：PA_4 为数据输入口
[3]	PA_3_DIR	R/W	PA_3 数据输入输出方向： 1：PA_3 为数据输出口； 0：PA_3 为数据输入口
[2]	PA_2_DIR	R/W	PA_2 数据输入输出方向： 1：PA_2 为数据输出口； 0：PA_2 为数据输入口
[1]	PA_1_DIR	R/W	PA_1 数据输入输出方向： 1：PA_1 为数据输出口； 0：PA_1 为数据输入口
[0]	PA_0_DIR	R/W	PA_0 数据输入输出方向： 1：PA_0 为数据输出口； 0：PA_0 为数据输入口

3）PA 中断使能寄存器（PAIE）

SFR 地址 ： 0xA3

复位值 ： 0x00

访问 ： R/W

Bit	Name	R/W	Description
[7]	PA_7_IE	R/W	PA_7 中断使能信号： 1：使能； 0：除能
[6]	PA_6_IE	R/W	无效
[5]	PA_5_IE	R/W	PA_5 中断使能信号： 1：使能； 0：除能

转下页

接上页

Bit	Name	R/W	Description
[4]	PA_4_IE	R/W	PA_4 中断使能信号： 1：使能； 0：除能
[3]	PA_3_IE	R/W	PA_3 中断使能信号： 1：使能； 0：除能
[2]	PA_2_IE	R/W	PA_2 中断使能信号： 1：使能； 0：除能
[1]	PA_1_IE	R/W	PA_1 中断使能信号： 1：使能； 0：除能
[0]	PA_0_IE	R/W	PA_0 中断使能信号： 1：使能； 0：除能

注意：只有在端口方向为输入时，当中断条件满足了，PA 才会产生中断信号。

4）PA 中断方式选择寄存器（PAIES）

SFR 地址 ：0xA4

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	PA_7_IES	R/W	PA_7 中断方式选择信号： 0：上升沿中断； 1：下降沿中断
[6]	PA_6_IES	R/W	无效
[5]	PA_5_IES	R/W	PA_5 中断方式选择信号： 0：上升沿中断； 1：下降沿中断
[4]	PA_4_IES	R/W	PA_4 中断方式选择信号： 0：上升沿中断； 1：下降沿中断
[3]	PA_3_IES	R/W	PA_3 中断方式选择信号： 0：上升沿中断； 1：下降沿中断
[2]	PA_2_IES	R/W	PA_2 中断方式选择信号： 0：上升沿中断； 1：下降沿中断
[1]	PA_1_IES	R/W	PA_1 中断方式选择信号： 0：上升沿中断； 1：下降沿中断

转下页

接上页

Bit	Name	R/W	Description
[0]	PA_0_IES	R/W	PA_0 中断方式选择信号： 0：上升沿中断； 1：下降沿中断

注意：只有在端口方向为输入时，当中断条件满足了，PA 才会产生中断信号.

5）PA 中断标志寄存器（PAIFG）

SFR 地址 ：0xA5

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	PA_7_IFG	R/W	PA_7 中断标志信号，高有效
[6]	PA_6_IFG	R/W	无效
[5]	PA_5_IFG	R/W	PA_5 中断标志信号，高有效
[4]	PA_4_IFG	R/W	PA_4 中断标志信号，高有效
[3]	PA_3_IFG	R/W	PA_3 中断标志信号，高有效
[2]	PA_2_IFG	R/W	PA_2 中断标志信号，高有效
[1]	PA_1_IFG	R/W	PA_1 中断标志信号，高有效
[0]	PA_0_IFG	R/W	PA_0 中断标志信号，高有效

注意：只有在端口方向为输入时，当中断条件满足了，PA 才会产生中断信号。

6）PB 数据寄存器（PBREG）

SFR 地址 ：0xA9

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：4]			保留
[3]	PB_3_DATA	R/W	PB_3 输入/输出数据，当 PB3 选用 GPIO 功能时，数据从 PB3 端口输入输出： PB_3_DIR 为 1 时，PB_3_DATA 为 PB_3 输出数据； PB_3_DIR 为 0 时，PB_3_DATA 为 PB_3 输入数据
[2]	PB_2_DATA	R/W	PB_2 输入/输出数据，当 PB2 选用 GPIO 功能时，数据从 PB2 端口输入输出： PB_2_DIR 为 1 时，PB_2_DATA 为 PB_2 输出数据； PB_2_DIR 为 0 时，PB_2_DATA 为 PB_2 输入数据
[1]	PB_1_DATA	R/W	PB_1 输入/输出数据，当 PB1 选用 GPIO 功能时，数据从 PB1 端口输入输出： PB_1_DIR 为 1 时，PB_1_DATA 为 PB1 输出数据； PB_1_DIR 为 0 时，PB_1_DATA 为 PB1 输入数据
[0]	PB_0_DATA	R/W	PB_0 输入/输出数据，当 PB0 选用 GPIO 功能时，数据从 PB0 端口输入输出： PB_0_DIR 为 1 时，PB_0_DATA 为 PB0 输出数据； PB_0_DIR 为 0 时，PB_0_DATA 为 PB0 输入数据

7）PB 数据方向寄存器（PBDIR）

SFR 地址 ：0xAA

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：4]			保留
[3]	PB_3_DIR	R/W	PB_3 数据输入输出方向： 1：PB_3 为数据输出口； 0：PB_3 为数据输入口
[2]	PB_2_DIR	R/W	PB_2 数据输入输出方向： 1：PB_2 为数据输出口； 0：PB_2 为数据输入口
[1]	PB_1_DIR	R/W	PB_1 数据输入输出方向： 1：PB_1 为数据输出口； 0：PB_1 为数据输入口
[0]	PB_0_DIR	R/W	PB_0 数据输入输出方向： 1：PB_0 为数据输出口； 0：PB_0 为数据输入口

8）PB 中断使能寄存器（PBIE）

SFR 地址 ：0xAB

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：4]			保留
[3]	PB_3_IE	R/W	PB_3 中断使能信号： 1：使能； 0：除能
[2]	PB_2_IE	R/W	PB_2 中断使能信号： 1：使能； 0：除能
[1]	PB_1_IE	R/W	PB_1 中断使能信号： 1：使能； 0：除能
[0]	PB_0_IE	R/W	PB_0 中断使能信号： 1：使能； 0：除能

注意：只有在端口方向为输入时，当中断条件满足了，PB 才会产生中断信号。

9）PB 中断方式选择寄存器（PBIES）

SFR 地址 ：0xAC

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：4]			保留
[3]	PB_3_IES	R/W	PB_3 中断方式选择信号： 0：上升沿中断； 1：下降沿中断
[2]	PB_2_IES	R/W	PB_2 中断方式选择信号： 0：上升沿中断； 1：下降沿中断
[1]	PB_1_IES	R/W	PB_1 中断方式选择信号： 0：上升沿中断； 1：下降沿中断
[0]	PB_0_IES	R/W	PB_0 中断方式选择信号： 0：上升沿中断； 1：下降沿中断

注意：只有在端口方向为输入时，当中断条件满足了，PB 才会产生中断信号.

10）PB 中断标志寄存器（PBIFG）

SFR 地址 ：0xAD

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：4]			保留
[3]	PB_3_IFG	R/W	PB_3 中断标志信号，高有效
[2]	PB_2_IFG	R/W	PB_2 中断标志信号，高有效
[1]	PB_1_IFG	R/W	PB_1 中断标志信号，高有效
[0]	PB_0_IFG	R/W	PB_0 中断标志信号，高有效

注意：只有在端口方向为输入时，当中断条件满足了，PB 才会产生中断信号。

4. 10、端口控制

PA0~PA7、PB0~PB3 引脚是由寄存器控制选择的多功能复用引脚，但是一个时间只能选择一种功能，这些功能的选择是通过端口控制来实现的。

端口控制寄存器：

地址	寄存器名称	有效位	访问	复位值
0x0270	PINSEL0	8	R/W	0x05
0x0271	PINSEL1	3	R/W	0x00
0x0272	PAPU	6	R/W	0x3F
0x0273	PAPD	6	R/W	0x00
0x0274	PBPD	5	R/W	0x00
0x0275	PBPU	5	R/W	0x1F

寄存器描述

1）PINSEL0 寄存器

地址 ：0x0270

复位值 ：0x05

访问 ：R/W

Bit	Name	R/W	Description
[7]	PA4	R/W	PA4 功能选择： 1：温度检测输入； 0：通用 IO 功能使能
[6：5]	PA3	R/W	PA3 功能选择： 11：PA3 作为 CAPIS 输入； 10：PA3 作为 AN0 输入； 01：PA3 作为 RXD 输入； 00：通用 IO
[4：3]	PA2	R/W	PA2 功能选择： 11：PA2 作为 AN2 输入； 10：PA2 作为 PWM2 输出，开漏输出； 01：PA2 作为 TXD 输出； 00：通用 IO
[2：1]	PA1	R/W	PA1 功能选择： 11：保留； 10：PA1 作为 IRED2 模拟输出； 01：PA1 作为 PWM1 输出； 00：通用 IO
[0]	PA0	R/W	PA0 功能选择： 1：默认 PA0 作为 PWM0 输出，开漏输出； 0：通用 IO

2）PINSEL1 寄存器

地址 ：0x0271

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：3]			保留
[2]	PA6	R/W	PA6 功能选择： 1：PWM3 输出； 0：通用 IO 输出
[1]	PB3	R/W	PB3 功能选择： 1：PB3 作为 AN3 输入； 0：通用 IO
[0]	PA5	R/W	PA5 功能选择： 1：PA5 作为 AN1SW 输出，开漏输出； 0：通用 IO

3）PAPU 寄存器

地址 ：0x0272

复位值 ：0x3f

访问 ：R/W

Bit	Name	R/W	Description
[7：6]			保留
[5]	PAPU5	R/W	PA5 上拉配置： 1：上拉； 0：不上拉
[4]	PAPU4	R/W	PA4 上拉配置： 1：上拉； 0：不上拉
[3]	PAPU3	R/W	PA3 上拉配置： 1：上拉； 0：不上拉
[2]	PAPU2	R/W	PA2 上拉配置： 1：上拉； 0：不上拉
[1]	PAPU1	R/W	PA1 上拉配置： 1：上拉； 0：不上拉
[0]	PAPU0	R/W	PA0 上拉配置： 1：上拉； 0：不上拉

4）PAPD 寄存器

地址 ：0x0273

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：6]			保留
[5]	PAPD5	R/W	PA5 上拉配置： 1：下拉； 0：不下拉
[4]	PAPD4	R/W	PA4 上拉配置： 1：下拉； 0：不下拉
[3]	PAPD3	R/W	PA3 上拉配置： 1：下拉； 0：不下拉
[2]	PAPD2	R/W	PA2 上拉配置： 1：下拉； 0：不下拉
[1]	PAPD1	R/W	PA1 上拉配置： 1：下拉； 0：不下拉
[0]	PAPD0	R/W	PA0 上拉配置： 1：下拉； 0：不下拉

5）PBPD 寄存器

地址 ：0x0274

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：5]			保留
[4]	PBPD3	R/W	PB3 下拉配置： 1：下拉； 0：不下拉
[3]	PBPD2	R/W	PB2 下拉配置： 1：下拉； 0：不下拉
[2]	PBPD1	R/W	PB1 下拉配置： 1：下拉； 0：不下拉

转下页

接上页

Bit	Name	R/W	Description
[1]	PBPD0	R/W	PB0 下拉配置： 1：下拉； 0：不下拉
[0]	PAPD7	R/W	PA7 下拉配置： 1：下拉； 0：不下拉

6）PBPU 寄存器

地址 ：0x0275

复位值 ：0x1f

访问 ：R/W

Bit	Name	R/W	Description
[7：5]			保留
[4]	PBPU3	R/W	PB3 上拉配置： 1：上拉； 0：不上拉
[3]	PBPU2	R/W	PB2 上拉配置： 1：上拉； 0：不上拉
[2]	PBPU1	R/W	PB1 上拉配置： 1：上拉； 0：不上拉
[1]	PBPU0	R/W	PB0 上拉配置： 1：上拉； 0：不上拉
[0]	PAPU7	R/W	PA7 上拉配置： 1：上拉； 0：不上拉

MCU 设置\读取管脚控制模块寄存器工作流程：

（1）PA0 端口默认作 PWM0 开漏输出功能，PA1 端口默认作 IRED2 模拟输出功能，PA2~PA5、PA7、PB0~PB3 默认为 GPIO 口，且默认为上拉；

（2）MCU 根据需要配置管脚功能选择寄存器 PINSEL0 和 PINSEL1，并可通过上下拉配置寄存器（PAPU、PAPD、PBPU 和 PBPD）对各端口进行上下拉配置；

（3）当作为 GPIO，端口方向设置要通过 PADIR/PBDIR 寄存器配置为输入或输出，作为复用功能时方向默认绑定。

4. 11、FLASH 控制模块

FLASH 作为程序存储器，上电后 MCU 自动执行 FLASH 存放的程序，期间 MCU 程序可以通过控制 DRAM 控制寄存器实现 Flash 数据存储区和 MCU 数据 RAM 的数据交互。FLASH 控制寄存器：

地址	寄存器名称	有效位	访问	复位值
0x0220	DRAM_CTRL	4	R/W	0x00
0x0221	FLASH_PROTECT_REG	8	R/W	0x00
0x0222	FM_ERS_PRG_TIME_L	8	R/W	0xF0
0x0223	FM_ERS_PRG_TIME_H	8	R/W	0x55

寄存器描述

1）DRAM 控制寄存器（DRAM_CTRL）

地址 ：0x0220

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：6]			保留
[5]	FM_OEN	R/W	FLASH 数据搬到 DRAM 使能位： 1：将 FLASH 数据搬到 DRAM； 0：无操作
[4]	FM_IEN	R/W	DRAM 数据搬到 FLASH 使能位： 1：将 DRAM 数据搬到 FLASH； 0：无操作
[3：2]			保留
[1]	DM_PG1	R/W	Page1 或 256dram 的后 128bytes 选择控制位： 1：选择； 0：不选择
[0]	DM_PG0	R/W	Page0 或 256dram 的前 128bytes 选择控制位： 1：选择； 0：不选择

2）FLASH_PROTECT_REG

地址 ：0x0221

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	FLASHCTRL_PROTECT_REG	R/W	当 MCU 将 0xA5 写入该寄存器后，对 DRAM_CTRL 寄存器写入的值才有效

3）FM_ERS_PRG_TIME_L

地址 ：0x0222

初始值 ：0xf0
访问 ：R/W

Bit	Name	R/W	Description
[7：0]	FM_ERS_PRG_TIME_L	R/W	FLASH 烧写时间寄存器低八位

4）FM_ERS_PRG_TIME_H
地址 ：0x0223
复位值 ：0x55
访问 ：R/W

Bit	Name	R/W	Description
[7：0]	FM_ERS_PRG_TIME_H	R/W	FLASH 烧写时间寄存器高八位

FLASH 烧写时间寄存器的复位值为 0x55F0，FLASH 烧写的时间要求如下，

Mode	Min	Typ	Max
Page erase time	4.750ms	5.000ms	6.250ms
Page program time	4.750ms	5.000ms	6.250ms

若考虑到时钟偏差的情况下，配置值为 4.750ms~6.250ms 的中间值为 5.5ms

在不同时钟时 FM_ERS_PRG_TIME 配置值如下：

时钟	计算方式	配置值
4MHz	5500μs/0.25	0x55f0
2MHz	5500μs/0.5	0x2af8
1MHz	5500μs/1	0x157c
500KHz	5500μs/2	0xabe
400KHz	5500μs/2.5	0x898
550KHz	5500μs/0.19	0xbd1

用户控制 FLASH 和 DRAM 的数据交换流程：

1、用户根据需要配置 DRAM_CTRL 寄存器，寄存器描述如下；

2、FM_OEN 和 FM_IEN 只能使能其中一个，若同时使能则执行 FM_OEN 操作，可同时选择 Page0 和 Page1，也可只选其中一个 Page（128bytes）；

3、用户发 Stop 命令：将特殊功能寄存器 PCON 的 bit1 写 1，发 Stop 后需增加 4 条 NOP 指令。硬件电路将让 CPU 休眠（暂停工作），硬件电路根据 MCU 配置的寄存器值自动完成数据搬运；

4、硬件电路发 Run 信号，唤醒 MCU；

5、用户将 Dram 控制寄存器清 0，MCU 继续其他操作。

说明：

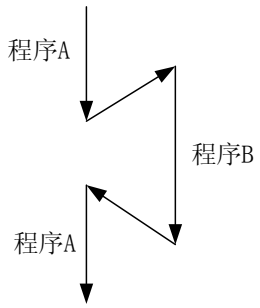
1）在系统中 Flash 和 Dram 数据交换的优先级最高，数据交换过程中，有外部中断产生将延迟唤醒 MCU（直到数据交换结束）；

2）将 Dram 256byte 数据全部烧到 FLASH，需要约 25ms。如果选择烧写 128bytes 到 FLASH，时间减半。将 FLASH 的 256bytes 数据读出写到 DRAM，需要约 65μs，同样，如果是 128byts，时间减半。

因为在烧写过程中，不允许 WDT 对接口进行复位，所以在执行这个数据交换之前，软件需要清 WDT。

4. 12、中断

所谓中断，即打断 MCU 目前正在执行的程序而转向执行其他程序的一种 MCU 运行机制。MCU 正在执行程序 A，但是因为某些事件的发生使得 MCU 不得不暂时放弃目前的程序 A 而转向执行程序 B，在程序 B 执行完成之后再重新回来执行程序 A。这就是一个中断过程，其中程序 A 叫做被中断程序，而程序 B 叫做中断服务程序，引起中断的事件叫做中断源。芯片内部许多部件都可以作为中断源，比如 UART、定时器等。下图为 MCU 中断示意图：



FLASH 控制寄存器：

地址	寄存器名称	有效位	访问	复位值
0x0280	IRQSTATL	7	R	0x00
0x0281	IRQSTATH	5	R	0x00
0x0282	INTML	7	R/W	0xff
0x0283	INTMH	5	R/W	0xff
0x0284	IRQ_MSKDL	7	R	0x00
0x0285	IRQ_MSKDH	5	R	0x00
0x0286	INTSEL0L	7	R/W	0x00
0x0287	INTSEL0H	5	R/W	0x00

寄存器描述：

1）中断请求寄存器 1（IRQSTATL）

地址　　：0x0280

复位值　：0x00

访问　　：R

Bit	Name	R/W	Description
[7]	ADDR_IRQ	R	无效地址中断请求标志： 无效地址产生系统复位
[6]	LN24_IRQ	R	总线供电及通讯模块的中断请求标志： 通过清除 LN24STAT1 中的所有有效中断来使 LN24_IRQ 复位

转下页

接上页

Bit	Name	R/W	Description
[5]	ME_IRQ	R	烟雾检测模块的中断请求标志： 该位映射自 MESTAT.ME_IRQ，通过读取任一个 ADC_VAL 寄存器或向 MESTAT.ME_IRQ 写 0b 来清除该位
[4]	GPIO_IRQ	R	GPIO 的组合中断请求标志： 通过清除 PAIFG 和 PBIFG 寄存器中的有效中断来使其复位
[3]	TIM_IRQ	R	定时器模块的组合中断请求标志： 通过清除 TTCON 中的有效中断来使其复位
[2]	LN24_LTH_IRQ	R	总线电压监测模块的中断请求标志： 通过清除 LN24_LVRST_REG 来使其复位
[1]			保留
[0]	ADC_IRQ	R	ADC 的中断请求标志： 该位映射自MESTAT.ADC_IRQ，通过读取任意 ADC_VAL来清除该位

2）中断请求寄存器 2（IRQSTATH）

地址　　：0x0281

复位值　：0x00

访问　　：R

Bit	Name	R/W	Description
[7：6]			保留
[5]	CAP_OVFL_IRQ	R	捕获溢出中断请求标志： 通过清除 CAPCON[7]的中断来使其复位
[4]	CAP_IRQ	R	捕获中断请求标志： 通过清除 CAPCON[6]的中断来使其复位
[3]			保留
[2]	TRIMF_ERR	R	来自 FLASH 接口的中断请求标志
[1]	UART_IRQ	R	来自串口的组合中断请求标志： 通过清除 SCON 寄存器中的有效中断来使其复位
[0]	WDT_IRQ	R	看门狗功能复位中断请求标志 通过清除WDCON寄存器中的有效中断来使其复位

3）中断屏蔽寄存器 1（INTML）

地址 ：0x0282

复位值 ：0xff

访问 ：R/W

Bit	Name	R/W	Description
[7]	ADDR_IM	R/W	ADDR_IRQ 的中断请求屏蔽： 1：被屏蔽； 0：未被屏蔽
[6]	LN24_IM	R/W	LN24_IRQ 的中断请求屏蔽： 1：被屏蔽； 0：未被屏蔽
[5]	ME_IM	R/W	ME_IRQ 的中断请求屏蔽： 1：被屏蔽； 0：未被屏蔽
[4]	GPIO_IM	R/W	GPIO_IRQ 的中断请求屏蔽： 1：被屏蔽； 0：未被屏蔽
[3]	TIM_IM	R/W	TIM_IRQ 的中断请求屏蔽： 1：被屏蔽； 0：未被屏蔽
[2]	LN24_LTH_IM	R/W	LN24_LTH_IRQ 的中断请求屏蔽： 1：被屏蔽； 0：未被屏蔽
[1]			保留
[0]	ADC_IM	R/W	ADC_IRQ 的中断请求屏蔽： 1：被屏蔽； 0：未被屏蔽

4）中断屏蔽寄存器 2（INTMH）

地址 ：0x283

复位值 ：0xff

访问 ：R/W

Bit	Name	R/W	Description
[7：6]			保留
[5]	CAP_OVFL_IM	R/W	CAP_OVFL_IRQ 的中断请求屏蔽： 1：被屏蔽； 0：未被屏蔽
[4]	CAP_IM	R/W	CAP_IRQ 的中断请求屏蔽： 1：被屏蔽； 0：未被屏蔽
[3]			保留

转下页

接上页

Bit	Name	R/W	Description
[2]	TRIMF_IM	R/W	TRIMF_ERR 的中断请求屏蔽： 1：被屏蔽； 0：未被屏蔽
[1]	UART_IM	R/W	UART_IRQ 的中断请求屏蔽： 1：被屏蔽； 0：未被屏蔽
[0]	WDT_IM	R/W	WDT_IRQ 的中断请求屏蔽： 1：被屏蔽； 0：未被屏蔽

