

WL8032 用户手册

8 位单片机

2022/11/08

目录

1	产品简介	4
1.1	产品特性	4
1.2	系统框图	5
1.3	引脚排列	5
1.4	引脚说明	7
2	中央处理器	8
2.1	指令集	8
2.2	程序存储器 (OTP)	10
2.3	数据存储器	11
2.4	堆栈	11
2.5	烧录配置选项	12
2.6	控制寄存器	13
3	系统时钟	16
3.1	外接晶体振荡器	16
3.2	内置高频 RC 振荡器	17
3.3	内置低频 RC 振荡器	17
3.4	工作模式	18
3.5	低功耗模式	20
4	复位	21
4.1	复位条件	21
4.2	上电复位	21
4.3	低电压复位	21
5	IO 口	23
5.1	IO 工作模式	23
5.2	上拉电阻控制	25
5.3	端口模式控制	25
6	定时器	27
6.1	定时器 T0	28
6.2	定时器 T1	31
7	键盘扫描模块	34
7.1	键盘扫描介绍	34
7.2	键盘扫描操作步骤	34
7.3	键盘扫描相关寄存器	35
8	低电压检测 (LVD)	36
9	液晶显示控制及驱动	36
9.1	LCD 相关寄存器	38
9.2	LCD 数据寄存器	40
10	中断	41
10.1	外中断	41
10.2	定时器中断	41
10.3	LVD 中断	41
10.4	中断相关寄存器	41

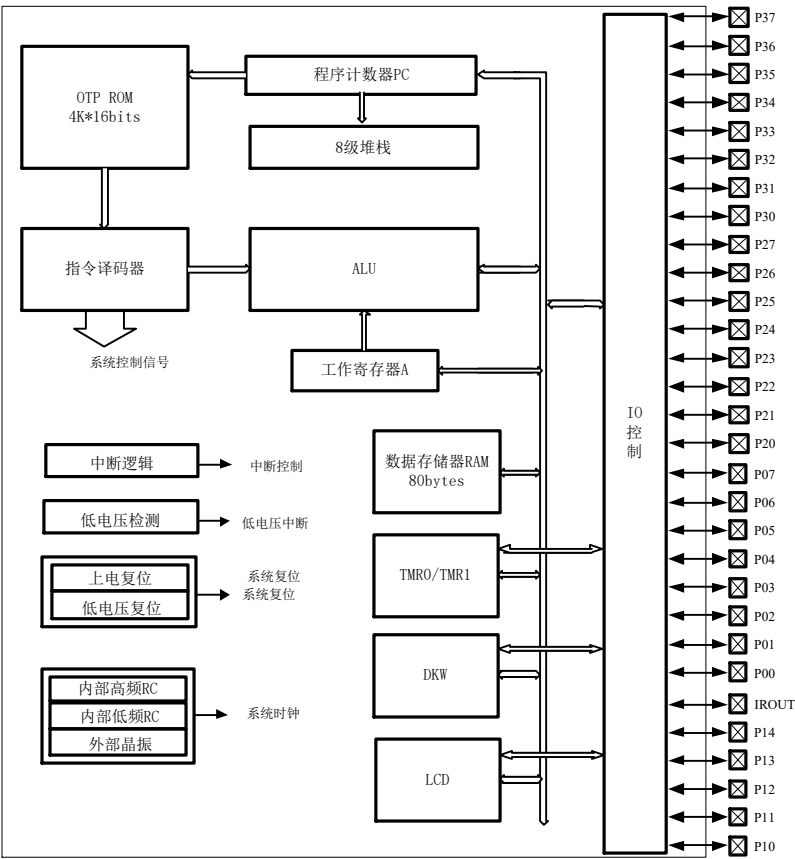
11	电气参数	43
11.1	极限参数	43
11.2	直流特性参数	43
11.3	交流电气参数	45
12	特性曲线图	46
12.1	IROUT 驱动电流 VS 输出电压	46
12.2	LCD 背光驱动电流 VS 输出电压	46
12.3	IO 输出高电平驱动电流 VS 输出电平	47
12.4	IO 输出低电平驱动电流 VS 输出电平	47
12.5	输入高低电平 VS 电源电压	48
12.6	上拉电阻 VS 电源电压	48
12.11	HOLD 模式 1 功耗 VS 电源电压	49
12.12	HOLD 模式 2（内部低频）功耗 VS 电源电压	50
12.13	HOLD 模式 2（外部低频）功耗 VS 电源电压	50
12.14	休眠模式功耗（LVR=1.8V）VS 电源电压	51
12.16	LIRC（内部低频）动态功耗 VS 电源电压 VS 温度	52
12.17	外部低频动态功耗 VS 电源电压 VS 温度	53
12.18	LVD 电压 VS 温度	54
12.19	上电复位电压 VS 温度	58
12.20	LVR 电压 VS 温度	58
12.21	HIRC=16MHz 频率	60
12.28	最低工作电压 VS 系统时钟 FCPU 关系图	63
13	封装外形尺寸	64
14	版本修订记录	65