5）屏蔽中断寄存器 1（IRQ_MSKDL）

地址 ：0x0284

复位值 ：0x00

访问 ：R

Bit	Name	R/W	Description
[7]	ADDR_INT	R	无效地址中断 =DDR_IRQ&~ADDR_IM
[6]	LN24_INT	R	总线供电及通讯模块中断 =N24_IRQ&~LN24_IM
[5]	ME_INT	R	烟雾检测模块中断 =E_IRQ&~ME_IM
[4]	GPIO_INT	R	GPIO 中断 =PIO_IRQ&~GPIO_IM
[3]	TIM_INT	R	定时器中断 =IM_IRQ&~TIM_IM
[2]	LN24_LTH_INT	R	总线电压监测中断 =N24_LTH_IRQ&~LN24_LTH_IM
[1]			保留
[0]	ADC_INT	R	ADC 中断 =DC_IRQ&~ADC_IM

6）屏蔽中断寄存器 2（IRQ_MSKDH）

地址 ：0x0285

复位值 ：0x00

访问 ：R

Bit	Name	R/W	Description
[7：6]			保留
[5]	CAP_OVFL_INT	R	CAP 溢出中断 =CAP_OVFL_IRQ&~CAP_OVFL_IM

转下页

接上页

Bit	Name	R/W	Description
[4]	CAP_INT	R	CAP 中断 =CAP_IRQ&~CAP_IM
[3]			保留
[2]	TRIMF_INT	R	FLASH 中断 =TRIMF_ERR&~TRIMF_IM
[1]	UART_INT	R	串口中断 =UART_IRQ&~ UART_IM
[0]	WDT_INT	R	看门狗中断 =WDT_IRQ&~WDT_IM

7）中断输出口选择寄存器 1（INTSEL0L）

地址 ：0x0286

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	ADDR_IS	R/W	ADDR_INT 的中断输出口选择： 1：从 INT0 输出； 0：从 INT1 输出
[6]	LN24_IS	R/W	LN24_INT 的中断输出口选择： 1：从 INT0 输出； 0：从 INT1 输出
[5]	ME_IS	R/W	ME_INT 的中断输出口选择： 1：从 INT0 输出； 0：从 INT1 输出
[4]	GPIO_IS	R/W	GPIO_INT 的中断输出口选择： 1：从 INT0 输出； 0：从 INT1 输出
[3]	TIM_IS	R/W	TIM_INT 的中断输出口选择： 1：从 INT0 输出； 0：从 INT1 输出
[2]	LN24_LTH_IS	R/W	LN24_ LTH _INT 的中断输出口选择： 1：从 INT0 输出； 0：从 INT1 输出
[1]			保留
[0]	ADC_IS	R/W	ADC_INT 的中断输出口选择： 1：从 INT0 输出； 0：从 INT1 输出

8）中断输出口选择寄存器 2（INTSEL0H）

地址 ：0x0287

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：6]			保留
[5]	CAP_OVFL_IS	R/W	CAP_OVFL_INT 的中断输出口选择： 1：从 INT0 输出； 0：从 INT1 输出
[4]	CAP_IS	R/W	CAP_INT 的中断输出口选择： 1：从 INT0 输出； 0：从 INT1 输出
[3]			保留
[2]	TRIMF_IS	R/W	TRIMF_INT 的中断输出口选择： 1：从 INT0 输出； 0：从 INT1 输出
[1]	UART_IS	R/W	UART_INT 的中断输出口选择： 1：从 INT0 输出； 0：从 INT1 输出
[0]	WDT_IS	R/W	WDT _INT 的中断输出口选择： 1：从 INT0 输出； 0：从INT1输出

MCU 设置中断控制模块寄存器工作流程：

1、中断屏蔽寄存器默认为 0x1FFF，所有中断被屏蔽；

2、MCU 根据需要配置中断屏蔽寄存器 INTMH、INTML，选通对应位中断；

3、默认所有中断从 INT1 中断端口输出，MCU 根据需要配置中断输出口选择寄存器 INTSEL0H、INTSEL0L，可配置对应位使中断从 INT0 端口输出，否则中断从 INT1 输出；

4、MCU 通过读中断请求寄存器 IRQSTATH、IRQSTATL 和屏蔽后中断寄存器 IRQ_MSKDH、IRQ_MSKDL 查询中断源，从而执行对应的中断服务程序。

4. 13、LN24 总线电压监测

LN24 总线电压正常为 24V，当其电压降至 7V 时会产生中断信号通知 MCU。总线电压监测寄存器：

地址	寄存器名称	有效位	访问	复位值
0x02D0	LVR_REG	3	R/W	0x60
0x02D3	LN24_IE_REG	1	R/W	0x00
0x02D4	LN24_LVRST_REG	2	R/W	0x00

寄存器描述

1）LVR_REG

地址 ：0x02D0

复位值 ：0x60

访问 ：R/W

Bit	Name	R/W	Description
[7：5]	LVR_SEL[2：0]	R/W	AVDD 低压复位点选择寄存器： 000：3.8V； 011：2.5V（初始值）； 101：4.1V
[4：0]			保留，只能写 0

2）LN24_IE_REG

地址 ：0x02D3

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：1]			保留
[0]	LN24_LTH_IE	R/W	LN24 低压检测中断使能信号

3）LN24_LVRST_REG

地址 ：0x02D4

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：4]			保留
[3]	LN24_LTH_IRQ	R/W	LN24 低压检测中断标志，写 0 清中断
[2：1]			保留
[0]	LN24_LTH_OUT	R	LN24 低压检测输出信号

4. 14、烟雾检测

烟雾检测时，先由电路红外发射端 IRED 控制的红外发光二极管周期性的发射红外线，红外接收二极管检测光电仓内烟雾颗粒散射来的红外线强度，此时如果有烟雾，光电二极管会微弱导通，其电流通过端口 IR+和 IR-进入光电 I-V 放大电路转换成电压信号，积分器的增益可以通过寄存器设置，输出电压信号通过一个 10 位的 ADC 转换为 10 位代表烟雾浓度的数字信号，用户可通过寄存器的设置对数字信号进行比较处理，得到烟雾检测结果。

对红外接收二极管的微弱电流提供两种处理方式，I-V 放大（默认）和积分放大。可通过 AMPCONF 寄存器配置。

4. 14. 1、检测控制单元

检测控制单元（MEAS_CTRL）作用是控制总线电压、温度、暗电流和烟雾的 ADC 检测。根据 MCU 的不同配置，在每一次测试循环中控制 IRED 模块、ADC 模块和放大模块的各种工作状态；及时保存 ADC 值到对应的寄存器并生成对应的中断信号。

根据 MCU 的配置和烟雾测试的特点，MEAS_CTRL 模块主要包括三个工作状态、四种工作模式和两种放大方式。

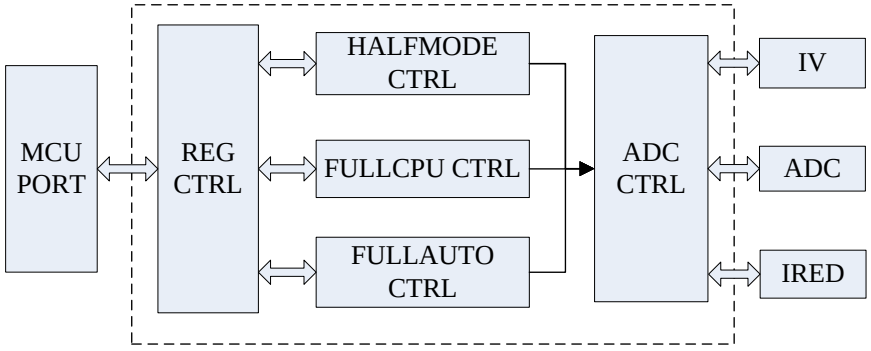
各状态方式介绍：

三种工作状态即 MEAS_CTRL 模块内部的工作状态，分别是 IDLE 状态、INITIAL 状态和 ACTIVE 状态。其中 IDLE 状态主要用于系统的休眠、等待；INITIAL 状态主要用于系统的初始化和半 MCU/半自动模式下的总线电压和温度的测试；ACTIVE 状态是指系统休眠初始化后唤醒 MCU 后的测试状态。

四种工作模式是根据一个测试循环中 MCU 的参与程度来划分的，分别为全 MCU 模式、全自动模式、半 MCU 模式和半自动模式

两种放大方式指的是在对暗电流和烟雾进行 ADC 采样时采取的不同放大方式，分别是 IV 放大和积分放大

根据本模块的工作特点，MEAS_CTRL 模块的内部结构示意图如下所示：



各模块功能介绍：

REG_CTRL：实现控制寄存器的读写，完成控制寄存器的读写操作以及某些寄存器位的跨时钟操作。

FULLAUTO_CTRL：全自动模式控制单元，该单元根据寄存器的配置主要完成全自动模式下各接口信号的自动产生。

FULLMCU_CTRL：全 MCU 模式控制单元，主要完成全 MCU 模式、半 MCU 模式和半自动模式下，唤醒 MCU 后的检测时序。

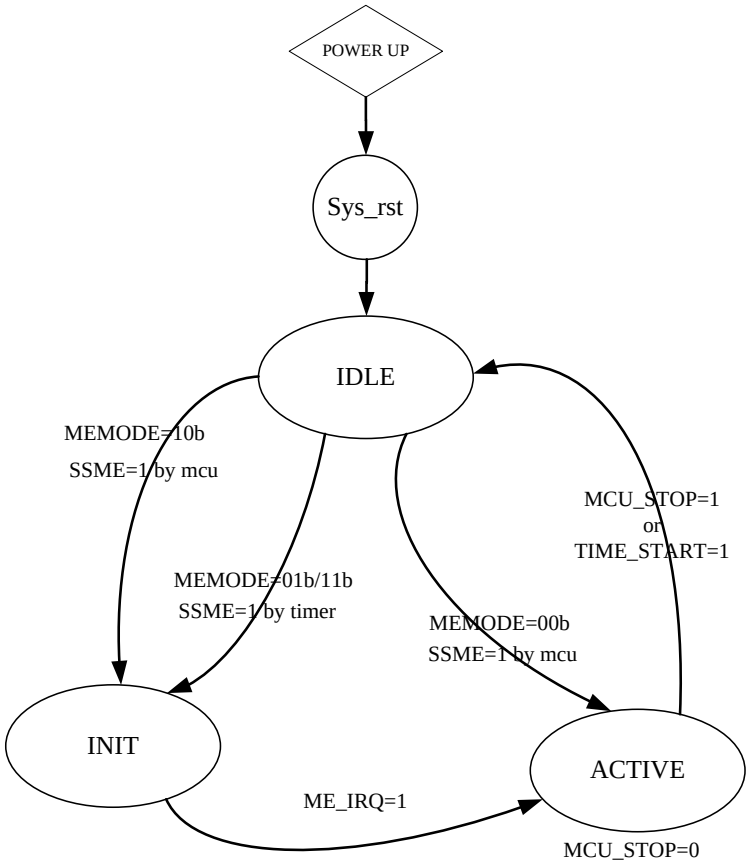
HALFMODE_CTRL：半 MCU/半自动控制单元，主要完成半 MCU 和半自动模式下，在 INITIAL 状态里进行 LN24 和温度检测时的时序控制。

ADC_CTRL：ADC 控制单元，根据不同工作模式产生的控制信号，生成 ADC_SOC 信号、IV 放大控制信号、积分放大控制信号和 IRED 控制信号。

MEAS_CTRL 模块通过一个有限状态机来实现检测时序的自动控制，该状态机有三个状态，分别是 IDLE、INITIAL 和 ACTIVE 状态。

下面分别对它们进行介绍：

MEAS_CTRL 模块主状态示意图：



IDLE 状态

复位后，主状态机处于 IDLE 状态。在全自动/半自动/半 MCU 工作模式下，IDLE 状态不进行 ADC 检测；在全 MCU 模式下，是否进行 ADC 检测由 MCU 配置的寄存器值进行决定。

在一个测试循环中，IDLE 状态主要用于系统休眠，降低功耗。当 TIMERCTRL.MEAS_START 置为 1 时，内部计数器开始工作，进入新一轮的测试循环。根据 MECONF.MEMODE 设置的不同（即选择不同的烟雾检测模式），不同的事件将引发 IDLE 状态转换为 INITIAL 状态或者 ACTIVE 状态。

各个工作模式状态的转换条件如下：

全 MCU/半 MCU 模式：MCU 模式下，当 TIMERCTRL.EN_TIMER_N 为 1 时，IDLE 状态可通过总线协议唤醒；当 TIMERCTRL.EN_TIMER_N 为 0 时，可以通过配置 IDLE 时长寄存器（TIMERH6、TIMERM8、TIMERL8）确定 IDLE 状态的计数时长，到达设置的时间后，硬件自动将 MESTAT.ME_IRQ 置 1，唤醒 MCU。其中全 MCU 模式下，唤醒 MCU 后直接进入 ACTIVE 状态；半 MCU 模式下，MCU 将 SSME 置 1 后，进入 INITIAL 状态

全自动/半自动模式：内部计时器根据寄存器设定确定 IDLE 时长，到达规定的时间，自动将 SSME 置 1，进入 INITIAL 状态。如果 TIMERCTRL.TIMER_CONF 为 1，IDLE 时长由寄存器 TIMERH6、TIMERM8、TIMERL8 决定，否则根据寄存器

TIMERCTRL.ME_PR 的值确定 IDLE 状态的计数时长

INITIAL 状态

在全自动、半自动和半 MCU 模式下，INITIAL 状态将根据寄存器的配置自动进行总线电压和温度的检测。另外为了降低功耗，在 INITIAL 状态结束前 1ms 硬件自动打开放大器，进行放大器初始化，以便进入 ACTIVE 状态能够立即使用放大器进行相应的 ADC 检测。

如果 MECONF.ADCS=3'b001 或 MECONF.ADCS=3'b011，在 INITIAL 状态硬件自动对总线电压或者温度进行 ADC 采样。

不同的工作模式，采样方式有所不同，具体如下：

全自动模式：首先进行总线电压采样，然后再进行外部温度的采样。每种都进行三次 ADC 采样，如果前两次的 ADC 值相差不超过设定的阈值，则把它们的平均值保存在专用的寄存器中，否则保存第三次的采样值到专用的寄存器中。其中温度的初始化时间由寄存器 MECONF.TIME_TEMP 的配置决定。INITIAL 状态计时结束，状态机自动跳转到 ACTIVE 状态

半 MCU/半自动模式：和全自动一样，首先进行总线电压的采样，然后再进行外部温度的采样。两种采样都只进行一次，其中总线电压的采样值保存在专用的寄存器中，温度的采样值同时保存在专用的寄存器和 ADCVAL 通用的寄存器中。INITIAL 状态计时结束，自动将 ME_IRQ 置 1，唤醒 MCU，状态机跳转到 ACTIVE 状态

ACTIVE 状态

不同的工作模式下，ACTIVE 状态进行的检测不同，具体如下：

全 MCU 模式：在该模式下，MCU 可以任意对 LN24 总线电压、温度（内部/外部）、暗电流和烟雾进行检测。ADC 检测值同时保存在 ADCVAL 通用寄存器和各自对应的专用寄存器中，且 ADC_IRQ 和各自对应的 IRQ（LN24_IRQ、EXTEMP_IRQ/INTEMP_IRQ、DARK_IRQ、SMOKE_IRQ）同时有效，在进行另一种 ADC 检测前读取 ADCVAL 通用寄存器或者当前检测类别对应的 ADC 专用寄存器时，ADC_IRQ 和对应的专用 IRQ 同时清除。

全自动模式：在 ACTIVE 状态，自动进行暗电流和烟雾检测，它们的检测值保存在各自专用的寄存器中。其中暗电流检测方法和 INITIAL 状态下总线电压、温度的检测方法相同，都采集三次 ADC 值；烟雾进行十多次检测，保存最大的 ADC 值。测试完成后，自动清除 SSME，并将 ME_IRQ 置 1，唤醒 MCU，同时将进行的各种检测对应的中断置位

半 MCU/半自动模式：这两种工作模式在 ACTIVE 状态下的检测方法和全 MCU 一样，不同的是不能进行总线电压和温度的检测

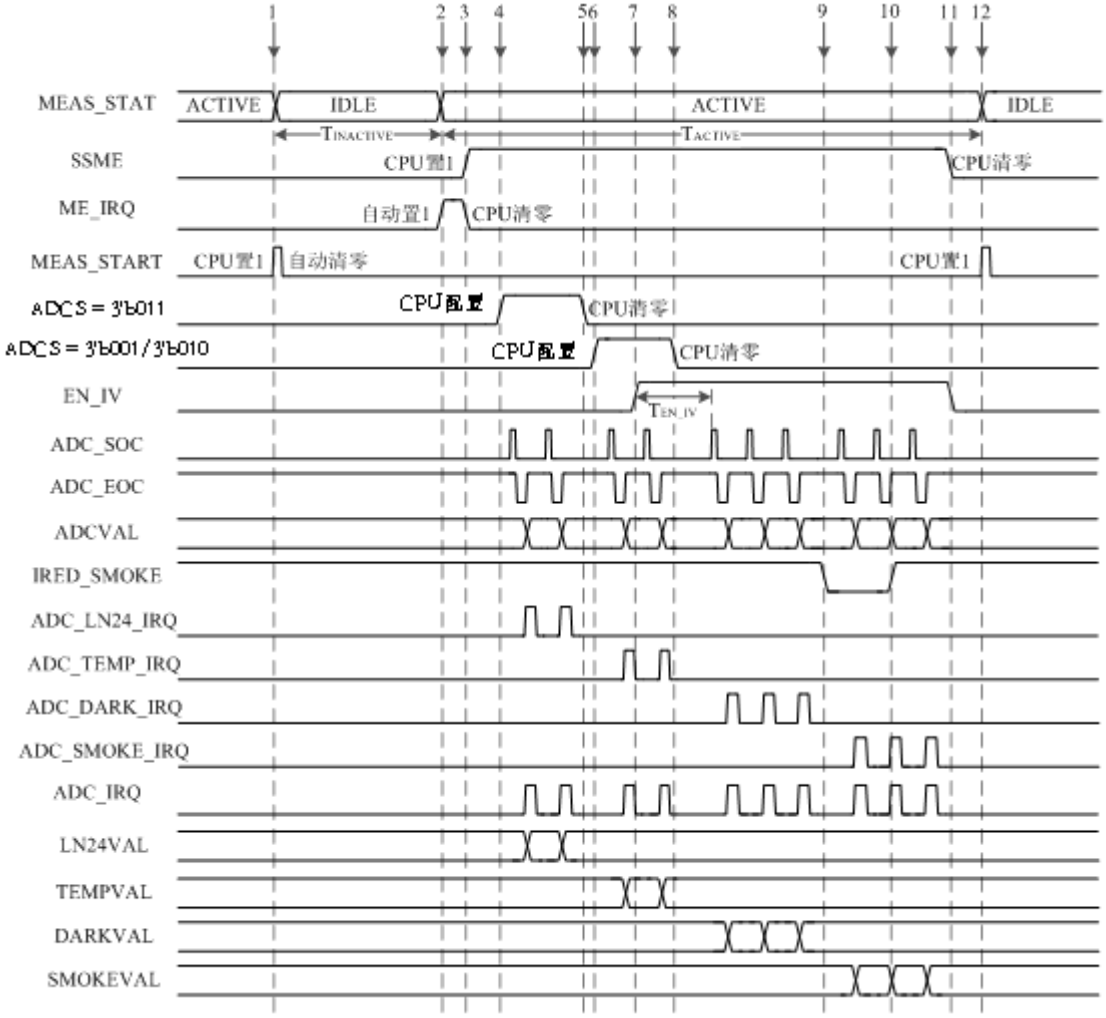
所有的检测结束后，将 SSME 清零，即结束本轮测试，MCU 进行一些后续处理，配置寄存器，通过 TIMERCTRL.MEAS_START 信号进入新的测试循环。

下面对四种工作模式下的测试时序进行详细的说明，它们都是基于内部计数器唤醒、IV 放大模式进行说明的。积分放大方式下，需要另外产生 IRED_DARK 信号。

1）全 MCU 模式

● TIMERCTRL.CYCL_EN=0

全 MCU 模式下由 MCU 控制烟雾检测时序示意图：



参数设置：MECONF.MEMODE=00b
AMPCONF.AMP_MODE=0b
TIMERCTRL.EN_TIMER_N=0b
TIMERCTRL.CYCL_EN=0

时序描述：

1）MCU 完成必要的后端处理和寄存器配置后，通过配置 TIMERCTRL.MEAS_START 为 1 进入检测循环；

2）根据 TIMERCTRL.TIMER_CONF 和时间寄存器（TIMERL8、TIMERM8、TIMEL6）的配置确定 IDLE 时间，到达设定时间，硬件自动将 ME_IRQ 置 1，唤醒 MCU，进入 ACTIVE 状态；

3）MCU 将寄存器信号 MECONF2.SSME 置 1，同时将 ME_IRQ 清零，开始烟雾检测；

4）MCU 配置 MECONF2.ADCS=3’b001，开始进行 LN24 总线电压的检测。进行总线电压检测过程中，在 ADC 转换结束时（ADC_EOC 的上升沿）会使 ADC_LN24_IRQ 和 ADC_IRQ 同时置 1，并且把得到的 ADC 值同时保存在寄存器 ADCVAL 和寄存器 LN24VAL 中。在进行检测其他种类前，读取它们中的任一个 ADC 寄存器硬件都会同时将 ADC_LN24_IRQ 和 ADC_IRQ 清零；

5）MCU 若配置 MECONF2.ADCS=3’b001，开始进行外部温度的检测。若配置 MECONF2.ADCS=3’b010，则开始进行内部温度检测，同时总线电压检测结束。内部或外部温度检测和总线电压类似，ADC 转换结束时会同时把 ADC_EXTEMP_IRQ/ADC_INTEMP_IRQ 和 ADC_IRQ 置 1，并且把得到的 ADC 值同时保存在寄存器 ADCVAL 和寄存器 TEMPVAL 中。中断的清除也和总线电压类似，进行检测其他种类前，读取 ADCVAL 和 TEMPVAL 中的任一个 ADC 寄存器硬件都会同时将 ADC_EXTEMP_IRQ/ADC_INTEMP_IRQ 和 ADC_IRQ 清零；

6）在进行暗电流和烟雾检测之前 1ms，MCU 需要通过写寄存器将 MECTRL3.EN_IV 置 1，开始放大器的初始化，该信号可以通过寄存器写 0 清除，也可以在 SSME 为 0 时由硬件自动清除；

7）MCU 配置 MECONF2.ADCS=3’b000，则进行暗电流和烟雾的检测。进行暗电流检测时，中断和 ADC 值的保存、读取以及清除都与 LN24、温度类似；

8）MCU 对 IRED3CONF.IRED_SMOKE 写 0，烟雾检测开始。烟雾检测产生的中断和 ADC 值的保存、读取以及清除与前三种的检测类似；

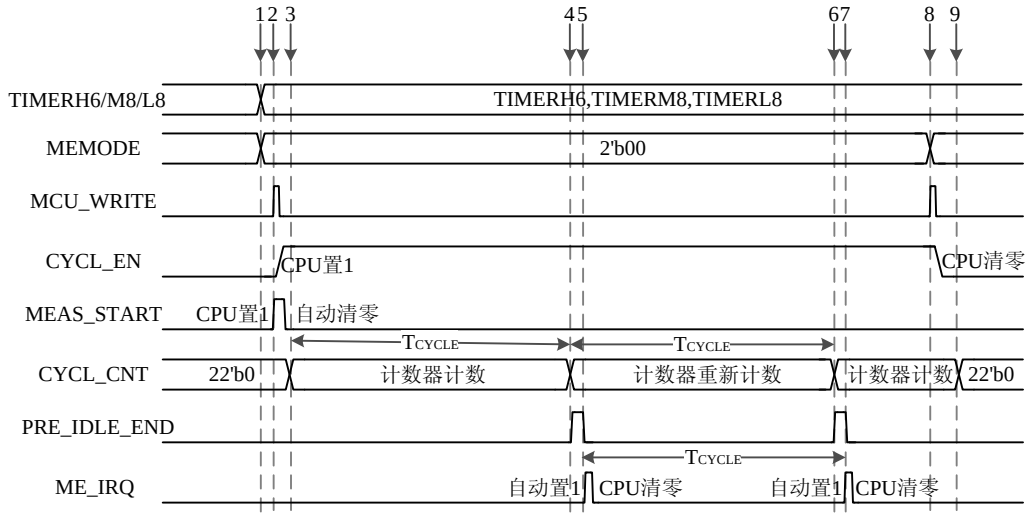
9）MCU 对 IRED3CONF.IRED_SMOKE 写 1，放大结束；

10）MCU 对 MECONF2.SSME 写 0，所有检测结束；

11）MCU 完成必要的后端处理和配置寄存器后，通过配置 TIMERCTRL.MEAS_START 为 1 进入新的检测循环。

● TIMERCTRL.CYCL_EN= 1

全 MCU 模式下循环计数检测时序示意图：



参数设置：MECONF.MEMODE=00b
TIMERCTRL.CYCL_EN=1

时序描述：

1）MCU 配置工作时钟和工作模式，并且写寄存器配置 IDLE 时长寄存器 TIMERH6、TIMERM8 和 TIMERL8，确定计数的长度；

2）MCU 配置 TIMERCTRL.CYCL_EN 为 1，使循环计数使能有效。MCU 配置 TIMERCTRL.MEAS_START 为 1，循环计数开始；

3）由于需要进行跨时钟传输（系统时钟域到工作时钟域），延迟大约 24 各系统时钟周期后，CYCL_CNT 开始从 0 计数；

4）经过 TCYCLE（该时间为：(1/工作时钟周期)*{TIMERH6，TIMERM8，TIMERL8}），计数器从 0 开始重新计数，并将 PRE_IDLE_END 置 1，打开 8M 时钟振荡器；

5）经过约 30μs，发出中断 ME_IRQ，唤醒 MCU，进行各种 ADC 检测；

6）又经过 TCYCLE，重复（4）操作；

7）重复（5）操作；

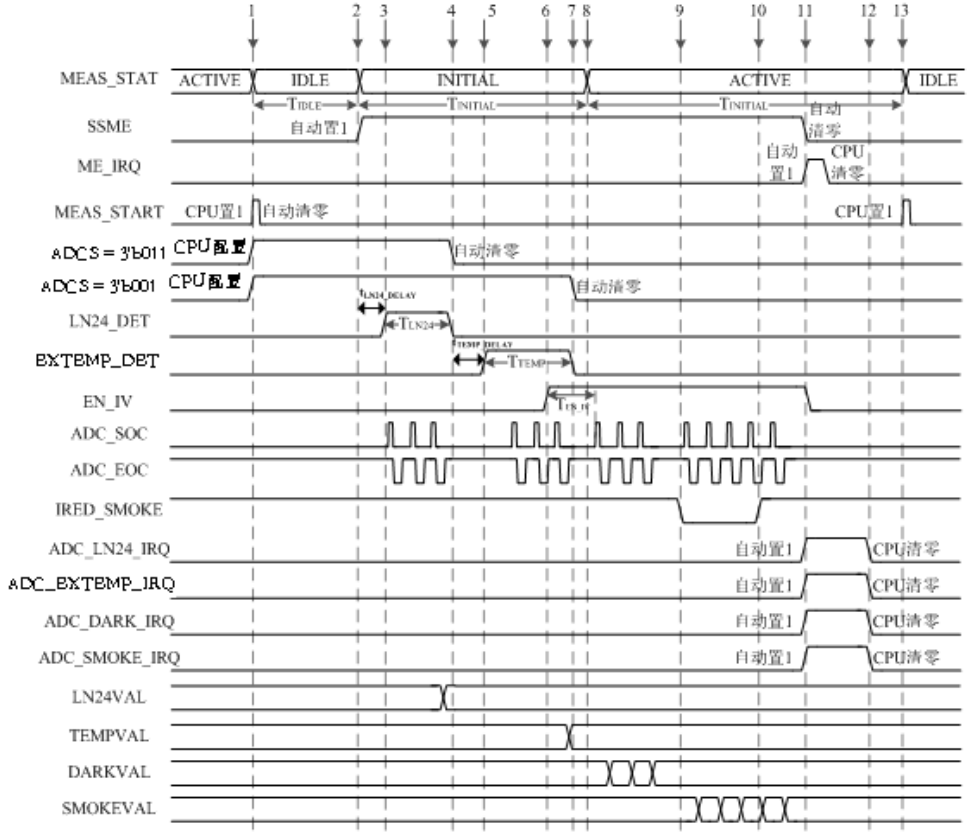
8）当 MCU 写寄存器 TIMERCTRL.CYCL_EN 为 0 清除使能或者配置为其他工作模式时，循环计数状态结束；

9）由于需要完成跨时钟操作，循环计数状态结束后大约延时 24 个系统时钟周期才会完成 CYEL_CNT 归 0 操作。但是不影响配置寄存器开始下一轮测测试，即循环测试状态结束后可任意时间写寄存器 TIMERCTRL.MEAS_CTRL 为 1，开始新的测试。

备注：循环计数时长不能小于提前打开晶振的时间与跨时钟传输的时间之和，大约 50μs

2）全自动模式

全自动模式烟雾检测时序时序示意图：



参数设置：MECONF.MEMODE=01b

AMPCONF.AMP_MODE=0b

TIMERCTRL.EN_TIMER_N=0b

系统时钟须配置为 4MHz，工作时钟须配置为 400KHz

时序描述：

1）MCU 完成必要的后端处理和寄存器配置后，通过配置 TIMERCTRL.MEAS_START 为 1 进入检测循环；

2）根据寄存器 TIMERCTRL.TIMER_CONF 和寄存器 TIMERCTRL.ME_PR 的配置，确定 IDLE 状态的时长 TIDLE，到达设定的时间，硬件自动将 SSME 置 1，进入 INITIAL 状态，开始进行 ADC 检测；

3）如果 MCU 配置 MECONF2.ADCS=3’b011，进入 INITIAL 状态和 TLN24_DELAY 时间，开始自动进行总线电压检测，自动采集三次总线电压的 ADC 值。如果前两次的 ADC 差值小于寄存器 LN24TH 设定的阈值，则保存前两次 ADC 值的平均值到寄存器 LN24VAL 中。如果大于设定的阈值，则保存第三次采集的 ADC 值到寄存器 LN24VAL 中；

4）经过 TLN24 的检测时长，自动关闭总线电压检测；

5）如果同时也配置 MECONF2.ADCS=3’b001，总线电压检测结束后，经过 TTEMP_DELAY 开始自动进行外部温度检测（不进行内部温度检测）。温度检测方

式，ADC 值的处理和总线电压类似，最后将得到的 ADC 值保存在寄存器 TEMPVAL 中。如果没有进行总线电压检测，则进入 INITIAL 状态后经过 TLN24_DELAY + TLN24 + TTEMP_DELAY 时长开始进行外部温度检测；

6）进行暗电流和烟雾检测前 TEN_IV 时间，自动将 EN_IV 置 1，进行放大器初始化；

7）根据寄存器 TIMETEMP 的配置，确定温度检测时间 TTEMP，检测时间满足后自动关闭温度检测；

8）INITIAL 状态持续时间满足 TINITIAL 后，跳转到 ACTIVE 状态，开始自动进行暗电流检测。暗电流检测三次，将得到 ADC 的最大值保存到寄存器 DARKVAL 中；

9）在进行烟雾检测之前，需 MCU 配置 MECONF2.ADCS=3’b000，打开烟雾检测 A/D 通道。暗电流检测完毕，硬件自动把 IRED_SMOKE 信号置 0。开始进行烟雾检测，烟雾检测 ADC 值的保存和暗电流相同，即将得到 ADC 的最大值保存到寄存器 SMOKEVAL 中；

10）根据 IREDCONF/IRED2CONF 设置的脉冲宽度，时间满足后，IRED_SMOKE 信号自动置 1，结束 IV 放大驱动；

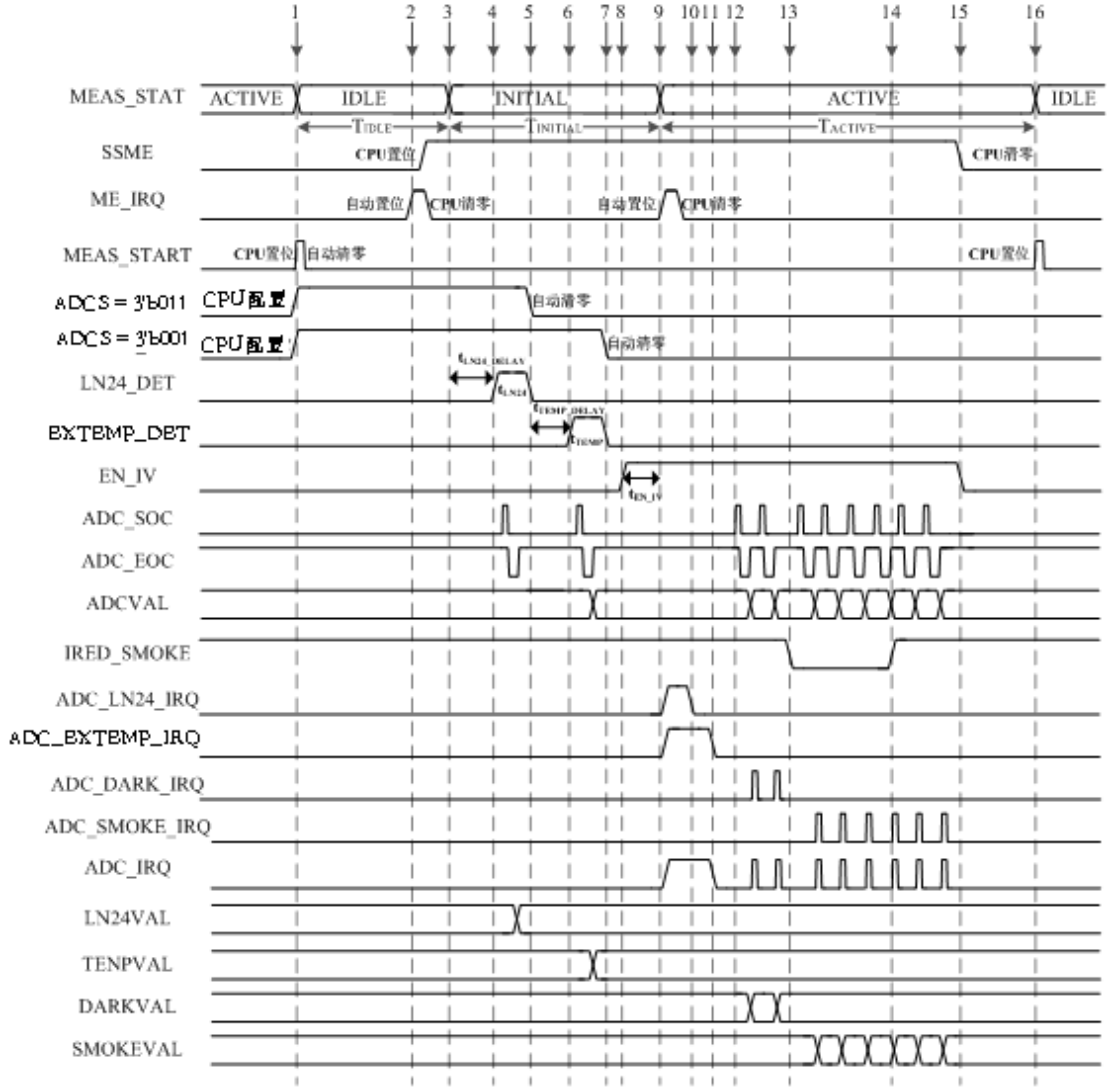
11）放大驱动结束后，IV 模式下硬件自动延迟 200μs 放电。在此过程中，采集到的烟雾 ADC 值仍采取保存最大值的方式进行更新寄存器。放电时间满足后，硬件自动将 SSME 清零，所有检测结束，ME_IRQ 自动置 1，唤醒 MCU，同时将检测到的各类中断位置 1；

12）MCU 读取各个 ADC 数据寄存器清除对应的中断；

13）MCU 完成后端处理后，通过配置 TIMERCTRL.MEAS_START 为 1，进入新的检测循环。

3）半 MCU 模式

半 MCU 模式下由 MCU 控制烟雾检测时序示意图：



参数设置：MECONF.MEMODE=10b

AMPCONF.AMP_MODE= 0b

TIMERCTRL.EN_TIMER_N= 0b

系统时钟须配置为 4MHz，工作时钟须配置为 400KHz

时序描述：

1）MCU 完成必要的后端处理和配置寄存器后，通过配置 TIMERCTRL.MEAS_START 为 1，进入检测循环；

2）根据 TIMERCTRL.TIMER_CONF 和时间寄存器（TIMERL8、TIMERM8、TIMEL6）的配置，确定 IDLE 时间。到达设定时间，硬件通过把 ME_IRQ 置 1，唤醒 MCU；

3）MCU 将写寄存器将 MESTAT.ME_IRQ 清零，将 MECONF2.SSME 置 1，测试系统进入 INITIAL 状态，进行 LN24 和温度的测量。MCU 可休眠等待总线电压和温度检测结束；

4）如果 MCU 配置 MECONF2.ADCS= 3’b011，经过 TLN24_DELAY 时长，硬件自动将 LN24_DET 置 1，开始进行总线电压检测；

5）总线电压检测时间 TLN24，采集一次 ADC 值，并保存在寄存器 LN24VAL 中；

6）如果同时也配置 MECONF2.ADCS= 3’b001，总线电压检测结束后，经过 TTEMP_DELAY 开始自动进行外部温度检测（不进行内部温度检测），硬件自动将 EXTEMP_DET 置 1，开始进行温度检测。如果总线电压没有检测，进入 INITIAL 状态后延时 TLN24_DELAY + TLN24 + TTEMP_DELAY 时长开始进行温度检测；

7）温度检测时间 TTEMP，采集一次 ADC 值，并保存在寄存器 TEMPVAL 和寄存器 ADCVAL 中，检测结束后自动将 EXTEMP_DET 清零；

8）进入 INITIAL 状态前 TEN_IV 时，自动将 EN_IV 信号置 1，开始放大器初始化；

9）整个 INITIAL 状态时长 7.8ms。时间满足后，自动将 ME_IRQ 置 1，唤醒 MCU，并将该状态中进行的 ADC 检测种类对应的 IRQ 置 1。如果进行了温度检测，硬件也同时会将 ADC_IRQ 置 1；

10）MCU 可以通过对 MESTAT.ME_IRQ 写 0 清除 ME_IRQ，也可以通过读取 ADC 数据寄存器由硬件自动清除。MCU 读寄存器 LN24VAL 的同时硬件将 ADC_LN24_IRQ 清零；

11）MCU 读寄存器 TEMPVAL 或者寄存器 ADCVAL，同时硬件自动将 ADC_EXTEMP_IRQ 和 ADC_IRQ 信号清零；

12）在进行烟雾检测之前需 MCU 配置 MECONF2.ADCS=3’b000，打开烟雾检测 A/D 通道，唤醒 MCU 后，开始进行暗电流和烟雾的检测。进行暗电流 ADC 采集时，硬件会将得到的 ADC 值，同时保存在寄存器 DARKVAL 和寄存器 ADCVAL 中，并且同时将 ADC_DARK_IRQ 和 ADC_IRQ 置 1。在此过程中，MCU 读取任意一个 DARKVAL 寄存器或者 ADCVAL 寄存器时，硬件都会将自动将 ADC_IRQ 和 ADC_DARK_IRQ 清零；

13）MCU 对 IRED3CONF.IRED_SMOKE 写 0，烟雾检测开始。烟雾检测产生的中断和 ADC 值的保存、读取以及清除与暗电流类似；

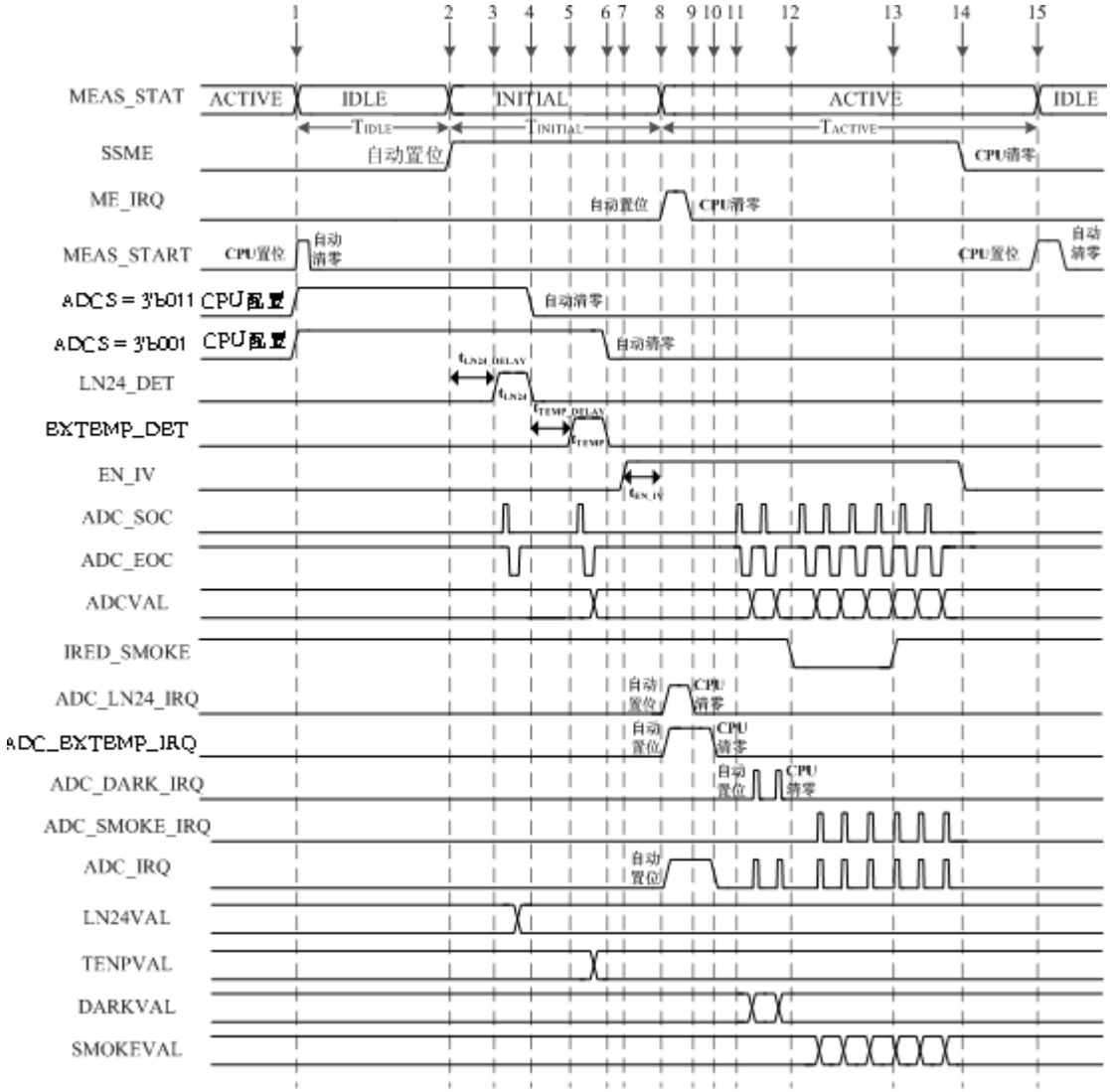
14）MCU 对 IRED3CONF.IRED_SMOKE 写 1，放大结束；

15）MCU 对 MECONF2.SSME 写 0，所有检测结束；

16）MCU 完成必要的后端处理、配置寄存器后，通过配置 TIMERCTRL.MEAS_START 为 1 进入新的检测循环。

4）半自动模式半

自动模式烟雾检测时序时序示意图：



参数设置：MECONF.MEMODE=11b

AMPCONF.AMP_MODE=0b

TIMERCTRL.EN_TIMER_N=0b

系统时钟须配置为 4MHz，工作时钟须配置为 400KHz

时序描述：

1）MCU 完成必要的后端处理和寄存器配置后，通过配置 TIMERCTRL.MEAS_START 为 1 进入检测循环；

2）根据寄存器 TIMERCTRL.TIMER_CONF 和寄存器 TIMERCTRL.ME_PR 的配置，确定 IDLE 状态的时长 TIDLE，到达设定的时间，硬件自动将 SSME 置 1，进入 INITIAL 状态，开始进行 ADC 检测；