1 产品简介

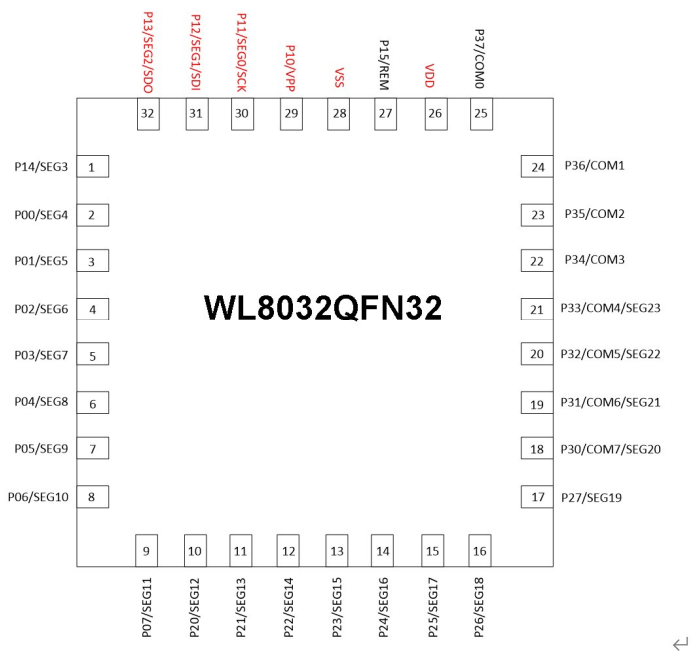
1.1 产品特性

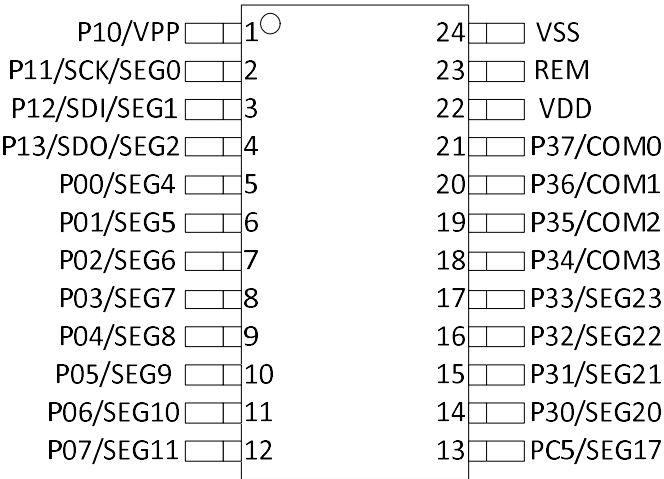
- ✧ 8 位 CPU 内核
 - ✓ 精简指令集, 仅 33 条指令
 - ✓ 高频 16MHz 模式下 4T/8T 可设;
- ✧ 4K*16 程序存储器空间 (OTP)
- ✧ 存储器
 - ✧ 8 级深度硬件堆栈
 - ✧ 80 字节通用数据寄存器空间
- ✧ 29 IO + 1 大电流 PWM 输出 IO
 - ✓ 6 位 P1 端口, P10 复用高压编程管脚 (开漏管脚); P15 驱动能力 300mA; P13、P14 与 SEG2、SEG3 复用; P11、P12 可选为外部低频晶振口也可以设置为 SEG0 和 SEG1
 - ✓ 8 位 P0 端口, 可设置为端口扫描管脚, 与 SEG4-SEG11 复用, 做 IO 口输出电流 4 级可调
 - ✓ 8 位 P2 端口, 可设置为端口扫描管脚, 与 SEG12-SEG19 复用, 做 IO 口输出电流 4 级可调
 - ✓ 8 位 P3 端口, 可设置为端口扫描管脚, 与 SEG20-SEG23 以及 COM0-COM7 复用
- ✧ 内置驱动电流 4 级可变的大电流输出管脚 IROUT
- ✧ 3 种工作模式
 - ✓ 高频运行模式: 系统在高频时钟下运行
 - ✓ 休眠模式: 所有振荡器停止运行
 - ✓ HOLD 模式: CPU 停止运行, 高频振荡器停止工作, 低频振荡器工作
- ✧ 定时器
 - ✓ 1 个 8 位定时器 T1, 1 个 16 位定时器 T0, 可设置溢出中断; T0 可作 PWM0 输出, 输出口为 P10 口, T1 作为 PWM1 输出, 可选设置输出到 IROUT 口或 P14/P13/P12/P11 上。
- ✧ 内置 4 级可调电流的 LCD 背光驱动口
- ✧ 电阻型 LCD
 - ✓ 4*24/5*23/6*22/8*20 (1/4 或 1/5 或 1/6 或 1/8 占空比, 1/3 或 1/4 偏压)
- ✧ 中断
 - ✓ 定时器 0、定时器 1 中断
 - ✓ 一路所有外部管脚电平变化中断源
- ✧ 时钟振荡模式
 - ✓ 内嵌高频振荡器 (16M/8M) + 外接低频振荡器模式 (32768Hz)
 - ✓ 内嵌高频振荡器 (16M/8M) + 内嵌低频振荡器 (36K)
- ✧ 上电复位 POR (1.8V)
- ✧ 低电压复位 LVR (1.8V-2.4V, 每 0.2V 为一档)
- ✧ 低电压监测 LVD (2.2V-3.6V, 每 0.1V 为一档)
- ✧ 更宽的工作电压, 支持 5V 作为电源
 - ✓ 2.4V-5.5V @F_{cpu}=0-8MHz
 - ✓ 2.0V-5.5V @F_{cpu}=0-4MHz
- ✧ 封装形式:
 - ✓ DIE、QFN32、SSOP24、SSOP20