3）如果 MCU 配置 MECONF2.ADCS=3'b011，经过 TLN24_DELAY 时长，硬件自动将 LN24_DET 置 1，开始进行总线电压检测；

4）总线电压检测时间 TLN24，采集一次 ADC 值，并保存在寄存器 LN24VAL 中；

5）如果同时也配置 MECONF2.ADCS=3’b001，总线电压检测结束后，经过 TTEMP_DELAY 开始自动进行外部温度检测（不进行内部温度检测），硬件自动将 EXTEMP_DET 置 1，开始进行温度检测。如果总线电压没有检测，进入 INITIAL 状态后延时 TLN24_DELAY + TLN24 + TTEMP_DELAY 时长开始进行温度检测；

6）温度检测时间 TTEMP，采集一次 ADC 值，并保存在寄存器 TEMPVAL 和寄存器 ADCVAL 中。检测结束后自动将 EXTEMP_DET 清零；

7）进入 INITIAL 状态前 TEN_IV 时，自动将 EN_IV 信号置 1，开始放大器初始化；

8）整个 INITIAL 状态时长 7.8ms。时间满足后，自动将 ME_IRQ 置 1，唤醒 MCU，并将该状态中进行的 ADC 检测种类对应的 IRQ 置 1。如果进行了温度检测，硬件也同时会将 ADC_IRQ 置 1；

9）MCU 可以通过对 MESTAT.ME_IRQ 写 0 清除 ME_IRQ，也可以通过读取 ADC 数据寄存器，由硬件自动清除。MCU 读寄存器 LN24VAL 的同时硬件将 ADC_LN24_IRQ 清零；

10）MCU 读寄存器 TEMPVAL 或者寄存器 ADCVAL，同时硬件自动将 ADC_EXTEMP_IRQ 和 ADC_IRQ 信号清零；

11）在进行烟雾检测之前需 MCU 配置 MECONF2.ADCS=3’b000，打开烟雾检测 A/D 通道，唤醒 MCU 后，开始进行暗电流和烟雾的检测。进行暗电流 ADC 采集时，硬件会将得到的 ADC 值，同时保存在寄存器 DARKVAL 和寄存器 ADCVAL 中，并且同时将 ADC_DARK_IRQ 和 ADC_IRQ 置 1。在此过程中，MCU 读取任意一个 DARKVAL 寄存器或者 ADCVAL 寄存器时，硬件都会将自动将 ADC_IRQ 和 ADC_DARK_IRQ 清零；

12）MCU 对 IRED3CONF.IRED_SMOKE 写 0，烟雾检测开始。烟雾检测产生的中断和 ADC 值的保存、读取以及清除与暗电流类似；

13）MCU 对 IRED3CONF.IRED_SMOKE 写 1，放大结束；

14）MCU 对 MECONF2.SSME 写 0，所有检测结束。

MCU 完成必要的后端处理、配置寄存器后，通过配置 TIMERCTRL.MEAS_START 为 1 进入新的检测循环。

5）时序参数

MEMODE	00：全 MCU	11：半 MCU	01：全自动	11：半自动
T_{IDLE}	根据 EN_TIMER_N 的配置分两种情况： EN_TIMER_N=0 TIMER_CONF 0：T_{IDLE}=1s 1：T_{IDLE}=(1/工作时钟周期)*{TIMERH6、TIMERM8、TIMERL8}	EN_TIMER_N=1 由 LN24 总线协议唤醒 MCU，进行测试	TIMER_CONF=1， T_{IDLE}=2.5μs*{TIMERH6、TIMERM8、TIMERL8} TIMER_CONF=0，由 ME_PR 决定 IDLE 时长 ME_PR： 000：1s 100：5s 001：2s 101：6s 010：3s 110：7s 011：4s 111：8s	
T_{INIT}	无	7.8ms~T_{TEMP} + 2.02ms		
T_{LN24_DELAY}	MCU 控制	1000μs		
T_{LN24}	MCU 控制	20μs	40μs	20μs
T_{TEMP_DELAY}	MCU 控制	1000μs		
T_{EN_IV}	1ms(min)	自动置 1，MCU 清零，T_{EN_IV} 由 EN_IV_MODE 决定	硬件自动产生	自动置 1，MCU 清零，T_{EN_IV} 由 EN_IV_MODE 决定
T_{TEMP}	MCU 控制	T_{TEMP} 的设定时间 000:250μs 100:4ms 001:500μs 101:8ms 010:1ms 110:16ms 011:2ms 111:32ms		
TCYCL	50μs~10.48s	无		

检测控制寄存器：

地址	寄存器名称	有效位	访问	复位值
0x0230	IRED1CONF	8	R/W	0x04
0x0231	IRED2CONF	8	R/W	0x04
0x0232	IRED3CONF	6	R/W	0x3C
0x0240	AMPCONF	8	R/W	0x20
0x0241	MECONF	8	R/W	0x00
0x0242	MECONF2	8	R/W	0x00
0x0243	MECTRL	8	R/W	0x02
0x0244	TIMERCTRL	8	R/W	0x00
0x0245	MESTAT	8	R/W	0x00

转下页

接上页

地址	寄存器名称	有效位	访问	复位值
0x0246	MEIRQEN	8	R/W	0x00
0x0247	MEMIRQ	8	R	0x00
0x0248	MESOFTRST	8	R/W	0x00
0x0249	LN24TH	8	R/W	0x7F
0x024A	TEMPTH	8	R/W	0x7F
0x024B	ADCVAL10_L	8	R	0x00
0x024C	ADCVAL10_H	8	R	0x00
0x024D	ADCVAL8	8	R	0x00
0x024E	LN24VAL_L	8	R	0x00
0x024F	LN24VAL_H	8	R	0x00
0x0250	TEMPVAL_L	8	R	0x00
0x0251	TEMPVAL_H	8	R	0x00
0x0252	DARKVAL_L	8	R	0x00
0x0253	DARKVAL_H	8	R	0x00
0x0254	SMOKEVAL_L	8	R	0x00
0x0255	SMOKEVAL_H	8	R	0x00
0x0256	SAMPLETIME	8	R/W	0x50
0x0257	TIMERL8	8	R/W	0x00
0x0258	TIMERM8	8	R/W	0x20
0x0259	TIMERH6	8	R/W	0x06

1）IRED1 的驱动电流参数的寄存器（IRED1CONF）

地址 ：0x0230

复位值 ：0x04

访问 ：R/W

Bit	Name	R/W	Description
[7：5]	IRED1_PW	R/W	IRED1 端口驱动电流脉冲宽度设置，在全 MCU 模式下无效： 111：tIRED1=400μs； 110：tIRED1=350μs； 101：tIRED1=300μs； 100：tIRED1=250μs； 011：tIRED1=200μs； 010：tIRED1=150μs； 001：tIRED1=100μs； 000：tIRED1=50μs

转下页

接上页

Bit	Name	R/W	Description
[4：1]	IRED1_DAC	R/W	IRED1 端口驱动电流脉冲大小设置： 0000：I IRED2=16mA； 0001：I IRED2=48mA； …（+32mA 每增加 1b 增加 32mA） 1111：I IRED2=496mA； 缺省值：80mA
[0]	VPOW_EN	R/W	总线供电开关功能的使能信号： 0b：使能 配合寄存器 DISCHAR_V 实现总线供电开关 1b：总线供电开关功能关闭

2）IRED2 的驱动电流参数的寄存器（IRED2CONF）

地址 ：0x0231

复位值 ：0x04

访问 ：R/W

Bit	Name	R/W	Description
[7：5]	IRED2_PW	R/W	IRED2 端口驱动电流脉冲宽度设置，在全 MCU 模式下无效： 111：tIRED2=400μs； 110：tIRED2=350μs； 101：tIRED2=300μs； 100：tIRED2=250μs； 011：tIRED2=200μs； 010：tIRED2=150μs； 001：tIRED2=100μs； 000：tIRED2=50μs
[4：0]	IRED2_DAC	R/W	IRED2 端口驱动电流脉冲大小设置： 00000：IIRED1=0mA； 00001：IIRED1=6.45mA； …（+6.45mA 每增加 1b 增加 6.45mA）； 11111：IIRED1=200.0mA 缺省值：25.8mA

3）配置 IRED1，IRED2 使能的寄存器（IRED3CONF）

地址 ：0x0232

复位值 ：0x3C

访问 ：R/W

Bit	Name	R/W	Description
[7：6]			保留

转下页

接上页

Bit	Name	R/W	Description
[5]	IRED2_DARK	R/W	IRED2 暗电流检测使能（仅对积分放大有效）： 0：暗电流检测开启； 1：暗电流检测关闭 半自动、半 MCU 工作模式的积分放大模式时，当暗电流检测驱动电流脉冲结束后，IRED2_DARK 寄存器自动置位； IV 放大模式时，不需要配置暗电流检测使能
[4]	IRED2_SMOKE	R/W	IRED2 烟雾检测电流使能： 0：烟雾检测电流使能； 1：烟雾检测电流关闭 半自动、半 MCU 工作模式时，当烟雾检测驱动电流脉冲结束后，IRED2_SMOKE 寄存器自动置位
[3]	IRED1_DARK	R/W	IRED1 暗电流检测使能（仅对积分放大有效）： 0：暗电流检测使能； 1：暗电流检测关闭 半自动、半 MCU 工作模式的积分放大模式时，当暗电流检测驱动电流脉冲结束后，IRED1_DARK 寄存器自动置位； IV 放大模式时，不需要配置暗电流检测使能
[2]	IRED1_SMOKE	R/W	IRED1 烟雾检测电流使能： 0：烟雾检测电流使能； 1：烟雾检测电流关闭 半自动、半 MCU 工作模式时，当烟雾检测驱动电流脉冲结束后，IRED1_SMOKE 寄存器自动置位
[1]	IRED2_EN	R/W	IRED2 红外发射管使能设置： 0：IRED2 红外发射管关闭； 1：IRED2 红外发射管开启
[0]	IRED1_EN	R/W	IRED1 红外发射管使能设置： 0：IRED1 红外发射管关闭； 1：IRED1 红外发射管开启

4）寄存器 AMPCONF

地址 ：0x0240

复位值 ：0x20

访问 ：R/W

Bit	Name	R/W	Description
[7]	EN_IV	R/W	放大器初始化使能信号，高有效。全 MCU 模式下，由 MCU 配置，全自动模式下自动产生，半自动/半 MCU 模式下唤醒 MCU 前 1ms 产生。 全 MCU 模式下，当 SSME 为 1 时，该信号必须在进行暗电流和烟雾检测之前 1ms（至少）置 1。可以通过写 0 清除，也可以在 SSME 为 0 时由硬件自动清除
[6]	AMP_MODE	R/W	放大器模式选择位： 0：IV 放大模式； 1：积分放大模式
[5]	ON_MUX	R/W	ADC 输入连接多路选择器： 1：正常工作时置 1； 0：需观测放大器输出波形时置 0
[4]	AMP_TSW	R/W	高电平时，放大器输出连接到 VTSEN 端口： 1：需观测放大器输出波形时置 1； 0：正常工作时需置 0
[3：0]	AMP_GAIN	R/W	光电放大器电流电压转换增益设置： 1111：保留； 1110：保留； 1101：FCONV=0.022V/nA； 1100：FCONV=0.045V/nA； 1011：FCONV=0.065V/nA； 1010：FCONV=0.085V/nA； 1001：FCONV=0.105V/nA； 1000：FCONV=0.135V/nA； 0111：FCONV=0.165V/nA； 0110：FCONV=0.21V/nA； 0101：FCONV=0.27V/nA； 0100：FCONV=0.34V/nA； 0011：FCONV=0.43V/nA； 0010：FCONV=0.53V/nA； 0001：FCONV=0.61V/nA； 0000：FCONV=0.67V/nA

5）寄存器 MECONF

地址 ：0x0241

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：6]	MEMODE	R/W	工作模式： 00：全 MCU 模式； 01：全自动模式； 10：半 MCU 模式； 11：半自动模式
[5：4]	EN_IV_MODE	R/W	EN_IV 时间选择位，选择提前打开 EN_IV 的时间： 00：1ms； 01：2ms； 10：4ms； 11：6ms
[3]	LOSC_SW	R/W	低速时钟频率选择位（设置为 1 时，内部振荡器 2 的频率由 400K 切换到 100K，所有由内部震荡器 2 产生的时钟均对应改变）： 0：选择 400K； 1：选择 100K
[2：0]	TIME_TEMP	R/W	全自动/半自动/半 MCU 模式下温度检测时 AN1SW 驱动及 ADC 打开的时间： 000：250μs； 001：500μs； 010：1ms； 011：2ms； 100：4ms； 101：8ms； 110：16ms； 111：32ms

6）寄存器 MECONF2

地址 ：0x0242

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	VREF_CS	R/W	ADC 基准选择位： 0：选择带隙基准作为产生 ADC 基准的源； 1：选择 AVDD 内部分压作为产生 ADC 基准的源 典型情况下建议设置为 0

转下页

接上页

Bit	Name	R/W	Description
[6]	AAC_EN	R/W	ADC 自动周期性采样使能： 0：由 MCU 控制每一次 ADC 采样； 1：自动进行周期性 ADC 采样，采样频率由 ADCRA 设置
[5]	SAMPLE_EN	R/W	AAC_EN=1 时的采样频率寄存器控制使能位： 0：自动采样周期由 MECONF.ADCRA 决定； 1：自动采样周期 T=0.25μs* Sample_Time
[4]	ADCRA	R/W	ADC 采样频率设置位。MCU 模式下，AAC_EN=1b 时有效： 0：15μs； 1：20μs
[3：1]	ADCS	R/W	A/D 通道选择控制位： 000：A/D 默认通道，使能烟雾信号进入 A/D，全 MCU 模式可由 MCU 控制选择。其他模式该通道一直使能打开； 001：使能外部温度传感器信号(AN1)进入 A/D，温度检测结束时，全 MCU 模式由 MCU 清零，其他模式在检测到 meas_start 信号为高后后自动使能，自动清零； 010：使能内部温度传感器信号进入 A/D，温度检测结束时，全 MCU 模式由 MCU 清零，其他模式不使能； 011：使能 LN24 总线电压信号进入 A/D。总线电压检测结束时，全 MCU 模式由 MCU 清零，其他模式在检测到 meas_start 信号为高后后自动使能，自动清零； 100：使能外部 AN0 模拟信号进入 A/D，全 MCU 模式可由 MCU 控制选择，其他模式不使能； 101：使能外部 AN2 模拟信号进入 A/D，全 MCU 模式可由 MCU 控制选择，其他模式不使能； 110：使能外部 AN3 模拟信号进入 A/D，全 MCU 模式可由 MCU 控制选择，其他模式不使能

转下页

接上页

Bit	Name	R/W	Description
[0]	SSME	R/W	烟雾检测开始标志位，高有效。开始烟雾检测时序标志位。该位写入 1，开始一个检测时序；该位写入 0，检测结束。休眠之前必须把该标志位清零： 全 MCU 模式：由 MCU 完全控制，进行 ADC 采样时该位必须为 1； 全自动模式：由硬件自动生成，检测结束自动清零； 半 MCU 模式：IDLE 状态结束时唤醒 MCU，由 MCU 设置为 1，然后 MCU 休眠；ACTIVE 状态唤醒 MCU，在 MCU 控制下完成 Dark、Smoke 检测，然后由 MCU 对该位清零； 半自动模式：ACTIVE 状态唤醒 MCU，在 MCU 控制下完成 Dark、Smoke 检测，然后由 MCU 对该位清零

7）寄存器 MECTRL

地址 ：0x0243

复位值 ：0x02

访问 ：R/W

Bit	Name	R/W	Description
[7]	EN_INTEG_N	R/W	积分模式下，积分完成标志位： 1：硬件自动置 1，表示积分完成，只有当该位为 1 时才能进行 AD 转换； 0：软件清零（除全自动模式外），清零后会积分器自动放电。
[6]	LEAK_FLAG	R	积分模式下放电标志位： 1：正在放电； 0：放电完成，只有当 0 时才能进行下一次检测
[5：3]			保留
[2]	DLOST	R/W	数据丢失标志。ADC_IRQ=1b 时，EOC 的上升沿触发该标志位置 1，即旧的 ADC 值被覆盖。MCU 读取 ADC 值，该标志位清零
[1]	EOC	R	ADC 转换结束标志位。ADC 转换完成后，该位置 1。转换开始时，该位清零。 注：判断到该 EOC 信号后需等一个 400KHz 时钟周期（2.5μs）后方可读取相应 ADC 值

转下页

接上页

Bit	Name	R/W	Description
[0]	SOC	R/W	ADC 转换开始标志位，全自动模式自动生成，其他模式如下： MECONF2.AAC_EN=0b：SSME 为 1 时对该位写 1，进行一次 ADC 采样，随后硬件自动清零； MECONF2.AAC_EN=1b：SSME 为 1 时对该位写 1，自动进行周期性 ADC 采样，采样频率由 MECONF.ADCRA 和 SAMPLE_EN 决定

8）寄存器 TIMERCTRL

地址 ：0x0244

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	TIMER_CONF	R/W	IDLE 状态时间(TIDLE)标志位： 1：根据 MCU 配置的时间 TIDLE=2.5μs*{TIMERH6，TIMERM8，TIMERL8}唤醒 MCU； 0：全/半 MCU 工作模式，TIDLE=1s；全/半自动工作模式，TIDLE 由 ME_PR 决定
[6：4]	ME_PR	R/W	当 TIMER_CONF 为 0 时，全自动/半自动模式下检测周期设置（IDLE 状态）： 000：1s； 001：2s； 010：3s； 011：4s； 100：5s； 101：6s； 110：7s； 111：8s
[3]	MEAS_START	R/W	循环检测启动信号，高有效，脉冲信号。 复位后，对该信号置位，开始进入循环检测 MCU 被 ME_IRQ 唤醒，执行完各种操作，ssme 清零。如果对该信号置 1，开始新一轮的检测
[2]	EN_TIMER_N	R/W	MCU 模式下，时间计数器使能位，当 CYCL_EN 为 0 时该位有效： 0：默认状态，半 MCU/全 MCU 工作模式时，IDLE 状态下由内部时间计数器唤醒 MCU； 1：半 MCU/全 MCU 工作模式时，IDLE 状态下由总线协议唤醒 MCU
[1]			保留

转下页

接上页

Bit	Name	R/W	Description
[0]	CYCL_EN	R/W	循环计数使能位 全 MCU 模式下，该位为 1 时，IDLE 计数器循环计数，计数时间由 TIMER_CONF 决定，计数时间到，把 ME_IRQ 置 1，然后从 0 开始重新计数

9）寄存器 MESTAT

地址 ：0x0245

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	ME_IRQ	R/W	中断请求标志位，唤醒 MCU。 MCU 可直接向该标志位写 0 或读取 ADC 值来将该标志位清零
[6]			保留
[5]	ADC_IRQ	R/W	ADC 中断请求标志位，高有效。 MCU 读取 ADC 值或对该位写 0 清除 全 MCU 模式：除了对 AN0、AN2、AN3 端口外的所有 AD 采样均反应在此位中断（AN0、AN2、AN3 端口的 AD 采样结束只能通过 EOC 信号来判断）； 半自动/半 MCU 模式：温度、暗电流和烟雾 AD 检测反应，在此位中断
[4]	ADC_SMOKE_IRQ	R/W	烟雾检测完成中断请求标志位，高有效 MCU 读取 Smoke ADC 值或对该位写 0 清除
[3]	ADC_DARK_IRQ	R/W	暗电流检测完成中断请求标志位，高有效 MCU 读取 Dark ADC 值或对该位写 0 清除
[2]	ADC_INTEMP_IRQ	R/W	内部 TEMP 总线电压检测完成中断请求标志位，高有效 MCU 读取 Temp ADC 值或对该位写 0 清除（非全 MCU 模式该标志为 0）
[1]	ADC_EXTEMP_IRQ	R/W	外部 TEMP 总线电压检测完成中断请求标志位，高有效 MCU 读取 Temp ADC 值或对该位写 0 清除
[0]	ADC_LN24_IRQ	R/W	LN24 总线电压检测完成中断请求标志位，高有效。 MCU 读取 Ln24 ADC 值或对该位写 0 清除，或者全自动模式下检测 LN24 时读取 ADCVAL 寄存器清除

10）寄存器 MEIRQEN
地址 ：0x0246
复位值 ：0x00
访问 ：R/W

Bit	Name	R/W	Description
[7：6]			保留
[5]	EN_ADC_IRQ	R/W	半自动/半 MCU/全 MCU 模式下的 ADC_IRQ 的使能信号： 1：使能； 0：除能
[4]	EN_SMOKE_IRQ	R/W	SMOKE 烟雾检测完成中断的使能信号： 1：使能； 0：除能
[3]	EN_DARK_IRQ	R/W	DARK 暗电流检测完成中断的使能信号： 1：使能； 0：除能
[2]	EN_INTEMP_IRQ	R/W	内部温度检测完成产生中断的使能信号： 1：使能； 0：除能 非全 MCU 模式该标志为 0
[1]	EN_EXTEMP_IRQ	R/W	外部温度检测完成产生中断的使能信号： 1：使能； 0：除能
[0]	EN_LN24_IRQ	R/W	LN24 总线电压检测完成产生中断的使能信号： 1：使能； 0：除能

11）寄存器 MEMIRQ
地址 ：0x0247
复位值 ：0x00
访问 ：R

Bit	Name	R/W	Description
[7：6]			保留
[5]	MIRQ_ADC	R	ADC 中断的 MASKED 信号 =EN_ADC_IRQ & ADC_IRQ
[4]	MIRQ_SMOKE	R	SMOKE 烟雾检测完成标志信号 =EN_SMOKE_IRQ & ADC_SMOKE_IRQ
[3]	MIRQ_DARK	R	DARK 暗电流检测完成标志信号 =EN_DARK _IRQ & ADC_DARK_IRQ
[2]	MIRQ_INTEMP	R	内部温度检测完成标志信号 =EN_INTEMP _IRQ & ADC_INTEMP_IRQ

转下页

接上页

Bit	Name	R/W	Description
[1]	MIRQ_EXTEMP	R	外部温度检测完成标志信号 =EN_EXTEMP _IRQ & ADC_EXTEMP_IRQ
[0]	MIRQ_LN24	R	LN24 总线电压检测完成标志信号 =EN_LN24_IRQ & ADC_LN24_IRQ

12）寄存器 MESOFTRST

地址 ：0x0248

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	ME_SOFT_RST	R/W	软复位信号，高电平有效： 由 MCU 置位自动清除，保持 20 个时钟周期完成各个状态的复位
[6：0]			保留

13）寄存器 LN24TH

地址 ：0x0249

复位值 ：0x7f

访问 ：R/W

Bit	Name	R/W	Description
[7]	LN24VAL_ERR	R/W	全自动模式下，采集三次 LN24 电压，如果前两次 ADC 差值大于设定的阈值，该位置 1，MCU 写 0 清除。
[6：0]	LN24TH	R/W	全自动模式下，LN24 电压检测阈值

14）寄存器 TEMPTH

地址 ：0x024A

复位值 ：0x7f

访问 ：R/W

Bit	Name	R/W	Description
[7]	TEMPVAL_ERR	R/W	全自动模式下，采集三次 TEMP 电压，如果前两次 ADC 差值大于设定的阈值，该位置 1，MCU 写 0 清除
[6：0]	TEMPTH	R/W	全自动模式下，TEMP 检测阈值

15）寄存器 ADCVAL10_L

地址 ：0x024B

复位值 ：0x00

访问 ：R

Bit	Name	R/W	Description
[7：2]			保留
[1：0]	ADCVAL10_L	R	全 MCU 模式存储所有 ADC_DATA[1：0]，半自动/半 MCU 模式存储 LN24 之外的所有 ADC_DATA[1：0]

16）寄存器 ADCVAL10_H

地址 ：0x024C

复位值 ：0x00

访问 ：R

Bit	Name	R/W	Description
[7：0]	ADCVAL10_H	R	全 MCU 模式存储所有 ADC_DATA[9：2]，半自动/半 MCU 模式存储 LN24 之外的所有 ADC_DATA[9：2]

17）寄存器 ADCVAL8

地址 ：0x024D

复位值 ：0x00

访问 ：R

Bit	Name	R/W	Description
[7：0]	ADCVAL8	R	全 MCU 模式存储所有 ADC_DATA[7：0]，半自动/半 MCU 模式存储 LN24 之外的所有 ADC_DATA[7：0]

18）寄存器 LN24VAL_L

地址 ：0x024E

复位值 ：0x00

访问 ：R

Bit	Name	R/W	Description
[7：2]			保留
[1：0]	LN24VAL_L	R	存储 LN24 总线电压 ADC 采样值的低 2 位，即 ADC_DATA[1：0]

19）寄存器 LN24VAL_H
地址 ：0x024F
复位值 ：0x00
访问 ：R

Bit	Name	R/W	Description
[7：0]	LN24VAL_H	R	存储 LN24 总线电压 ADC 采样值的高 8 位，即 ADC_DATA[9：2]

20）寄存器 TEMPVAL_L
地址 ：0x0250
复位值 ：0x00
访问 ：R

Bit	Name	R/W	Description
[7：2]			保留
[1：0]	TEMPVAL_L	R	温度采样时存储 ADC_DATA 的低 2 位，即 ADC_DATA[1：0]

21）寄存器 TEMPVAL_H
地址 ：0x0251
复位值 ：0x00
访问 ：R

Bit	Name	R/W	Description
[7：0]	TEMPVAL_H	R	温度采样时存储 ADC_DATA 的高 8 位，即 ADC_DATA[9：2]

22）寄存器 DARKVAL_L
地址 ：0x0252
复位值 ：0x00
访问 ：R

Bit	Name	R/W	Description
[7：2]			保留
[1：0]	DARKVAL_L	R	暗电流采样时存储 ADC_DATA 的低 2 位，即 ADC_DATA[1：0]

23）寄存器 DARKVAL_H
地址 ：0x0253
复位值 ：0x00
访问 ：R

Bit	Name	R/W	Description
[7：0]	DARKVAL_H	R	暗电流采样时存储 ADC_DATA 的高 8 位，即 ADC_DATA[9：2]

24）寄存器 SMOKEVAL_L

地址 ：0x0254

复位值 ：0x00

访问 ：R

Bit	Name	R/W	Description
[7：2]			保留
[1：0]	SMOKEVAL_L	R	烟雾采样时存储 ADC_DATA 的低 2 位，即 ADC_DATA[1：0]

25）寄存器 SMOKEVAL_H

地址 ：0x0255

复位值 ：0x00

访问 ：R

Bit	Name	R/W	Description
[7：0]	SMOKEVAL_H	R	烟雾采样时存储 ADC_DATA 的高 8 位，即 ADC_DATA[9：2]

26）寄存器 SAMPLETIME

地址 ：0x0256

复位值 ：0x50

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	SAMPLE_TIME	R/W	采样频率配置寄存器，由于 ADC 采样处理需要一定时间，使用该寄存器时配置时间不要小于 12μs，即不要小于 0x30

27）寄存器 TIMERL8

地址 ：0x0257

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	TIMER_L8	R/W	MCU 唤醒时间寄存器低 8 位

28）寄存器 TIMERM8

地址 ：0x0258

复位值 ：0x20

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	TIMER_M8	R/W	MCU 唤醒时间寄存器中 8 位

29）寄存器 TIMERH6

地址 ：0x0259

复位值 ：0x06

访问 ：R/W

Bit	Name	R/W	Description
[7：6]			保留
[5：0]	TIMER_H6	R/W	MCU 唤醒时间寄存器高 6 位

4. 15、总线供电及通讯

电路集成了两线式总线的供电和通讯单元，外部 LN24 和 LN0 可直接连接外部总线，实现供电和通讯功能。总线的正常工作电压范围为 8V~32V，最大值不能超过 39V。

4. 15. 1、总线供电

总线电压 24V 通过 5V LDO 降压到 5V 给模拟模块提供电源，AVDD 为模拟电源输出口，需外接 100μF 储能电容。5V 模拟电源再通过 1.8V LDO 给 CPU、FLASH 等数字单元供电。

4. 15. 2、接收模式

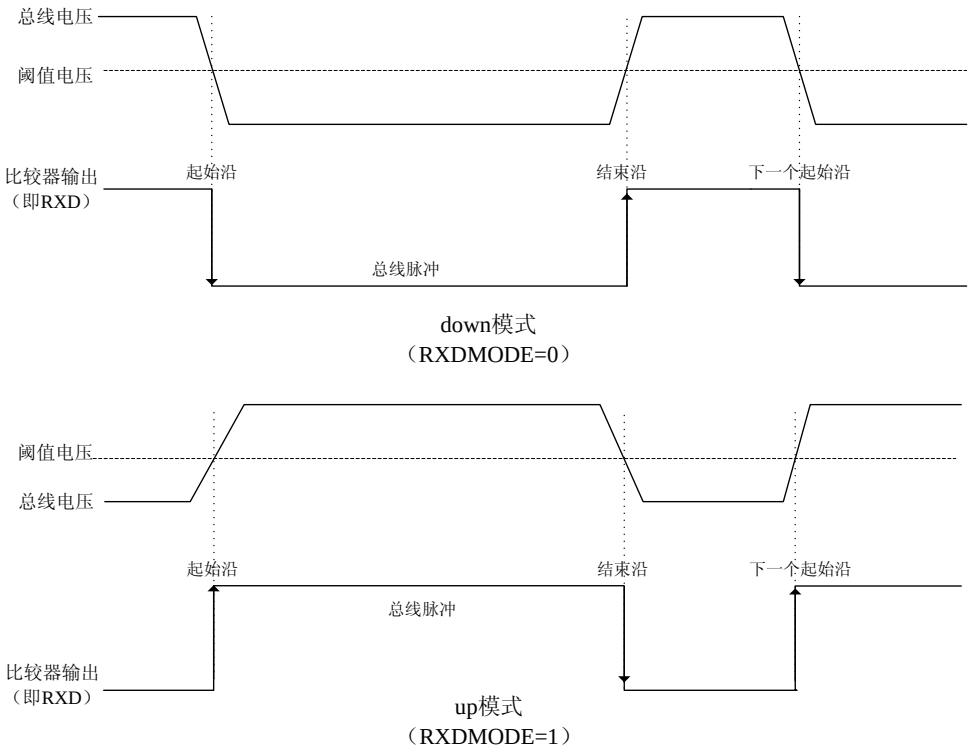
LN24 总线支持二或三种电平状态的总线协议。通过寄存器 RXVTH 对总线协议电平数量进行配置。

RXVTH=1 表示总线协议上中有三种电平状态，此时可通过 LN24CONF1.PROT_THR1 和 LN24CONF2.PROT_THR2 来分别设置两个阈值比较点。比较结果存放在 LN24STAT1.RXD1 和 LN24STAT2.RXD2 中。

RXVTH=0 表示总线协议中总线只有高低两种电平。通过 LN24CONF1.PROT_THR1 或 LN24CONF1.PROT_THR2 寄存器配置总线高低电平判断的阈值电压，该阈值电压是一个绝对电压，当设置的阈值电压比总线常态电压高，主机通过升高总线电压对从机进行发码通讯，那么就认为是 Up 接收模式，需将寄存器 LN24CONF1.RXDMODE 配置为 1。当设置的阈值电压比总线常态电压低，主机通过降低总线电压对从机进行发码通讯，那么就认为是 Down 接收模式，需将寄存器 LN24CONF1.RXDMODE 配置为 0。

当 RXMODE=0（即 Down 模式）时，总线正常工作时为高电平，发码时为低电平，总线脉冲时间是总线下降沿到上升沿的时间。当 RXMODE=1（即 Up 模式）时，总线正常工作时为低电平，发码时为高电平，总线脉冲时间是总线上升沿到下降沿的时间。

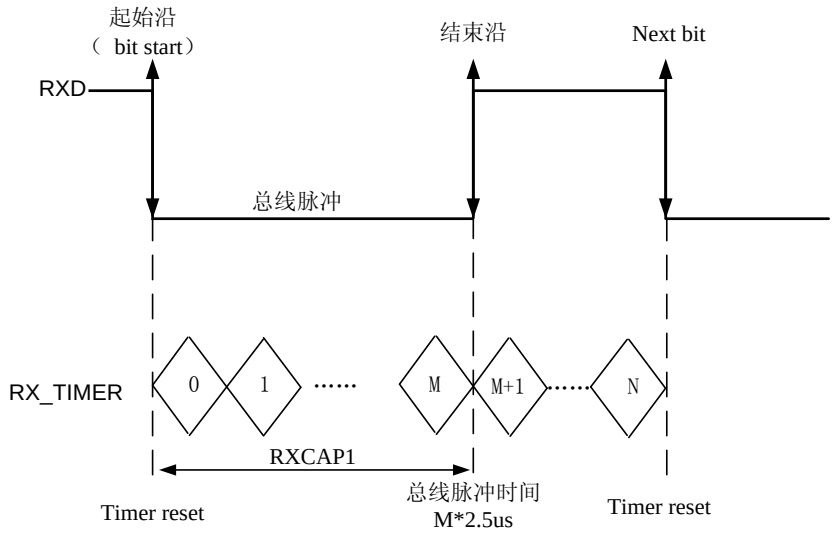
如下图所示：



因为总线协议中存在三种电平状态时对时间的定义较为复杂，所以以下涉及到的关于时间控制的寄存器仅对 RXVTH= 0 有效。

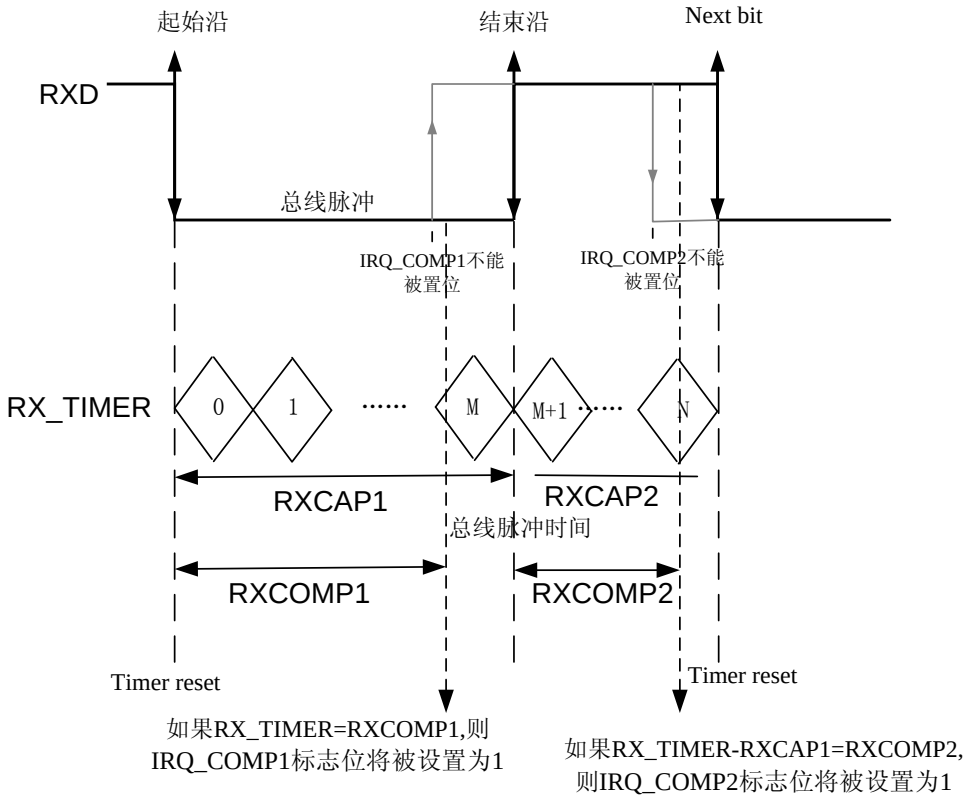
RXVTH=1 时（总线存在三种电平），只支持 RXREAD 算法。判断 RXREAD 寄存器的值，当总线起始沿开始时，计数器开始计数，计到 RX_TIMER=RXREAD 时，分别采样 RXD1 和 RXD2 的值，并存在 RXD1_RD 和 RXD2_RD 寄存器中，用户通过读寄存器 RXD1_RD 和 RXD2_RD 来判断总线电平。

总线通讯时，利用 RX_TIMER 来获取总线脉冲的时间。当总线电压起始沿触发 LN24CONF1.PROT_THR1 寄存器设置的阈值电压时，计时器 RX_TIMER 将会启动，同时中断标志位 LN24STAT.IRQ_LE 置位，当总线电压结束沿触发 LN24CONF1.PROT_THR1 寄存器设置的阈值电压时，此时计时器的计数值将会存到寄存器 RXCAP1 中，代表总线脉冲时间，同时中断标志位 LN24STAT.IRQ_CAP 被置位。



以下算法均以 Down 接收模式为例来进行说明。

（1）利用 RXCOMP 算法的接收模式时序

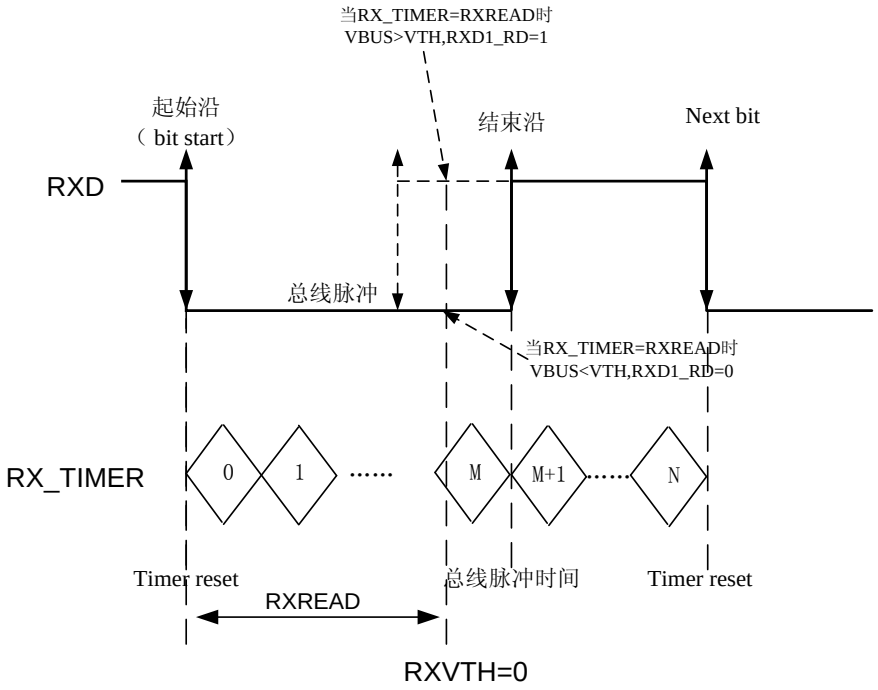


MCU 分别写两个比较值到寄存器 RXCOMP1 和 RXCOMP2，总线发码电压起始沿开始时，RX_TIMER 计数器开始计数，若 RX_TIMER= RXCOMP1 时，标志位 IRQ_COMP1 将会被设置为 1，在 RXD1 结束沿时会将 RX_TIMER 的值存到寄存器 RXCAP1，在 RXD1 下个起始沿时将 RX_TIMER 计数计到的高电平时间值存到寄存器 RXCAP2 中。RX_TIMER 继续计数，当总线电压结束沿触发 LN24CONF1.PROT_THR1 寄存器设置的阈值电压时，当 RX_TIMER-RXCAP1= RXCOMP2 时，标志位 IRQ_COMP2 将会被设置

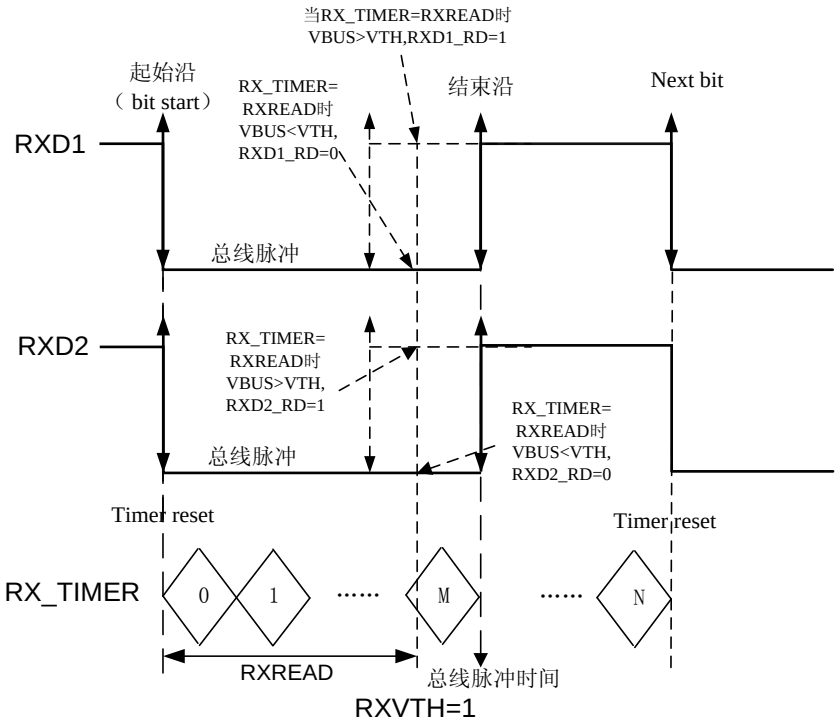
为 1，继续计数到下降沿时 RX_TIMER 复位。这个功能用于对总线脉冲时间的控制。

（2）采用 RXREAD 算法的接收模式时序

① 总线存在两种电平时：



② 总线存在三种电平时：



除了以上的总线脉冲时间控制功能之外，电路还提供读取总线高低电平的功能。MCU 写一个固定的时间值到寄存器 RXREAD。如果当 RX_TIMER=RXREAD 时，总线上的电平状态将会存储在寄存器 LN24STAT1.RXD1_RD，并且标志位 LN24STAT1.IRQ_READ 置 1。该标志位用来产生一个由 LN24CTRL.ENIRQ_READ 使能的中断/唤醒信号。

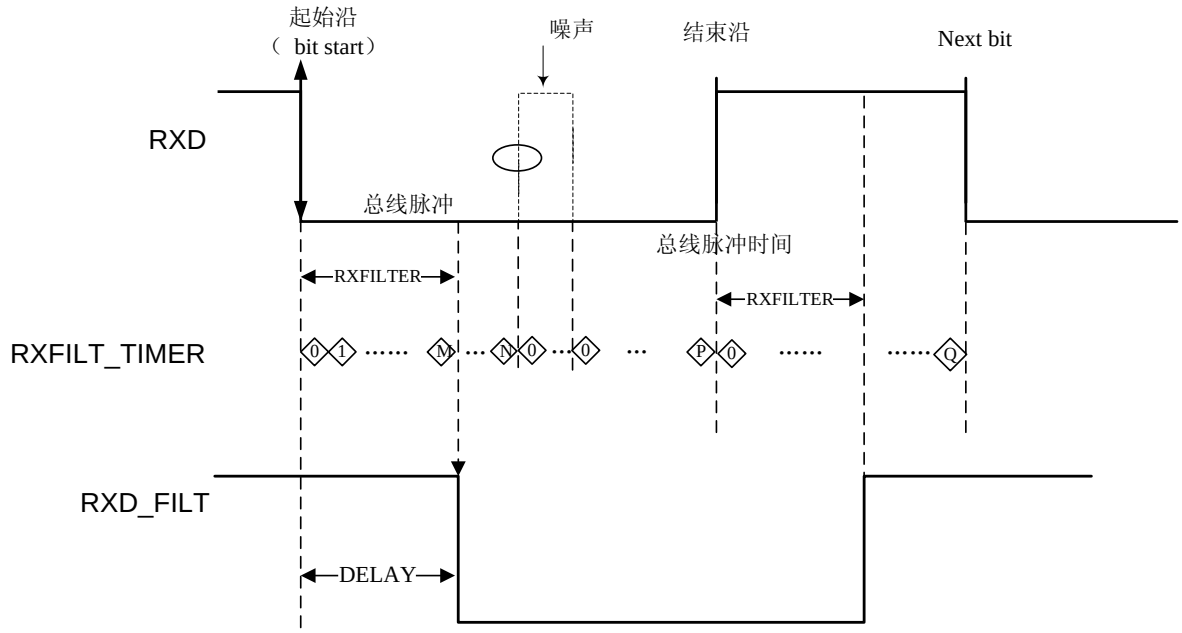
电路提供去抖功能来保证上述自动读取功能的可靠性。寄存器 RXSAMP 和 RXGATE 用来设定相关算法。