1.2 系统框图

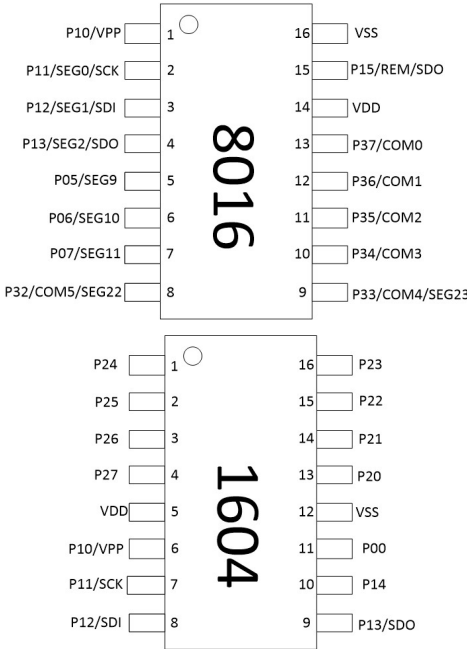


1.3 引脚排列



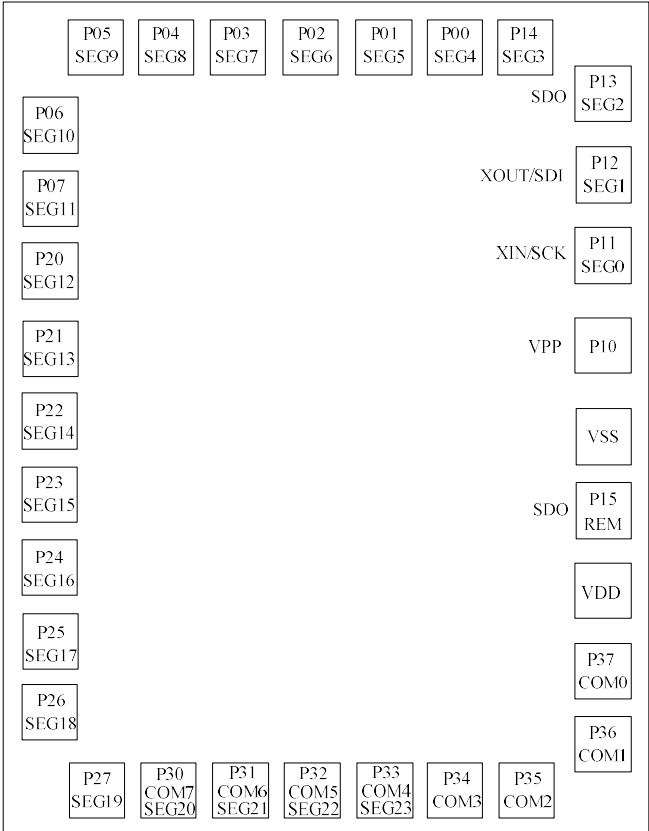


WL8024-SSOP24



SOP16 封装管脚图

烧录管脚	
VPP	P10
SCK	P11
SDI	P12
SDO	P15(IROUT)/P13



裸片绑定示意图

1.4 引脚说明

引脚名	方向	功能描述
VDD	SOURCE	电源
P10	I/O	双向 IO 口，可上拉，键扫口，高压编程管脚，PWM0 输出口，触摸按键输入口
P11-P12	I/O	双向 IO 口，可上拉，键扫口，可外接 32768 晶振，LCD 的 SEG 端口，PWM1/2 输出口
P13-P14	I/O	双向 IO 口，可上拉，键扫口，LCD 的 SEG 端口，PWM3/4 输出口
P00-P07	I/O	双向 IO 口，可上拉，键扫口，LCD 的 SEG 端口
P20-P27	I/O	双向 IO 口，可上拉，键扫口，LCD 的 SEG 端口
P30-P33	I/O	双向 IO 口，可上拉，键扫口，LCD 的 SEG/COM 端口
P34-P37	I/O	双向 IO 口，可上拉，键扫口，LCD 的 COM 端口
IROUT	O	输出口 IROUT，可设 OPENDRAIN 输出，PWM5 输出
GND	SOURCE	地

2 中央处理器

2.1 指令集

WL8032 的指令是精简指令集。下表是指令汇总表。

助记符	说明	操作	周期数	影响
ADDAR R	寄存器 R 内容和 ACC 相加, 结果存到 ACC	$R + ACC \rightarrow ACC$	1	C, DC, Z
ADDRA R	寄存器 R 内容和 ACC 相加, 结果存到 R	$R + ACC \rightarrow R$	1	C, DC, Z
RSUBAR R	寄存器 R 内容和 ACC 相减, 结果存到 ACC	$R - ACC \rightarrow ACC$	1	C, DC, Z
RSUBRA R	寄存器 R 内容和 ACC 相减, 结果存到 R	$R - ACC \rightarrow R$	1	C, DC, Z
ANDAR R	寄存器 R 内容和 ACC 与操作, 结果存到 ACC	$R \text{ and } ACC \rightarrow ACC$	1	Z
ANDRA R	寄存器 R 内容和 ACC 与操作, 结果存到 R	$R \text{ and } ACC \rightarrow R$	1	Z
ORAR R	寄存器 R 内容和 ACC 或操作, 结果存到 ACC	$R \text{ or } ACC \rightarrow ACC$	1	Z
ORRA R	寄存器 R 内容和 ACC 或操作, 结果存到 R	$R \text{ or } ACC \rightarrow R$	1	Z
XORAR R	寄存器 R 内容和 ACC 异或操作, 结果存到 ACC	$R \text{ xor } ACC \rightarrow ACC$	1	Z
XORRA R	寄存器 R 内容和 ACC 异或操作, 结果存到 R	$R \text{ xor } ACC \rightarrow R$	1	Z
COMAR R	对 R 取反, 结果存到 ACC	$R \text{ 取反} \rightarrow ACC$	1	Z
COMR R	对 R 取反, 结果存到 R	$R \text{ 取反} \rightarrow R$	1	Z
CLRA	对 ACC 清零	$0 \rightarrow ACC$	1	Z
CLRR R	对 R 清零	$0 \rightarrow R$	1	Z
RLA	ACC 循环左移(带 C 标志)	$ACC[7] \rightarrow C$ $ACC[6:0] \rightarrow ACC[7:1]$ $C \rightarrow ACC[0]$	1	C
RLAR R	寄存器 R 循环左移(带 C 标志), 结果存到 ACC	$R[7] \rightarrow C$ $R[6:0] \rightarrow ACC[7:1]$ $C \rightarrow ACC[0]$	1	C
RLR R	寄存器 R 循环左移(带 C 标志), 结果存到 R	$R[7] \rightarrow C$ $R[6:0] \rightarrow R[7:1]$ $C \rightarrow R[0]$	1	C
RRA	ACC 循环右移(带 C 标志)	$C \rightarrow ACC[7]$ $ACC[7:1] \rightarrow ACC[6:0]$ $ACC[0] \rightarrow C$	1	C
RRAR R	寄存器 R 循环右移(带 C 标志), 结果存到 ACC	$C \rightarrow ACC[7]$ $R[7:1] \rightarrow ACC[6:0]$ $R[0] \rightarrow C$	1	C
RRR R	寄存器 R 循环右移(带 C 标志), 结果存到 R	$C \rightarrow R[7]$ $R[7:1] \rightarrow R[6:0]$ $R[0] \rightarrow C$	1	C
SWAPAR R	交换 R 的高低字节, 结果存到 ACC	$R[7:4] \rightarrow ACC[3:0]$ $R[3:0] \rightarrow ACC[7:4]$	1	-
SWAPR R	交换 R 的高低字节, 结果存到 R	$R[7:4] \rightarrow R[3:0]$ $R[3:0] \rightarrow R[7:4]$	1	-
MOVAR R	将 R 存到 ACC	$R \rightarrow ACC$	1	Z
MOVRA R	将 ACC 存到 R	$ACC \rightarrow R$	1	-
INCA	ACC 自加 1	$ACC + 1 \rightarrow ACC$	1	-
INCAR R	R 加 1, 结果存到 ACC	$R + 1 \rightarrow ACC$	1	Z