RXSAMP	RXGATE	描　述
0	0	当 RX_TIMER=RXREAD 时，对总线电压仅判断一次，结果存在 RXD1_RD[1]
0	1	当 RX_TIMER=RXREAD 时，对总线电压间隔 RXSAMP_TIME 连续采样三次，三次采样的结果经逻辑“加”后得到两位的值存到 RXD1_RD[1：0]中，IRQ_READ 在第三次采样时置位
1	0	当 RX_TIMER=RXREAD 时，对总线电压间隔 2.5μs 连续采样两次，两次采样结果通过“与”逻辑存在 RXD1_RD[1]，IRQ_READ 在第二次采样时置位
1	1	当 RX_TIMER=RXREAD 时，对总线电压间隔 2.5μs 连续检测两次，两次检测结果通过“或”逻辑存在 RXD1_RD[1]，IRQ_READ 在第二次采样时置位

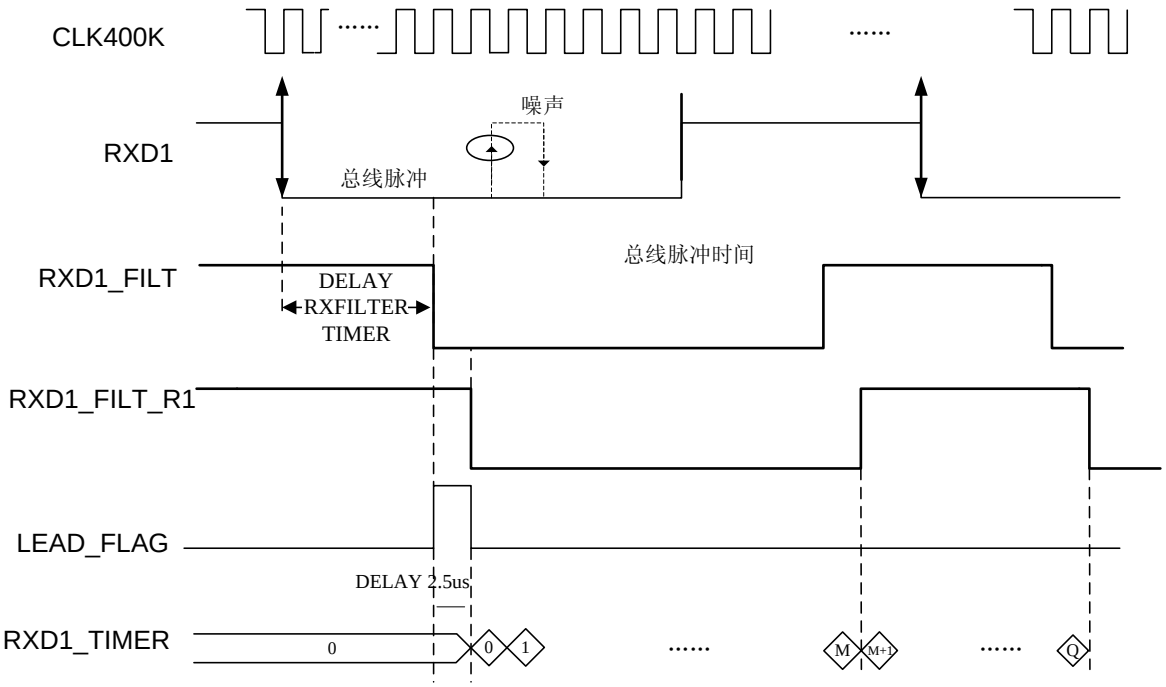
为了尽量消除总线噪声对通讯的干扰，电路提供了两种数字滤波的方法：RXFILTER 算法可以滤除在总线发码时由于电压波动等原因导致噪声信号，RXBLK 算法可以滤除回码过程中回码电流等其他原因导致的噪声信号。

（1）采用 RXFILTER 算法的接收模式时序

RXFILTER 算法数字滤波器的最大毛刺宽度可通过 8 位的寄存器 RXFILTER 设置，可滤除 2.5us~635us 以下的毛刺。

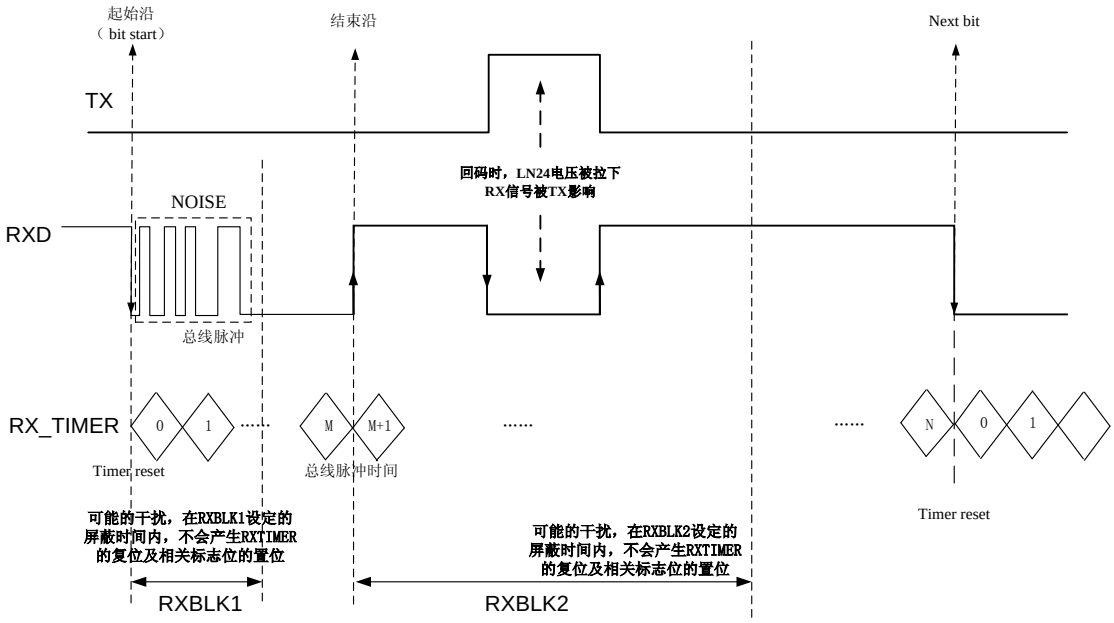


若设置寄存器 RXFILTER(RXFILTER>8’b0)进行数字滤波，滤波后得到的 RXD1_FILT 信号存在 RXFILTER 设置时间的延迟。RXD1_TIMER 从 RXD1_FILT 起始沿延迟 2.5us 时间后开始计数。



（2）采用 RXBLK 算法的接收模式时序

RXBLK 算法可配置 RXBLK1，RXBLK2 寄存器设定屏蔽时间。



MCU 写入 RXBLK1、RXBLK2 的屏蔽时间，当总线发码信号的第一个起始沿有效时，RXBLK1 设定的时间内总线上的干扰都被屏蔽；当总线发码信号的第一个结束沿有效时，RXBLK2 设定的时间内的干扰都会被屏蔽。

4. 15. 3、发送模式

在总线发送模式下，电路利用内部电流强度和持续时间可编程的电流源对总线抽取电流的形式发送信息，电流强度通过寄存器 LN24CONF3.TX_CUR 设置，最小值 0mA，最大值 240mA。

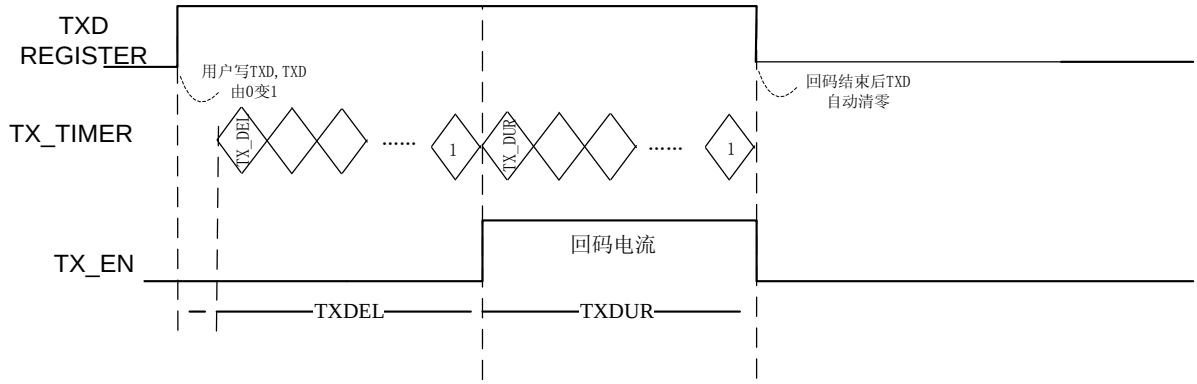
发送信息位由写入到寄存器 LN24CTRL.TXD 的数据决定。

（1）当 TXDMODE=0 时，用户对 TXD 写 1 后，延迟 Ttxd、jit+Ttxd、synch+TXDEL 时间后将使能发送模式，产生回码电流。

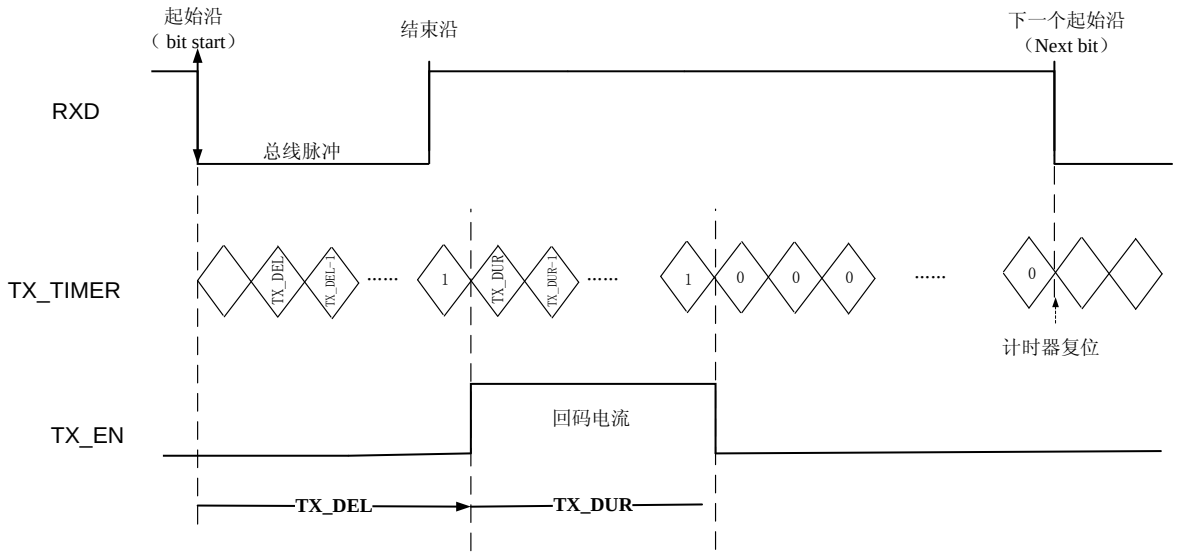
（2）当 TXDMODE=1 时，TXD 数据必须在总线起始沿开始之前写入。如果数据在总线发码某 bit 的起始沿之后写入，那么该信息将会延迟到下一个 bit 中发出。在起始沿时，TXD 的信息将被锁存并自动清零，同时 TXD 的值会被写到 TXD_LAT 中，用于检测读取正在发送的信息。因此即使在发送模式使能时，也可以实现对下一位 TXD 的操作。

在总线发送模式下，电路利用十六位计时器 TX_TIMER 来实现对回码电流的时间控制，TXDEL 用来设置从总线起始沿到产生回码电流的延迟时间，TXDUR 用来设置回码电流的持续时间。

- TXDMODE=0



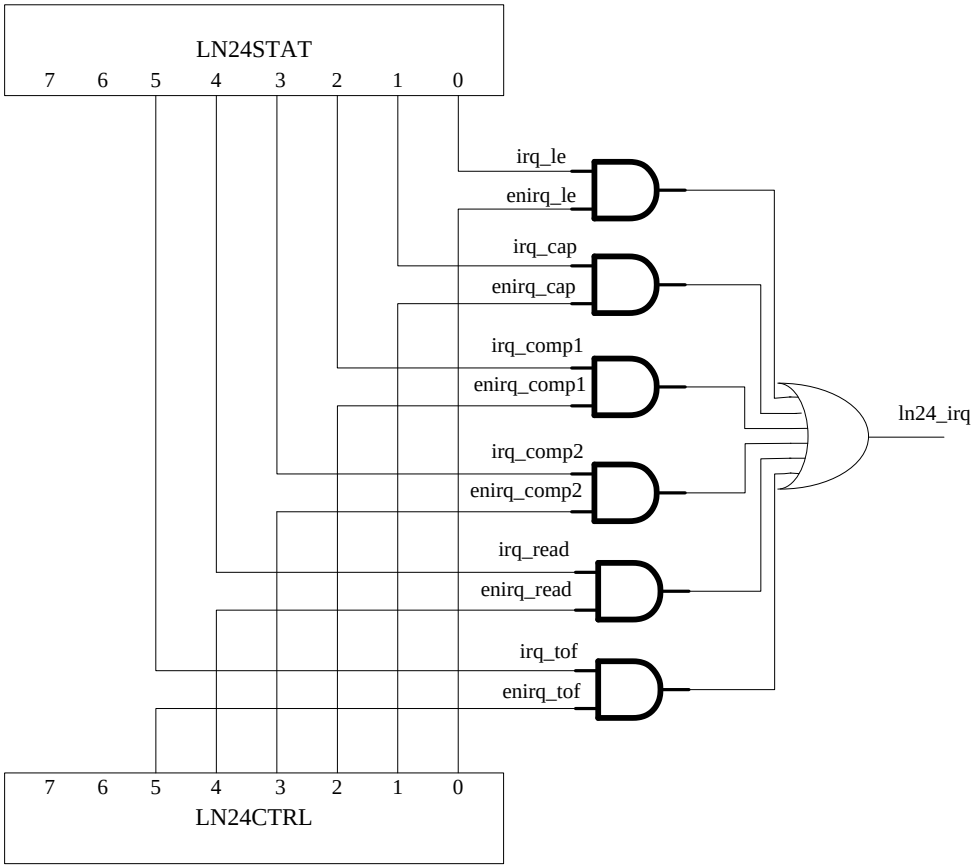
● TXDMODE=1’b1



4. 15. 4、中断

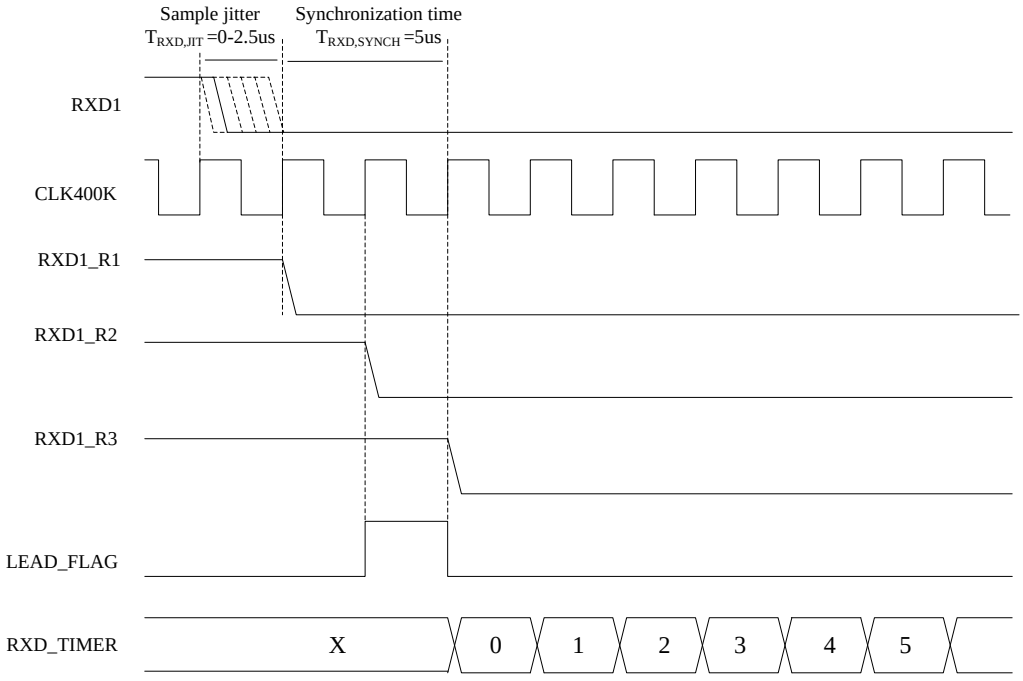
中断标志位	中断标志使能	描述
IRQ_LE	ENIRQ_LE	RXD 起始沿处产生的发码开始标志位
IRQ_CAP	ENIRQ_CAP	RXD 结束沿处，当发码的持续时间存到 rxcap 中时产生中断标志 IRQ_CAP
IRQ_COMP1	ENIRQ_COMP1	发码结束沿之前，RX_TIMER=RXCOMP1 时产生中断标志位 IRQ_COMP1
IRQ_COMP2	ENIRQ_COMP2	发码的下一个起始沿之前，RX_TIMER-RXCAP=RXCOMP2 时产生中断标志位 IRQ_COMP2
IRQ_READ	ENIRQ_READ	RX_TIMER=RXREAD 时产生中断标志位 IRQ_READ
IRQ_TOF	ENIRQ_TOF	RX_TIMER 溢出时，RX_TIMER 在 0xFFFF 处停止，产生中断标志位 IRQ_TOF

下图描述了总线控制模块的中断信号产生方法：

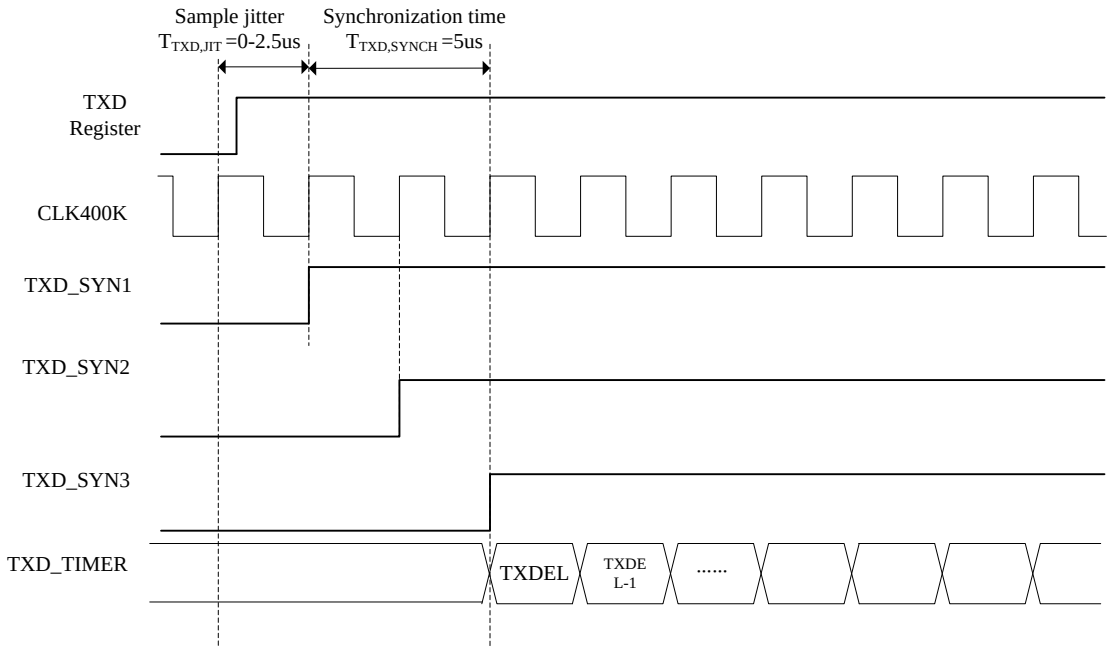


4. 15. 5、同步

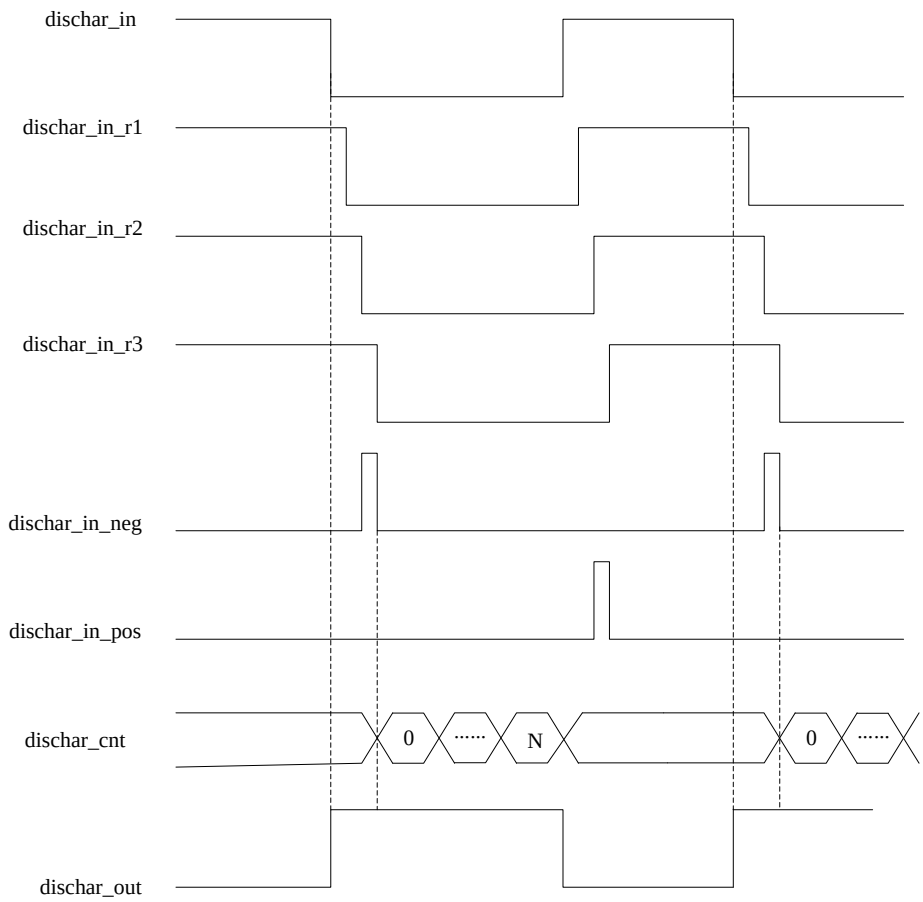
经阈值比较后得到的总线发码信号 RXD1 需同步到 CLK400K，因此 RXD1 信号存在 0~2.5μs 的采样延迟和 5μs 的同步延迟。若总线存在三种电平，RXD2 与 RXD1 一样。



同样地，当回码模式采用 txdmode= 0 时，对 TXD 寄存器写 1 后，延迟 Ttxd，jit+Ttxd，synch+TXDEL 时间发送模式使能，产生回码电流。



4. 15. 6、放电使能时序



寄存器描述

1）电源配置寄存器（POWCONF）

SFR 地址 ：0xC0

复位值 ：0x45

访问 ：R/W

Bit	Name	R/W	Description
[7]	LN24C_EN	R/W	LN24_CTRL 模块使能控制： 1：模块工作； 0：模块不工作
[6]	RX_EN	R/W	接收模式使能信号： 0：使能； 1：不使能
[5：0]			保留

2）总线接口配置寄存器 1（LN24CONF1）

SFR 地址 ：0xC1

复位值 ：0x05

访问 ：R/W

<table>
<tr><th>Bit</th><th>Name</th><th>R/W</th><th>Description</th></tr>
<tr><td>[7]</td><td>RXMODE</td><td></td><td>接收模式选择
0：下降沿为起始沿，即 Down 接收模式
1：上升沿为起始沿，即 Up 接收模式</td></tr>
<tr><td>[6]</td><td></td><td></td><td>保留</td></tr>
<tr><td>[5：0]</td><td>PROT_THR1</td><td>R/W</td><td>总线发码阈值电压 1
<table>
<tr><th>PROT_THR1</th><th>VTH1 (V)</th><th>PROT_THR1</th><th>VTH1 (V)</th></tr>
<tr><td>000010</td><td>6</td><td>100010</td><td>2</td></tr>
<tr><td>000011</td><td>7</td><td></td><td></td></tr>
<tr><td>000100</td><td>7.5</td><td>100100</td><td>2.5</td></tr>
<tr><td>000101</td><td>8</td><td></td><td></td></tr>
<tr><td>000110</td><td>9</td><td></td><td></td></tr>
<tr><td>000111</td><td>9.5</td><td></td><td></td></tr>
<tr><td>001000</td><td>10</td><td></td><td></td></tr>
<tr><td>001001</td><td>11</td><td></td><td></td></tr>
<tr><td>001010</td><td>12</td><td></td><td></td></tr>
<tr><td>001011</td><td>13</td><td></td><td></td></tr>
<tr><td>001100</td><td>14</td><td></td><td></td></tr>
<tr><td>001101</td><td>14.5</td><td></td><td></td></tr>
<tr><td>001110</td><td>15</td><td></td><td></td></tr>
<tr><td>001111</td><td>16</td><td></td><td></td></tr>
<tr><td>010000</td><td>17</td><td></td><td></td></tr>
<tr><td>010001</td><td>18</td><td>110001</td><td>3</td></tr>
<tr><td>010010</td><td>19</td><td></td><td></td></tr>
<tr><td>010011</td><td>20</td><td></td><td></td></tr>
<tr><td>010100</td><td>21</td><td>110100</td><td>3.5</td></tr>
<tr><td>010101</td><td>22</td><td></td><td></td></tr>
<tr><td>010110</td><td>23</td><td></td><td></td></tr>
<tr><td>010111</td><td>24</td><td>110111</td><td>4</td></tr>
<tr><td>011000</td><td>25</td><td></td><td></td></tr>
<tr><td>011001</td><td>26</td><td></td><td></td></tr>
<tr><td>011010</td><td>27</td><td>111010</td><td>4.5</td></tr>
<tr><td>011011</td><td>28</td><td></td><td></td></tr>
<tr><td>011100</td><td>29</td><td></td><td></td></tr>
<tr><td>011101</td><td>30</td><td>111101</td><td>5</td></tr>
<tr><td>011110</td><td>31</td><td></td><td></td></tr>
<tr><td>011111</td><td>32</td><td></td><td></td></tr>
</table></td></tr>
</table>

3）总线接口配置寄存器 2（LN24CONF2）

SFR 地址 ：0xC2

复位值 ：0x05

访问 ：R/W

Bit	Name	R/W	Description
[7]	TXDMODE	R/W	发送模式选择： 0：LN24CTRL.TXD 寄存器数据写入时刻由用户决定； 1：LN24CTRL.TXD 寄存器数据必须在总线起始沿开始之前写入
[6]	RXD_SEL	R/W	两比较器结果输出选择（仅在两总线电平模式下有效）： 0：选择输出 RXD1； 1：选择输出 RXD2

转下页

接上页

<table>
<tr><th>Bit</th><th>Name</th><th>R/W</th><th>Description</th></tr>
<tr><td>[5：0]</td><td>PROT_THR2</td><td>R/W</td><td>总线发码阈值电压 2
<table>
<tr><th>PROT_THR2</th><th>VTH2 (V)</th><th>PROT_THR2</th><th>VTH2 (V)</th></tr>
<tr><td>000010</td><td>6</td><td>100010</td><td>2</td></tr>
<tr><td>000011</td><td>7</td><td></td><td></td></tr>
<tr><td>000100</td><td>7.5</td><td>100100</td><td>2.5</td></tr>
<tr><td>000101</td><td>8</td><td></td><td></td></tr>
<tr><td>000110</td><td>9</td><td></td><td></td></tr>
<tr><td>000111</td><td>9.5</td><td></td><td></td></tr>
<tr><td>001000</td><td>10</td><td></td><td></td></tr>
<tr><td>001001</td><td>11</td><td></td><td></td></tr>
<tr><td>001010</td><td>12</td><td></td><td></td></tr>
<tr><td>001011</td><td>13</td><td></td><td></td></tr>
<tr><td>001100</td><td>14</td><td></td><td></td></tr>
<tr><td>001101</td><td>14.5</td><td></td><td></td></tr>
<tr><td>001110</td><td>15</td><td></td><td></td></tr>
<tr><td>001111</td><td>16</td><td></td><td></td></tr>
<tr><td>010000</td><td>17</td><td></td><td></td></tr>
<tr><td>010001</td><td>18</td><td>110001</td><td>3</td></tr>
<tr><td>010010</td><td>19</td><td></td><td></td></tr>
<tr><td>010011</td><td>20</td><td></td><td></td></tr>
<tr><td>010100</td><td>21</td><td>110100</td><td>3.5</td></tr>
<tr><td>010101</td><td>22</td><td></td><td></td></tr>
<tr><td>010110</td><td>23</td><td></td><td></td></tr>
<tr><td>010111</td><td>24</td><td>110111</td><td>4</td></tr>
<tr><td>011000</td><td>25</td><td></td><td></td></tr>
<tr><td>011001</td><td>26</td><td></td><td></td></tr>
<tr><td>011010</td><td>27</td><td>111010</td><td>4.5</td></tr>
<tr><td>011011</td><td>28</td><td></td><td></td></tr>
<tr><td>011100</td><td>29</td><td></td><td></td></tr>
<tr><td>011101</td><td>30</td><td>111101</td><td>5</td></tr>
<tr><td>011110</td><td>31</td><td></td><td></td></tr>
<tr><td>011111</td><td>32</td><td></td><td></td></tr>
</table></td></tr>
</table>

4）总线接口配置寄存器 3（LN24CONF3）

SFR 地址 ：0xC3

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	RXSAMP_TIME	R/W	总线接收采样时间控制： 00：三次采样时间间隔为 2.5μs； 01：三次采样时间间隔为 5μs； 10：三次采样时间间隔为 7.5μs； 11：三次采样时间间隔为 10μs
[6]	RXVTH	R/W	总线发码阈值电压控制寄存器： 0：总线协议存在两种电平状态，阈值点由 PROT_THR1 决定； 1：总线协议存在三种电平状态，阈值点由 PROT_THR1 和 PROT_THR2 决定
[4：0]	TXD_CUR	R/W	总线回码电流强度设置： 00000：IMOD=0mA； 00001：IMOD=3.7mA ……+3.7mA； 11111：IMOD=115mA

5）总线发送延迟时间低八位设置寄存器（TXDEL_L）

SFR 地址 ：0xC4

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	TXDEL_L	R/W	总线发送延迟时间设置寄存器低位[7：0]

6）总线发送延迟时间高八位设置寄存器（TXDEL_H）

SFR 地址 ：0xC5

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	TXDEL_L	R/W	总线发送延迟时间设置寄存器高位[15：8]： TXDEL=4’h 0000：0s； TXDEL=4’h0001：2.5μs ……+2.5μs； TXDEL=4’h00ff：637.5μs ……； TXDEL=4’hffff：163837.5μs

7）总线发送持续时间低八位设置寄存器（TXDUR_L）

SFR 地址 ：0xC6

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	TXDUR_L	R/W	总线发送持续时间设置寄存器低位[7：0]

8）总线发送持续时间高八位设置寄存器（TXDUR_H）

SFR 地址 ：0xC7

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	TXDUR_H	R/W	总线发送持续时间设置寄存器高位[15：8]： TXDUR=4’h0000：0s； TXDUR=4’h0001：2.5μs ……+2.5μs； TXDUR=4’h00ff：637.5μs ……； TXDUR=4’hffff：163837.5μs

9）总线接收时间比较低八位寄存器 1（RXCOMP1_L）

SFR 地址 ：0xC9

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	RXCOMP1_L	R/W	总线接收比较时间寄存器 1 低位[7：0]

10）总线接收时间比较高八位寄存器 1（RXCOMP1_H）

SFR 地址 ：0xCA

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	RXCOMP1_H	R/W	总线接收比较时间寄存器 1 高位[15：8]： RXCOMP1=4’h0000：0s； RXCOMP1=4’h0001：2.5μs ……+2.5μs； RXCOMP1=4’h00ff ：627.5μs ……； RXCOMP1=4’hffff：163837.5μs

11）总线接收时间比较低八位寄存器 2（RXCOMP2_L）

SFR 地址 ：0xCB

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	RXCOMP2_L	R/W	总线接收比较时间寄存器 2 低位[7：0]

12）总线接收时间比较高八位寄存器 2（RXCOMP2_H）

SFR 地址 ：0xCC

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	RXCOMP2_H	R/W	总线接收比较时间寄存器 2 高位[15：8]： RXCOMP2=4’h0000：0s； RXCOMP2=4’h0001：2.5μs ……+2.5μs； RXCOMP2=4’h00ff：627.5μs ……； RXCOMP2=4’hffff：163837.5μs

13）总线接收屏蔽时间低八位寄存器 1（RXBLK1_L）

SFR 地址 ：0xCD

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	RXBLK1_L	R/W	总线接收屏蔽时间寄存器低位[7：0]

14）总线接收屏蔽时间高八位寄存器 1（RXBLK1_H）

SFR 地址 ：0xCE

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	RXBLK1_H	R/W	总线接收屏蔽时间寄存器高位[15：8] RXBLK1=4’h0000：0s RXBLK1=4’h0001：2.5μs ……+2.5μs RXBLK1=4’h00ff：627.5μs …… RXBLK1=4’hffff：163837.5μs

15）总线接收屏蔽时间低八位寄存器 2（RXBLK2_L）

SFR 地址 ：0xE6

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	RXBLK2_L	R/W	总线接收屏蔽时间寄存器低位[7：0]

16）总线接收屏蔽时间高八位寄存器 2（RXBLK2_H）

SFR 地址 ：0xE7

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	RXBLK2_H	R/W	总线接收屏蔽时间寄存器高位[15：8]： RXBLK2=4’h0000：0s； RXBLK2=4’h0001：2.5μs ……+2.5μs； RXBLK2=4’h00ff：627.5μs ……； RXBLK2=4’hffff：163837.5μs

17）总线接收读取时间低八位寄存器（RXREAD_L）

SFR 地址 ：0xD1

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	RXREAD_L	R/W	总线接收读取时间寄存器低位[7：0]

18）总线接收读取时间高八位寄存器（RXREAD_H）

SFR 地址 ：0xD2

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	RXREAD_H	R/W	总线接收读取时间寄存器高位[15：8]： RXREAD=2’h00：0s； RXREAD=2’h01：2.5μs ……+2.5μs； RXREAD=4’h00ff：627.5μs ……； RXREAD=4’hffff：163837.5μs 如果 RX_TIMER 达到 RXREAD 设定的时间，则此时总线的电平状态将会被储存在 LN24STAT.RXD_RD。LN24STAT.IRQ_READ 的标志位将会设为 1

19）总线低电平时间低八位寄存器 1（RXCAP1_L）

SFR 地址 ：0xD3

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	RXCAP1_L	R	总线接收有效位时间寄存器低位[7：0]

20）总线低电平时间高八位寄存器 1（RXCAP1_H）

SFR 地址 ：0xD4

复位值 ：0x00

访问 ：R

Bit	Name	R/W	Description
[7：0]	RXCAP1_H	R	总线接收有效位时间寄存器高位[15：8]： 总线从起始沿到结束沿的时间将会被记录在 RXCAP 寄存器中，LN24STAT.IRQ_CAP 将会被置位

21）总线高电平时间低八位寄存器 2（RXCAP2_L）

SFR 地址 ：0xD5

复位值 ：0x00

访问 ：R

Bit	Name	R/W	Description
[7：0]	RXCAP2_L	R	总线高电平时间寄存器低位[7：0]

22）总线高电平时间高八位寄存器 2（RXCAP2_H）

SFR 地址 ：0xD6

复位值 ：0x00

访问 ：R

Bit	Name	R/W	Description
[7：0]	RXCAP2_H	R	总线接收有效位时间寄存器高位[15：8]： 总线从结束，沿到下一次起始沿的时间将会被记录在 RXCAP2 寄存器中

23）总线控制寄存器（LN24CTRL）

SFR 地址 ：0xE1

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	TXD	R/W	发送模式使能： 0：发送模式不使能，无电流； 1：发送模式使能，产生回码电流 注：TXDMODE=0 时，用户对 TXD 写 1 后，延迟 Ttxd、jit+ Ttxd、synch +TXDEL 时间后将产生回码电流； TXDMODE=1 时，TXD 数据必须在总线起始，沿开始之前写入。如果数据在总线发码某 bit 的起始沿之后写入，那么该信息将会延迟到下一个 bit 中发出。在起始沿时，TXD 的信息将被锁存至 TXD_LAT 并自动清零，用于检测读取正在发送的信息。因此即使在正在回码的过程中，也可以实现对下一位 TXD 的操作。
[6]	TXD_LAT	R	若 TXDMODE=1，总线起始沿时，TXD_LAT 将自动锁存住 TXD 的值
[5]	ENIRQ_TOF	R/W	使能 IRQ_TOF 产生中断/唤醒信号： 1：使能； 0：除能
[4]	ENIRQ_READ	R/W	使能 IRQ_READ 产生中断/唤醒信号： 1：使能； 0：除能
[3]	ENIRQ_COMP2	R/W	使能 IRQ_COMP2 产生中断/唤醒信号： 1：使能； 0：除能
[2]	ENIRQ_COMP1	R/W	使能 IRQ_COMP1 产生中断/唤醒信号： 1：使能； 0：除能
[1]	ENIRQ_CAP	R/W	使能 IRQ_CAP 产生中断/唤醒信号： 1：使能； 0：除能
[0]	ENIRQ_LE	R/W	使能 IRQ_LE 产生中断/唤醒信号： 1：使能； 0：除能

24）总线状态寄存器 1（LN24STAT1）

SFR 地址 ：0xE8

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：6]			保留
[5]	IRQ_TOF	R/W	RX_TIMER 溢出标志位： 1：当计时器溢出时自动置位； 0：写 0 清零
[4]	IRQ_READ	R/W	读取状态标志位： 1：当 RX_TIMER=RXREAD 时自动置位； 0：写 0 清零
[3]	IRQ_COMP2	R/W	时间比较功能 2 标志位： 1：当在下一次起始沿触发前， （RX_TIMER-RXCAP1）=RXCOMP2 时自动置位； 0：写 0 清零
[2]	IRQ_COMP1	R/W	时间比较功能 1 标志位： 1：当在结束沿触发前，RX_TIMER=RXCOMP1 时自动置位； 0：写 0 清零
[1]	IRQ_CAP	R/W	计时捕获标志位： 1：总线结束沿时自动置位； 0：写 0 清零
[0]	IRQ_LE	R/W	起始沿标志位： 1：当检测到起始沿时自动置位； 0：写 0 清零

25）总线状态寄存器 2（LN24STAT2）

SFR 地址 ：0xE2

复位值 ：0x1b

访问 ：R/W

Bit	Name	R/W	Description
[7]	RXSAMP	R/W	总线接收去抖模式选择({RXSAMP，RXGATE})： 00：1 次采样，采样写入 RXD1_RD[1]寄存器； 01：3 次采样，三次采样结果逻辑加写入。 RXD1_RD[1：0]寄存器，三次采样间隔时间由 LN24CONF3.RXSAMP_TIME[1：0]设置； 10：2 次采样，间隔 2.5μs 的两次采样结果逻辑“与”写入 RXD1_RD[1]寄存器； 11：2 次采样，间隔 2.5μs 的两次采样结果逻辑“或”写入 RXD1_RD[1]寄存器
[6]	RXGATE	R/W	
[5]	RXD1	R	总线发码时，电压比较器 1 输出状态： 0：VLN24< VTH1； 1：VLN24> VTH1 （VTH1=LN24CONF1. PROT_THR1）
[4：3]	RXD1_RD	R	在达到 RXREAD 时间时比较器 1 比较出的总线电平状态： 1 次采样时，采样结果写入 RXD1_RD[1]寄存器； 3 次采样时，三次采样结果逻辑“加”写入 RXD1_RD[1：0]寄存器； 2 次采样时间隔 2.5μs 的两次采样结果逻辑“与”(逻辑“或”)写入 RXD1_RD[1]寄存器
[2]	RXD2	R	总线发码时，电压比较器 2 输出状态： 0：VLN24< VTH2； 1：VLN24> VTH2 （VTH2=LN24CONF2. PROT_THR2）
[1：0]	RXD2_RD	R	总线存在三种电平时，在达到 RXREAD 时间时比较器 2 比较出的总线电平状态： 1 次采样时，采样结果写入 RXD2_RD[1]寄存器； 3 次采样时，三次采样结果逻辑“加”写入 RXD2_RD[1：0]寄存器； 2 次采样时间隔 2.5μs 的两次采样结果逻辑“与”(逻辑“或”)写入 RXD2_RD[1]寄存器

26）总线中断标志寄存器（LN24MIRQ）

SFR 地址 ： 0xF8

复位值 ： 0x00

访问 ： R

Bit	Name	R/W	Description
[7：6]			保留
[5]	MIRQ_TOF	R	RX_TIMER 溢出功能标识位 =IRQ_TOF and ENIRQ_TOF
[4]	MIRQ_READ	R	读取功能标识位 =IRQ_READ and ENIRQ_READ
[3]	MIRQ_COMP2	R	时间比较功能 2 标识位 =IRQ_COMP2 and ENIRQ_COMP2
[2]	MIRQ_COMP1	R	时间比较功能 1 标识位 =IRQ_COMP1 and ENIRQ_COMP1
[1]	MIRQ_CAP	R	计时捕获功能标识位 =IRQ_CAP and ENIRQ_CAP
[0]	MIRQ_LE	R	总线起始沿标识位 =IRQ_LE and ENIRQ_LE

27）数字滤波寄存器（RXFILTER）

SFR 地址 ： 0xE3

复位值 ： 0x00

访问 ： R/W

Bit	Name	R/W	Description
[7:0]	RXFILTER	R/W	总线接收数字滤波器最大毛刺时间设置寄存器 RXFILTER = 2’h00 ： 0s（即无滤波器功能） RXFILTER = 2’h01 ： 2.5us ……+2.5us RXFILTER = 2’hfe ： 635us RXFILTER 最大值为 2’hfe; 注：所有寄存器设置完成后，最后设置该寄存器。

28）放电电压电流阈值寄存器（DISCHAR_CTRL）

SFR 地址 ： 0xE4

复位值 ： 0x00

访问 ： R/W

Bit	Name	R/W	Description
[7：5]			保留

转下页

接上页

Bit	Name	R/W	Description
[4：2]	DISCHAR_V	R/W	放电电压阈值（同时也为关闭总线供电的阈值）： 000：功能关闭； 001：10V； 010：12V； 011：14V； 100：16V； 101：18V； 110：20V； 111：22V 备注：当使能总线供电开关功能时，总线电压低于此设置阈值时，将停止向总线上抽取供电，此时仅由 AVDD 处外挂电容为电路供电
[1：0]	DISCHAR_I	R/W	放电电流大小： 00：0μA； 01：60μA； 10：120μA； 11：180μA

29）放电使能控制寄存器（DISCHAR_THR）

SFR 地址 ：0xE5

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：4]			保留
[3：0]	DISCHAR_THR	R/W	放电使能时间控制： 0000：脉宽为 0； 0001：脉宽为 2.5+（5~7.5）μs； 0010：脉宽为 5+（5~7.5）μs +2.5μs every step ……； 1111：脉宽 37.5+（5~7.5）μs

4. 16、PWM

PWM 控制器会产生一连串脉冲，脉冲的周期和宽度可通过 CPU 进行配置。占空比被定义为脉冲宽度与周期的比值。PWM 有着广泛的应用，大多数情况下用于控制模拟电路。因为数字信号连续变化的速率相对较快(当然取决于信号周期)，因此最终会形成一个用来控制模拟设备的平均电压值。当 PWM 脉冲流应用于马达时，马达的转速就能正比于占空比(从 0%到 100%)。如果占空比增加，马达转速就会提高，反之，如果占空

比减小，马达的转速随之也会降低。端口控制寄存器：

地址	寄存器名称	有效位	访问	复位值
0x0290	PWM_PERIOD0L	8	R/W	0x00
0x0291	PWM_PERIOD0H	8	R/W	0x00
0x0292	PWM_PULSE0L	8	R/W	0x00
0x0293	PWM_PULSE0H	8	R/W	0x00
0x0294	PWM_PERIOD1	8	R/W	0x00
0x0295	PWM_PULSE1	8	R/W	0x00
0x0296	PWM_PERIOD2	8	R/W	0x00
0x0297	PWM_PULSE2	8	R/W	0x00
0x0298	PWM_PERIOD3L	8	R/W	0x00
0x0299	PWM_PERIOD3H	8	R/W	0x00
0x029A	PWM_PULSE3L	8	R/W	0x00
0x029B	PWM_PULSE3H	8	R/W	0x00
0x029C	PWM_E	8	R/W	0x00