INCR R	R 加 1, 结果存到 R	$R+1 \rightarrow R$	1	Z
DECA	ACC 自减 1	$ACC-1 \rightarrow ACC$	1	-
DECAR R	R 减 1, 结果存到 ACC	$R-1 \rightarrow ACC$	1	Z
DECR R	R 减 1, 结果存到 R	$R-1 \rightarrow R$	1	Z
JZA	ACC 自加 1; 结果为 0, 则跳过下一条指令	$ACC+1 \rightarrow ACC$, 结果为 0, 则 $PC+2 \rightarrow PC$	1 或 2	-
JZAR R	R 加 1, 结果存到 ACC; 结果为 0, 则跳过下一条指令	$R+1 \rightarrow ACC$, 结果为 0, 则 $PC+2 \rightarrow PC$	1 或 2	-
JZR R	R 加 1, 结果存到 R; 结果为 0, 则跳过下一条指令	$R+1 \rightarrow R$, 结果为 0, 则 $PC+2 \rightarrow PC$	1 或 2	-
DJZA	ACC 自减 1; 结果为 0, 则跳过下一条指令	$ACC-1 \rightarrow ACC$, 结果为 0, 则 $PC+2 \rightarrow PC$	1 或 2	-
DJZAR R	R 减 1, 结果存到 ACC; 结果为 0, 则跳过下一条指令	$R-1 \rightarrow ACC$, 结果为 0, 则 $PC+2 \rightarrow PC$	1 或 2	-
DJZR R	R 减 1, 结果存到 R; 结果为 0, 则跳过下一条指令	$R-1 \rightarrow R$, 结果为 0, 则 $PC+2 \rightarrow PC$	1 或 2	-
BCLR R,b	对 R 的第 b 位清零	$0 \rightarrow R[b]$	1	-
BSET R,b	对 R 的第 b 位置 1	$1 \rightarrow R[b]$	1	-
JBCLR R,b	如 R 的第 b 位为 0, 则跳过下一条指令	如 $R[b]=0$, 则 $PC+2 \rightarrow PC$	1 或 2	-
JBSET R,b	如 R 的第 b 位为 1, 则跳过下一条指令	如 $R[b]=1$, 则 $PC+2 \rightarrow PC$	1 或 2	-
ADDAI K	立即数 K 和 ACC 相加, 结果存到 ACC	$K+ACC \rightarrow ACC$	1	C, DC, Z
ISUBAI K	立即数和 ACC 相减, 结果存到 ACC	$K-ACC \rightarrow ACC$	1	C, DC, Z
ANDAI K	立即数 K 和 ACC 与操作, 结果存到 ACC	$K \text{ and } ACC \rightarrow ACC$	1	Z
ORAI K	立即数 K 和 ACC 或操作, 结果存到 ACC	$K \text{ or } ACC \rightarrow ACC$	1	Z
XORAI K	立即数和 ACC 异或, 结果存到 ACC	$K \text{ xor } ACC \rightarrow ACC$	1	Z
MOVAI K	将立即数存到 ACC	$K \rightarrow ACC$	1	-
RETAI K	从子程序返回, 并将立即数存到 ACC	$TOS \rightarrow PC$ $K \rightarrow ACC$	2	-
RETURN	从子程序返回	$TOS \rightarrow PC$	2	-
RETIE	从中断返回	$TOS \rightarrow PC$ $1 \rightarrow GIE$	2	-
CALL K	子程序调用	$PC+1 \rightarrow TOS$ $K \rightarrow PC$	2	-
GOTO K	无条件跳转	$K \rightarrow PC$	2	-
NOP	空操作	空操作	1	-
CLRWDI	清看门狗定时器	$0 \rightarrow WDT$	1	TO, PD
STOP	进入休眠模式	$0 \rightarrow WDT$ 进入休眠模式	1	TO, PD

2.2 程序存储器 (OTP)

4K*16BIT 的程序存储器空间，程序存储器空间 (0000H - 0FF7H) 可通过 GOTO 或 CALL 直接访问

复位向量 (0000H)
通用程序区 (0001H - 0007H)
中断向量 (0008H)
通用程序区 (0009H - 0FFFH)

例:通过 **INDF** 访问 **FSR** 指向的程序存储器中内容

```
MOVAI    55H
MOVRA    FSR    ; 将 55H 写入 FSR
MOVAR    INDF   ; 读取 FSR 指向 (55H) 程序存储器的内容,
```

数据存储器

数据寄存器分为三个区，快速通用寄存器区GPR（160Byte空间）和特殊功能寄存器区SFR和LCD显示RAM数据区，具体地址分配参照下表。

数据存储器区地址映射表:

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
00H - 07H	INDF	FSR	PCL	PFALG	LVDCR	INTE	INTF	DKW
08H - 0FH	OSCM	T0DATA1	PWMOE	LCDCR0	LCDCR1	TRILOSC	LVR	TRIHOSC
10H - 17H	T0CR	T0CNT	T0LOAD	T0DATA	T1CR	T1CNT	T1LOAD	PWM1DATA
18H - 1FH	PWM2DATA	PWM3DATA	PWM4DATA	PWM5DATA				
20H - 27H	IOP0	OEP0	PUP0	ODP0	DKWP0	LCDSP0		
28H - 2FH	IOP1	OEP1	PUP1	ODP1	DKWP1	LCDSP1		
30H - 37H	IOP2	OEP2	PUP2	ODP2	DKWP2	LCDSP2		
38H - 3FH	IOP3	OEP3	PUP3	ODP3	DKWP3	LCDSP3		
40H - 57H	LCDRAM(直接寻址/INDF)							
59H - 5FH	保留							
60H - FFH	通用数据区(直接寻址/INDF) (最低位必须为1可访问，为0禁止访问)							