寄存器描述

1）PWM_PERIOD0L 寄存器

地址 ：0x0290

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	PERIOD0L	R/W	PERIOD0L：PWM0 周期低位设置寄存器

2）PWM_PERIOD0H 寄存器

地址 ：0x0291

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	PERIOD0H	R/W	PERIOD0H：PWM0 周期高位设置寄存器 {PERIOD0H，PERIOD0L}：PWM0 周期设置寄存器 0000h：PWM0 最小周期为 0 …… ffffh：PWM0 最大周期占 65535 个时钟周期

3）PWM_PULSE0L 寄存器

地址 ：0x0292

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	PULSE0L	R/W	PULSE0L：PWM0 脉宽低位设置寄存器

4）PWM_PULSE0H 寄存器

地址 ：0x0293

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	PULSE0H	R/W	PULSE0H：PWM0 脉宽高位设置寄存器 {PULSE0H，P PULSE0L}：PWM0 周期设置寄存器 00h：PWM0 最小脉宽为 0 …… ffh：PWM0 最大脉宽占 65535 个时钟周期（2.5μs）

5）PWM_PERIOD1 寄存器

地址 ：0x0294

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	PERIOD1	R/W	PERIOD1：PWM1 周期设置寄存器 00h：PWM1 最小周期为 0 …… ffh：PWM1 最大周期占 255 个时钟周期（2.5μs）

6）PWM_PULSE1 寄存器

地址 ：0x0295

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	PULSE1	R/W	PULSE1：PWM1 脉宽设置寄存器 00h：PWM1 最小脉宽为 0 …… ffh：最大脉宽占 255 个时钟周期（2.5μs）

7）PWM_PERIOD2 寄存器

地址 ：0x0296

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	PERIOD2	R/W	PERIOD2：PWM2 周期设置寄存器 00h：PWM2 最小周期为 0 …… ffh：PWM2 最大周期占 255 个时钟周期（2.5μs）

8）PWM_PULSE2 寄存器

地址 ：0x0297

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	PULSE2	R/W	PULSE2：PWM2 脉宽设置寄存器 00h：PWM2 最小脉宽为 0 …… ffh：PWM2 最大脉宽占 255 个时钟周期（2.5μs）

9）PWM_PERIOD3L 寄存器

地址 ：0x0298

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	PERIOD3L	R/W	PERIOD3L：PWM3 周期低位设置寄存器

10）PWM_PERIOD3H 寄存器

地址 ：0x0299

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	PERIOD3H	R/W	PERIOD3H：PWM3 周期高位设置寄存器 {PERIOD3H，PERIOD3L}：PWM3 周期设置寄存器 0000h：PWM3 最小周期为 0 …… ffffh：PWM3 最大周期占 65535 个时钟周期（2.5μs）

11）PWM_PULSE3L 寄存器

地址 ：0x029A

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	PULSE3L	R/W	PULSE3L：PWM3 脉宽低位设置寄存器

12）PWM_PULSE3H 寄存器

地址 ：0x029B

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7：0]	PULSE3H	R/W	PULSE3H：PWM3 脉宽高位设置寄存器 {PULSE3H，P PULSE3L}：PWM3 周期设置寄存器 00h：PWM3 最小脉宽为 0 …… ffh：PWM3 最大脉宽占 65535 个时钟周期（2.5μs）

13）PWM_E 寄存器

地址 ：0x029C

复位值 ：0x00

访问 ：R/W

Bit	Name	R/W	Description
[7]	PW3_OE	R/W	PW3_OE：PA6 PWM3 输出模式： 0：PWM3 循环输出； 1：PWM3 单次输出
[6]	PW2_OE	R/W	PW2_OE：PA2 PWM2 输出模式： 0：PWM2 循环输出； 1：PWM2 单次输出
[5]	PW1_OE	R/W	PW1_OE：PA1 PWM1 输出模式： 0：PWM1 循环输出； 1：PWM1 单次输出
[4]	PW0_OE	R/W	PW0_OE：PA0 PWM0 输出模式： 0：PWM0 循环输出； 1：PWM0 单次输出
[3]	PW3_E	R/W	PW3_E：PA6 端口 PWM3 输出使能： 0：禁止 PWM3 输出。单次触发模式，该位自动清零（也可由 MCU 关闭）。默认为循环模式，要 MCU 将其清零； 1：允许 PWM3 输出，MCU 写 1 开启 PWM3 输出
[2]	PW2_E	R/W	PW2_E：PA2 PWM2 输出使能： 0：禁止 PWM2 输出；单次触发模式，该位自动清零（也可由 MCU 关闭）。默认为循环模式，要 MCU 将其清零； 1：允许 PWM2 输出，MCU 写 1 开启 PWM2 输出
[1]	PW1_E	R/W	PW1_E：PA1 PWM1 输出使能： 0：禁止 PWM1 输出。单次触发模式，该位自动清零（也可由 MCU 关闭）。默认为循环模式，要 MCU 将其清零； 1：允许 PWM1 输出，MCU 写 1 开启 PWM1 输出
[0]	PW0_E	R/W	PW0_E：PA0 PWM0 输出使能： 0：禁止 PWM0 输出。单次触发模式，该位自动清零（也可由 MCU 关闭）。默认为循环模式，要 MCU 将其清零； 1：允许 PWM0 输出，MCU 写 1 开启 PWM0 输出

MCU 设置\读取 PWM 控制模块寄存器工作流程：

1）PWM 使能寄存器（地址 0x29c）默认为 8'h00，处于关闭。MCU 为相应位写“1”开启相应 PWM；

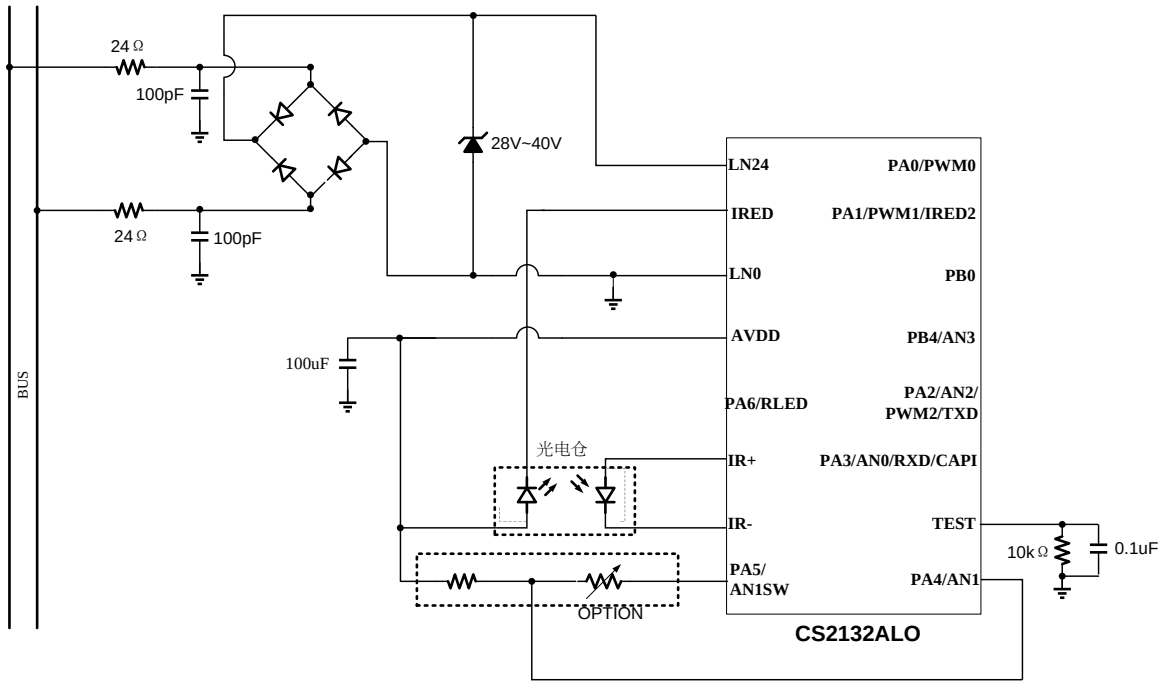
2）MCU 根据需要配置 PWM 周期和脉宽控制寄存器，并将相应管脚配置为 PWM 输出；

3）内部计数器根据设定周期计数并与脉宽寄存器比较得到 PWM 周期波形；

4）MCU 可读当前 PWM 的周期和脉宽以及控制信号信息；

5）设置好某路 PWM 的周期、脉宽及触发模式，然后 MCU 开启对应使能位，即输出想要的 PWM。默认为循环模式，由 MCU 关闭使能结束输出。如在单次触发使能位写“1”，则为单次触发模式，完成一个周期 PWM 输出同时清除对应使能位；为单次触发使能位写“0”，则进入循环模式。

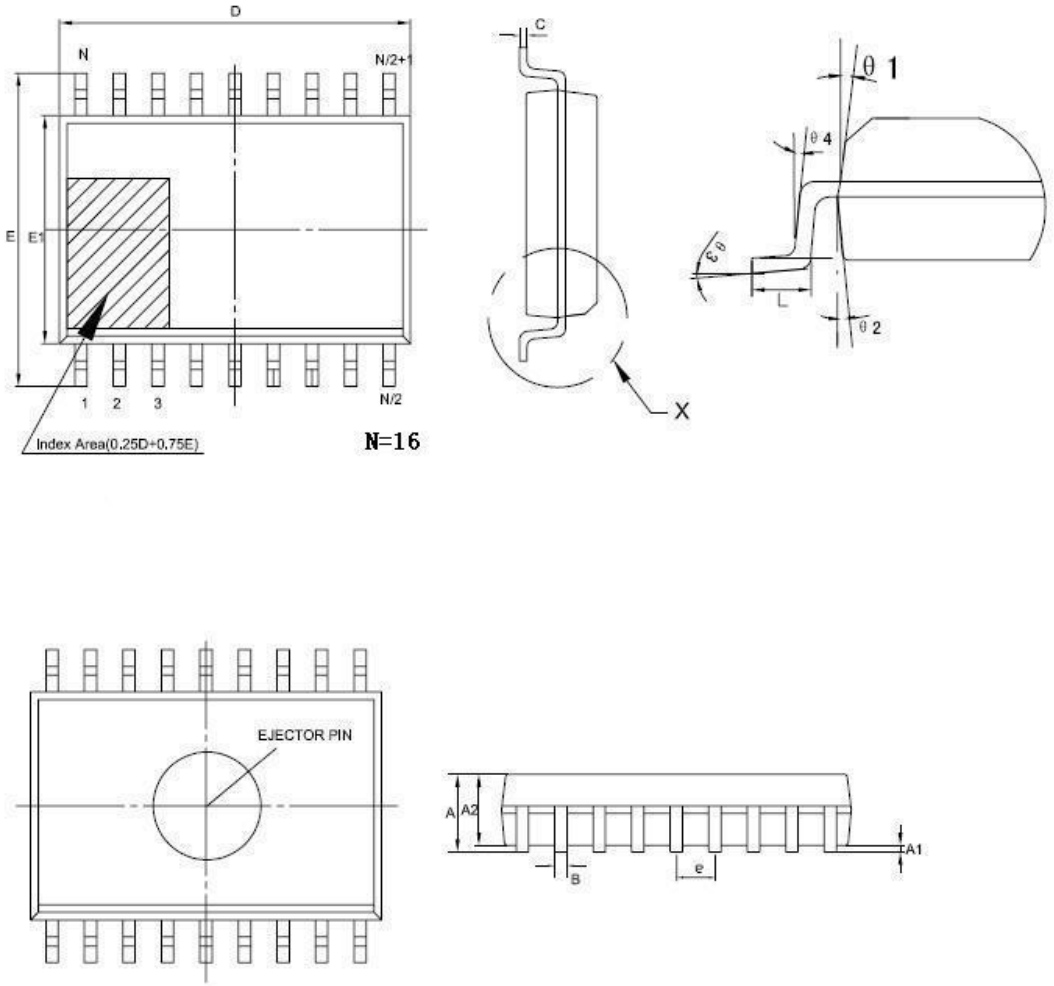
5、典型应用线路图



6、封装尺寸与外形图（单位：mm）

6. 1、SOP16 封装

6. 1. 1、封装外形图

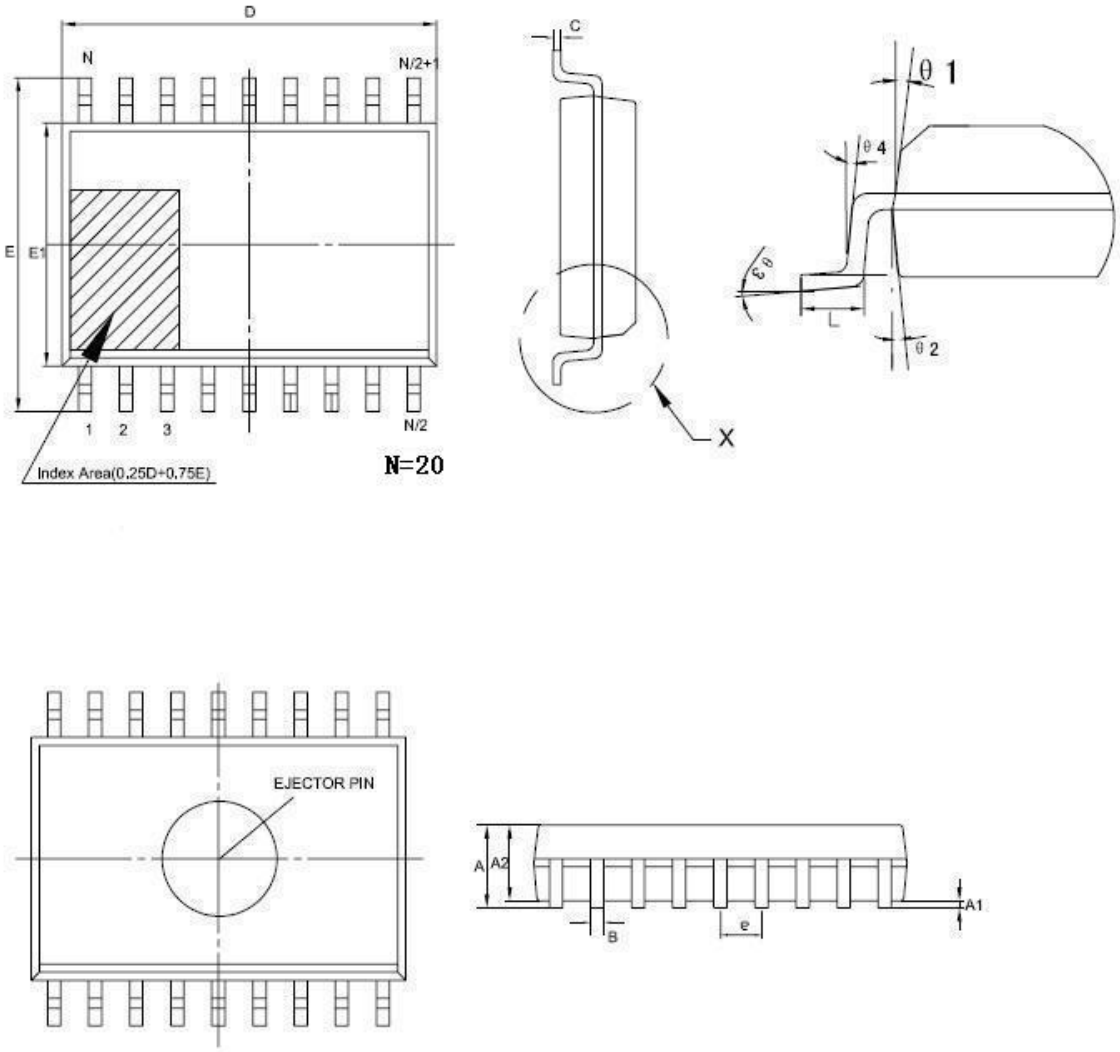


6. 1. 2、封装尺寸

Symbol	Min	Max	Symbol	Min	Max
A	1.40	1.73	e	1.27TYP	
A1	0.05	0.18	B	0.36	0.46
A2	1.35	1.55	C	0.20TYP	
E	5.84	6.24	θ1	8° TYP	
E1	3.84	4.04	θ2	8° TYP	
D	9.90	10.10	θ3	0°	8°
L	0.40	0.70	θ4	15° TYP	

6. 2、SOP20 封装

6. 2. 1、封装外形图

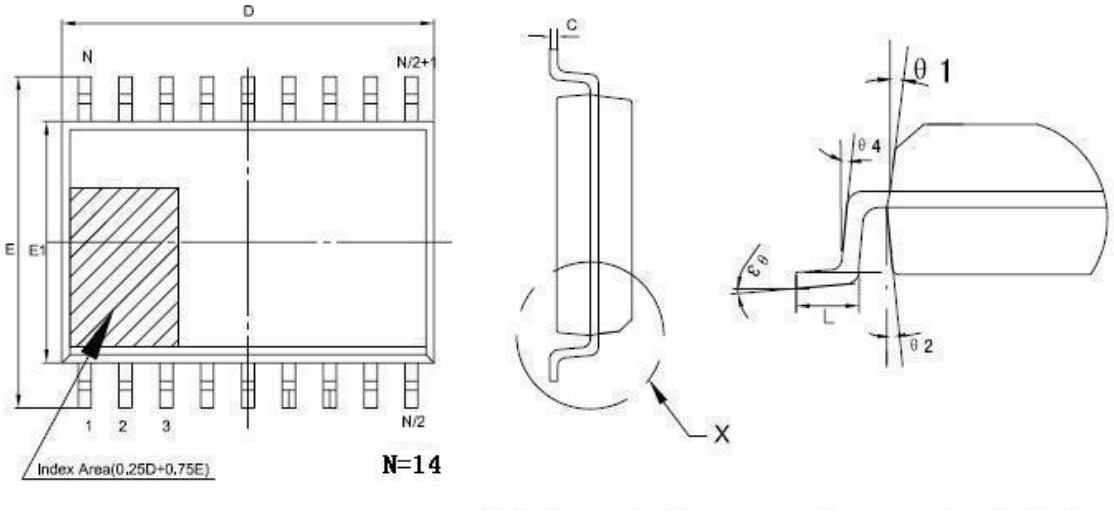


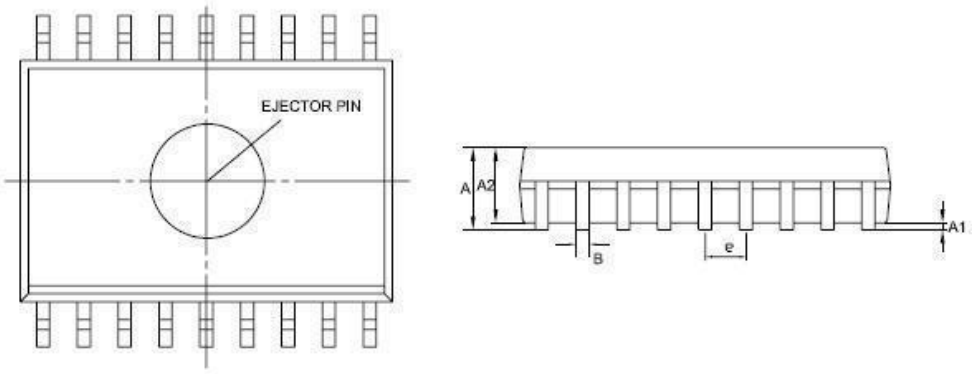
6. 2. 2、封装尺寸

Symbol	Min	Max	Symbol	Min	Max
A	1.85	2.20	e	1.27TYP	
A1	0.05	0.20	B	0.40	0.48
A2	1.80	2.00	C	0.20TYP	
E	7.60	8.20	θ1	15° TYP	
E1	5.20	5.40	θ2	8° TYP	
D	12.35	12.55	θ3	0°	10°
L	0.30	0.70	θ4	5° TYP	

6. 3、SOP14 封装

6. 3. 1、封装外形图





6. 3. 2、封装尺寸

Symbol	Min	Max	Symbol	Min	Max
A	1.45	1.80	e	1.27TYP	
A1	0.10	0.25	B	0.33	0.51
A2	1.35	1.55	C	0.20TYP	
E	5.80	6.20	θ1	8° TYP	
E1	3.80	4.00	θ2	8° TYP	
D	8.55	8.75	θ3	0°	8°
L	0.45	0.65	θ4	4° TYP	

产品中有毒有害物质或元素的名称及含量

部件名称	有毒有害物质或元素					
	铅（Pb）	汞（Hg）	镉（Cd）	六价铬（Cr^{+6}）	多溴联苯（PBB）	多溴联苯醚（PBDE）
引线框	○	○	○	○	○	○
塑封树脂	○	○	○	○	○	○
芯片	○	○	○	○	○	○
内引线	○	○	○	○	○	○
装片胶	○	○	○	○	○	○
说明	○：表示该有毒有害物质的含量在 SJ/T11363-2006 标准的限量要求以下。 ×：表示该有毒有害物质的含量超出 SJ/T11363-2006 标准的限量要求。					

无锡华润矽科微电子有限公司

WUXI CHINA RESOURCES SEMICO CO.，LTD.

地址：中国江苏省无锡市菱湖大道 180-22

邮编：214135

电话：0510-85810118

传真：（销售传真）0510-85874503

网址：http://www.semico.com.cn

公司主要销售联络点：

市场营销部：江苏省无锡市菱湖大道 180-22

邮编：214135 电话：0510-85810118-4445 / 2320 传真：0510-85874503

深圳分公司：深圳福田区梅林工业区梅秀路 1-1 号华强云产业园厂房 3 栋 4 楼 409 室

邮编：518000 电话：0755-83572766 传真：0755-83547381，0755-83547451

应用技术服务：

市场应用部：江苏省无锡市菱湖大道 180-22

邮编：214135 电话：0510-81805245，81805507 传真：0510-81805517

深圳分公司：深圳福田区梅林工业区梅秀路 1-1 号华强云产业园厂房 3 栋 4 楼 409 室

邮编：518000 电话：0755-83947357，83547123 传真：0755-83572811

注意：

建议您在使用矽科产品之前仔细阅读本资料。

希望您经常和矽科有关部门进行联系，索取最新资料，因为矽科产品在不断更新和提高。

本资料中的信息如有变化，恕不另行通知。

本资料仅供参考，矽科不承担任何由此而引起的损失。

矽科不承担任何在使用过程中引起的侵犯第三方专利或其它权利的责任。