注：上表中灰色部分数据存储器地址未用，读出数据为 0

TRILOSC 和 **TRIHOSC** 以及 **LVR** 三个寄存器分别放置烧录调校频率的值，一般使用中不用也最好不要改动；

数据寄存器地址组成

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

直接寻址模式:以指令的低8位作为数据存储器地址

例:通过直接寻址模式把 **55H** 数据写入 **10H** 地址

MOVAI 55H

MOVRA 10H ; 把数据 55H 写入 10H 地址数据存储器中

间接寻址模式0:当访问INDF时，FSR作为数据存储器地址

例:通过间接寻址模式把 **55H** 数据写入 **10H** 地址

MOVAI 10H

MOVRA FSR

MOVAI 55H

MOVRA INDF ; 把数据 55H 写入 FSR 指向数据存储器中

2.3 堆栈

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

2.4 烧录配置选项

用户配置字简称OPBIT是OTP中的3个特殊字，用于对系统功能进行配置。OPBIT在烧写用户程序时通过专用烧写器来设置。WL8032的OPBIT定义如下。

OPBIT0:

位	符号	功能说明
BIT[0]	FCPU	高速模式 Fcpu 速度选择 0:机器周期 Fcpu 为 4 个高速时钟周期 Fhosc/4 1:机器周期 Fcpu 为 8 个高速时钟周期 Fhosc/8
BIT[2:1]	VLVRS	系统复位电压选择位 11: LVR 电压=2.4V (参考值) 10: LVR 电压=2.2V (参考值) 01: LVR 电压=2.0V (参考值) 00: LVR 电压=1.8V (参考值)
BIT[3]	- GOTP	-低功耗模式选择位 0: 关闭 1: 打开
BIT[4]	ENCR	代码加密选项 0: 使能代码加密 1: 不使能代码加密

2.5 控制寄存器

WL8032全部控制寄存器列在下表中，具体功能详见各功能模块的说明。

间接寻址寄存器0

00	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INDF	INDF7	INDF6	INDF5	INDF4	INDF3	INDF2	INDF1	INDF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] INDF – 间接寻址寄存器

INDF: INDF 不是物理寄存器，对 INDF 寻址时间上是对 FSR 指向的数据存储器地址进行访问，从而实现间接寻址模式。

数据指针寄存器0

01	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSR	FSR7	FSR6	FSR5	FSR4	FSR3	FSR2	FSR1	FSR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] FSR – 数据指针寄存器

FSR: 间接寻址模式指针

程序指针计数器低位

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

BIT[7:0] PCn – 程序指针计数器低 8 位

程序指针计数器（PC）有以下几种操作模式

顺序运行指令：PC=PC+1

分支指令GOTO/CALL：PC=指令码低10位

子程序返回指令RETIE/RETURN/RETAI：PC=堆栈栈顶

对PCL操作指令：PC = (PC[9:0]+A[7:0])（对PCL操作的加法指令）

PC = {PC[9:8],ALU[7:0](ALU运算结果)}（对PCL操作的其它指

令) CPU状态寄存器

03	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PFLAG	-	LVD	PD	TO	-	Z		C
R/W	-	-	-	-	-	R/W		R/W
初始值	-	-	-	-	-	X		X

BIT[7] 未用

BIT[5] LVD – 低电压检测标志

0: 检测电压低于门限电压

1: 检测电压高于门限电压

BIT[5] TO – 看门狗溢出标志

	0: 上电复位, 执行CLRWDT或STOP指令
	1: 发生WDT溢出
BIT[4]	PD – 进入低功耗休眠模式标志
	0: 上电复位, 执行CLRWDT
	1: 执行STOP指令
BIT[3]	未用
BIT[2]	Z – 零标志
	0: 算术或逻辑运算的结果不为零
	1: 算术或逻辑运算的结果为零
BIT[1]	未用
BIT[0]	C – 进位标志
	0: 加法运算时没有进位/减法运算时有借位发生/移位后移出逻辑0
	1: 加法运算时有进位/减法运算时没有借位发生/移位后移出逻辑1

低压检测寄存器

04	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LVDCR	WDTE	TOUCH	LV DEN	LV DCK	LV DS3	LV DS2	LV DS1	LV DS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] WDTE – 看门狗功能使能位

- 1: 使能看门狗功能
- 0: 不使能看门狗功能

BIT[6] TOUCH – 触摸功能使能位,这里是利用内部的低频振荡器在外接上电容情况下可降低振荡频率的特性来实现触摸的功能;

- 1: 使能 P33 口做外接触摸功能, 当使用 P33 做触摸, 需要关闭 P33 的其他功能
- 0: 不使能触摸功能

BIT[5] LV DEN – LVD 功能使能位

- 1: 使能 LVD 功能 (使能 LVD, 仅仅给分压电阻通电, LVD 比较器还要 LV DCK 来开启)
- 0: 不使能 LVD 功能

BIT[4] LV DCK – LVD 比较器工作开始

- 1: 开始 LVD 检测 (需要先把 LV DEN 使能打开)
- 0: LVD 比较器不工作

LVD 功能内部包含一个比较器, 比较器平时不工作, 只有 LV DCK 的上升沿瞬间才开始工作, 通常比较器的工作时间只有 100nS, 并且只比较一次后就锁存该比较后的数据;

BIT[3:0] LV DS[3:0] – LVD 检测电压选择

LV DS[3:0]	LVD 检测电压设置
0000	2.1V
0001	2.2V
0010	2.3V
0011	2.4V
0100	2.5V
0101	2.6V
0110	2.7V
0111	2.8V

1000	2.9V
1001	3.0V
1010	3.1V
1011	3.2V
1100	3.3V
1101	3.4V
1110	3.5V
1111	3.6V

3 系统时钟

WL8032 为双时钟系统，可根据需要通过软件在高速时钟和低速时钟之间任意切换。高速时钟为内置 16MHz 的高精度 RC 振荡器。系统选用高速时钟时 CPU 的机器周期 F_{cpu} 由寄存器 FCPU 配置。

3.1 外接晶体振荡器

WL8032 可外接 32768 晶体振荡器，该振荡器可用于系统低频时钟。在实际使用中，用户应使晶体离 OSCI、OSCO 引脚的距离尽可能短，这样有助于振荡器的起振和振荡的稳定性。

注 1：32768 晶体振荡器输入输出引脚在芯片内部已接入 $CG=CD=10pF$ 电容对地。根据不同晶体的参数，可通过晶体两端外接对地电容调整 32768 晶体精度。

注 2：外接电容调整精度根据如下规则：

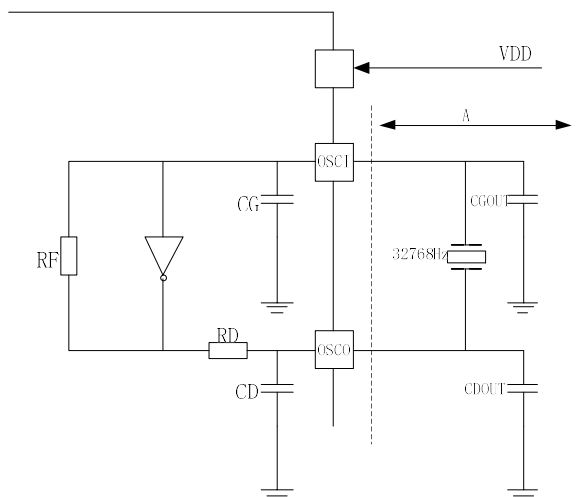
32768 晶体本身负载电容 CL 严格等于晶体两端的精度调整电容 $CADJ$ （注： $CADJ$ 为晶体两端负载电容并联值），可保证 32768 频率精准。

一般这种关系如下式：

$$CADJ = \frac{CG * CD}{CG + CD} + CS \quad CS: PCB \text{ 上的浮动电容。}$$

如果 $CADJ < CL$ ，晶体振荡频率会大于 32768，偏快，此时可以增大 $CADJ$ ，或者更换 CL 小的晶体。

如果 $CADJ > CL$ ，晶体振荡频率会小于 32768，偏慢，此时可以减小 $CADJ$ ，或者更换 CL 大的晶体。（见下图）



典型晶体参数：

X'_{tal} : 32.768KHz

($R1=30K\Omega$)

($CL=6.25pF$ 或者 $12.5pF$)

芯片典型内部参数

$RF=100M\Omega$

$RD=1K\Omega$

$CG=CD=10pF$

外接电容参数

$CGOUT=CDOUT=0\sim 15pF$

- ✧ 外接晶体尽可能的靠近芯片 IC。
- ✧ 在 PCB 布局时，靠近晶振电路附近不可以走任何信号线或者电源线，特别是上图标识的 A 区域内。
- ✧ 在 PCB 布线时，在 OSCI 和 OSCO 引脚之间使用高绝缘电阻。
- ✧ 在 PCB 布线时，避免使用长的平行连线连接 OSCI 和 OSCO 引脚
- ✧ 为了得到稳定的振荡频率，请勿使用 OSCO 引脚驱动其它 IC

注 3：数字化调整秒走时精度

通过 **TIMER0** 选择 **32768** 晶体做时钟源，选择 **1** 分频溢出中断输出 **16384** 频率，此频率实为 **32768** 晶体本身振荡频率二分频。可用频率计测试出中断输出频率，一般说来会与标准 **32768** 频率有差异。而 **1** 秒钟信号是 **32768** 频率经过 **32768** 次计数产生。根据这种计秒原理，可选择合适的计数次数以达到准确 **1** 秒信号。

- ✧ 如果测试出来的频率*2 大于 **32768**，比如为 **32769**，则可以将计数次数调整为 **32769** 次为 **1** 秒信号。
- ✧ 如果测试出来的频率*2 小于 **32768**，比如为 **32767**，则可心将计数次调整为 **32767** 次为 **1** 秒信号。
这样可以保证 **1** 秒的信号是准确的。精度可以保证 $\pm 0.5\text{Hz}/32768\text{Hz} = \pm 15\text{ppm}$ ，换算到 **30** 天 (**1** 个月)，偏差 **38.8** 秒。
- ✧ 如果再提高精度，可以 **10** 秒进行调整 **1** 次计数值，精度可以从 $\pm 15\text{ppm}$ 提高至 $\pm 1.5\text{ppm}$ ，换算到 **30** 天 (**1** 个月)，偏差 **3.88** 秒。

3.2 内置高频 RC 振荡器

WL8032 内置高精度 16MHz 高频 RC 振荡器，该振荡器可用于系统高速时钟。

3.3 内置低频 RC 振荡器

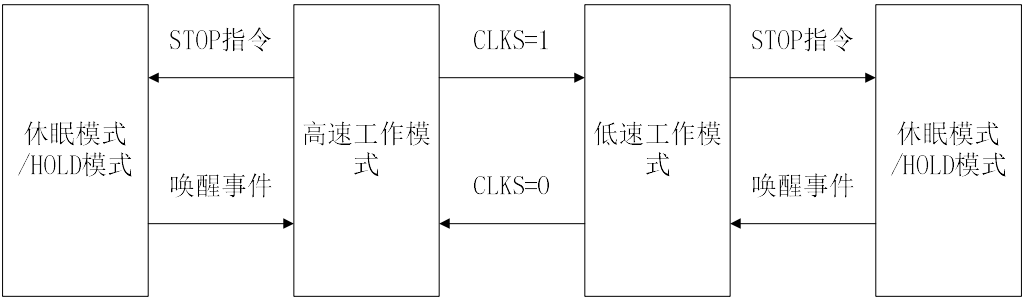
WL8032 内置一个低频 RC 振荡器，该振荡器可用于系统低频时钟，同时用于上电延时、WDT，键盘扫描。该振荡器频率典型值 36KHz。

3.4 工作模式

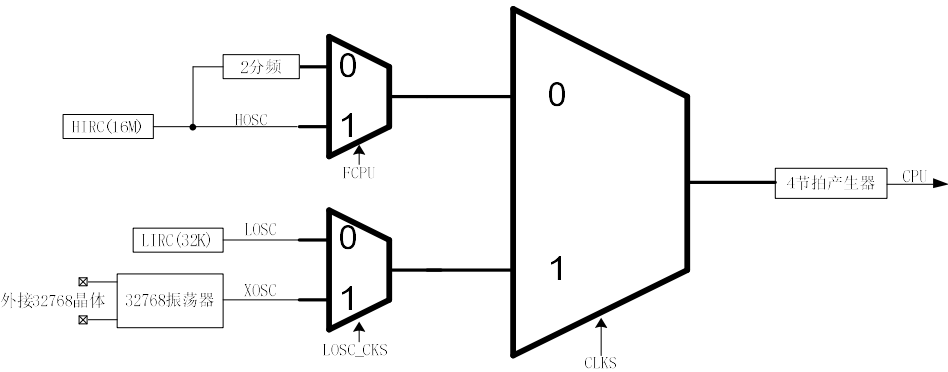
WL8032 支持高速工作模式、低速工作模式、休眠模式、HOLD 模式 1 和 HOLD 模式 2 共有 5 种工作模式。

工作模式	进入条件
高速工作模式	系统时钟切换到高频振荡器 (CLKS=0)
低速工作模式	系统时钟切换到低频振荡器 (CLKS=1)
休眠模式	执行 STOP 指令, HFEN=0, LFEN=0
HOLD 模式 1	执行 STOP 指令, HFEN=1, LFEN=X (定时器可在高速时钟模式下继续工作, 溢出可唤醒)
HOLD 模式 2	执行 STOP 指令, HFEN=0, LFEN=1 (定时器可在低速时钟模式下继续工作, 溢出可唤醒)

工作模式间的切换



系统时钟选择



注意，上述 CLKS 不可以为 1，CLKS=1 的模式不可用；

	高速工作模式	低速工作模式	休眠模式/HOLD 模式
高频振荡器	工作	HFEN 决定	HFEN 决定
低频振荡器	工作	工作	LFEN 决定
WDT 振荡器	工作	工作	WDTC 决定

注：当低速时钟选择为内部低频 RC 振荡器，则低频振荡器和 WDT 振荡器共用同一振荡器，低频振荡器工作或 WDT 振荡器工作都会使内部低频 RC 振荡器工作。

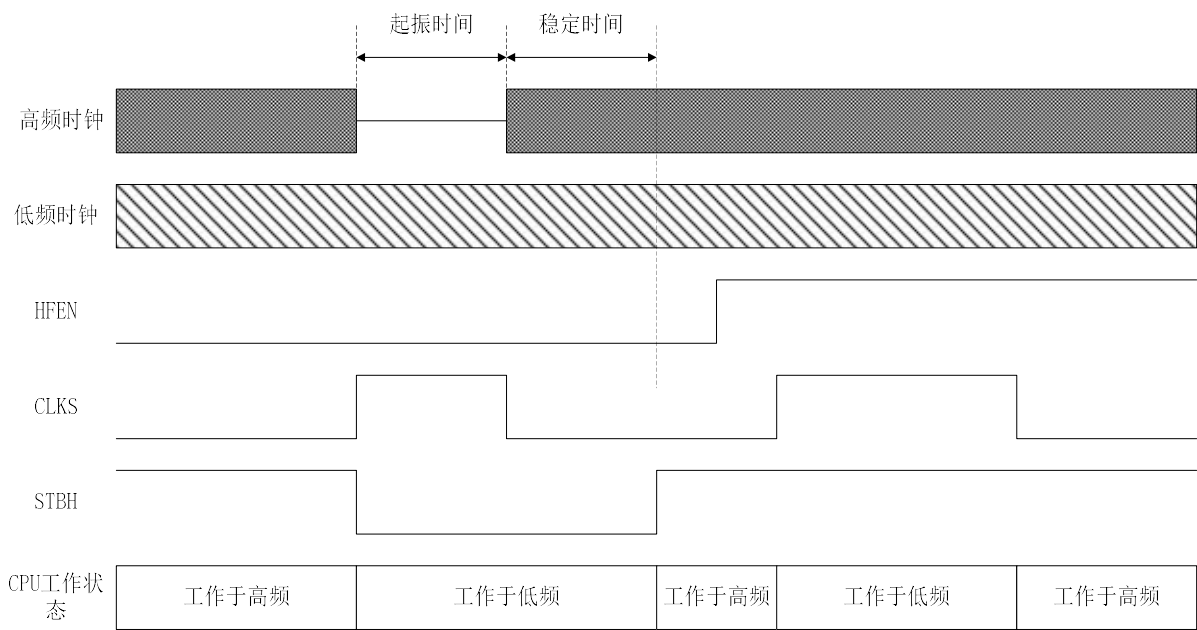
工作模式寄存器

08	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
----	-------	-------	-------	-------	-------	-------	-------	-------

OSCM	GOTP	-	-	-	LOSC_CLKS	CLKS	LFEN	HFEN
R/W	R/W	-	-	-	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

- BIT[7]** **GOTP** – OTP 采用绿色模式运行（不要使用，保持默认为 0）
0: 正常模式
1: 绿色模式（可降低功耗）
- BIT[6:4]** 未用
- BIT[3]** **LOSC_CLKS** – 系统工作时钟选择位（不要使用，保持默认为 0）
0: 内置低频时钟作为低频时钟
1: 外接低频32768时钟作为低频时钟
- BIT[2]** **CLKS** – 系统工作时钟选择位(不要使用，保持默认为 0)
0: 高频时钟作为系统时钟
1: 低频时钟作为系统时钟
- BIT[1]** **LFEN** – 低频振荡器使能（仅当 CPU 在高频运行的时候可写使用，其他由系统默认）
0: 在休眠/HOLD模式下，低频振荡器停止工作
1: 低频振荡器始终工作，在高频模式下强行写入1
- BIT[0]** **HFEN** – 高频振荡器使能（不可用，保持为默认 1）
0: 在低速/休眠/HOLD模式下，高频振荡器停止工作
1: 高频振荡器始终工作

高低速时钟切换时序图



3.5 低功耗模式

低功耗模式包括休眠模式、HOLD 模式 1、HOLD 模式 2

STOP 指令可使 MCU 进入低功耗模式，同时对 MCU 会产生以下影响

- ✧ 根据不同模式停止相应振荡器振荡
- ✧ RAM 内容保持不变
- ✧ 所有的输入输出端口保持不变
- ✧ 定时器根据其工作模式，可以保持继续工作

以下情况可使 MCU 退出低功耗模式

- ✧ 有外部中断请求发生
- ✧ 定时器 0、1 溢出中断请求发生
- ✧ 复位

MCU 退出低功耗模式后经过振荡等待（外部低频晶振等待 1024 个周期，内部高频等待 32 个周期，内部低频等待 4 个周期）然后开始工作。

4 复位

4.1 复位条件

WL8032 有 2 种复位方式

- ✧ 上电复位 POR
- ✧ 低电压复位 LVR

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

上电复位和低电压复位会关闭系统主时钟的振荡器，复位解除后才重新打开振荡器，系统会在上电延时后经过振荡等待（外部低频晶振等待 1024 个周期，内部高频等待 16 个周期，内部低频等待 16 个周期）然后开始工作。

4.2 上电复位

WL8032 的上电复位电路可以适应快速、慢速上电的情况，并且当芯片上电过程中出现电源电压抖动时都能保证系统可靠的复位。

上电复位过程系统工作步骤：

- ✧ 等待电压高于 VPOR 并保持稳定
- ✧ 开启内部低频振荡器加载 OPBIT，并根据 OPBIT 设置开启相应振荡器并进行相应时间的上电延时
- ✧ 等待电压高于 VLVR 才开始上电延时计数
- ✧ 上电延时结束，振荡等待结束后系统开始执行指令

4.3 低电压复位

WL8032 的 LVR 电压有 4 级（详见烧录配置选项），通过 OPBIT 的 VLVR5 进行配置。电压检测电路有一定的回滞特性，通常回滞电压为 0.02V 左右，则当电源电压下降到 LVR 电压时 LVR 复位有效，而电压需要上升到 LVR 电压+0.02V 时 LVR 复位才会解除。

LVR 寄存器，注意，此寄存器请谨慎使用，部分 bit 位建议使用 bset 或者 bclr 指令设置；

0x0E	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LVR	STATE_CNT[1:0]		-	FCPU	LVR_CNT[1:0]		TRIM_LVD[1:0]	
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

STATE_CNT[1:0]:烧录状态寄存器

仅烧录时可以置为 1，其他正常使用时，不要设置该寄存器；

FCPU: 建议使用 bset 或者 bclr 指令设置

1: CPU 指令速度 8M/2T

0: CPU 指令速度 4M/2T

LVR_CNT[1:0]:低电压复位设置寄存器，建议使用 bset 或者 bclr 指令设置

00: 复位电压选择 1.8V

01: 复位电压选择 2.0V

10: 复位电压选择 2.2V

11: 复位电压选择 2.4V

TRIM_LVD[1:0]: 校准低电压检测寄存器位

此寄存器为烧录的时候校准使用，通常应用中无须改变此寄存器位中的数据；

5 IO口

5.1 IO 工作模式

芯片共有 4 组 30 位普通端口 P15-P10, P07-P00, P27-P20, P37-P30 组成, 其中 1 位大电流输出端口 P15, 同时 P10 该口为高压 VPP 输入口。同时 P14-P10, P07-P00, P27-P20, P37-P30 可设置为端口扫描管脚; P14-P11, P07-P00, P27-P20, P37-P30 和 LCD 的 SEG 口 COM 口复用; P14-P11, P07-P00, P27-P10 可设置 LCD 背光驱动, 同时可设置 4 级驱动电流输出; P13 设置为外部中断 INT0 的输入口, 同时为 TIMER0 的外部时钟输入口; P11 可设置为外部晶振输入口 OSCI, P12 可设置为外部晶振输出口 OSC0。

除 P15 为大电流驱动外, 其余的各口均受 6 个寄存器位控制, 分别是输入输出方向控制寄存器 OEP, 输入输出数据寄存器 IOP, 上拉选择寄存器 PUP, OD 控制寄存器, DKWP 键盘扫描控制寄存器, IO 第 2 功能 (SEG/COM) 控制寄存器;

端口数据寄存器

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

BIT[7:0] P0nD – P0 口数据位 (n=7-0)

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

BIT[5:0] P1nD – P1 口数据位 (n=5-0)

30	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOP2	P27D	P26D	P25D	P24D	P23D	P22D	P21D	P20D
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] P2nD – P2 口数据位 (n=7-0)

38	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOP3	P37D	P36D	P35D	P34D	P33D	P32D	P31D	P30D
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	X	X	X	X	X	X	X	X

BIT[7:0] P3nD – P3 口数据位 (n=7-0)

端口方向寄存器

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

BIT[7:0] P0nOE – P0 口输出使能寄存器 (n=7-0)

0:作为输入口, 读 P0 口读取端口状态

1:作为输出口, 读 P0 口读取 P0 口数据寄存器值

29	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
----	-------	-------	-------	-------	-------	-------	-------	-------

OEP1	-	-	-	P14OE	P13OE	P12OE	P11OE	P10OE
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
初始值	-	-	-	0	0	0	0	0

BIT[4:0] P1nOE – P1 口输出使能寄存器 (n=4-0)

0:作为输入口, 读 P1 口读取端口状态

1:作为输出口, 读 P1 口读取 P1 口数据寄存器值

注意: P15 口作为 REM 口, 仅能做输出, 不可以做输入;

31	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OEP2	P27OE	P26OE	P25OE	P24OE	P23OE	P22OE	P21OE	P20OE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] P2nOE – P2 口输出使能寄存器 (n=7-0)

0:作为输入口, 读 P2 口读取端口状态

1:作为输出口, 读 P2 口读取 P2 口数据寄存器值

39	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OEP3	P37OE	P36OE	P35OE	P34OE	P33OE	P32OE	P31OE	P30OE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] P3nOE – P3 口输出使能寄存器 (n=7-0)

0:作为输入口, 读 P3 口读取端口状态

1:作为输出口, 读 P3 口读取 P3 口数据寄存器值

5.2 上拉电阻控制

P0, P1, P2, P3 每位都有独立的上拉控制寄存器位，控制其上拉电阻在端口作为输入状态时是使用 40K 的内部上拉电阻还是使用内部 100K 的上拉电阻，端口处于输出状态时，上拉电阻控制位无效。

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

BIT[7:0] P0nPU – P0 口上拉电阻选择 (n=7-0)

0:P0n 上拉电阻选择 100K

1:P0n 上拉电阻选择 40K

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

BIT[4:0] P1nPU – P1 口上拉电阻选择 (n=4-0)

0:P1n 上拉电阻选择 100K

1:P1n 上拉电阻选择 40K

32	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PUP2	P27PU	P26PU	P25PU	P24PU	P23PU	P22PU	P21PU	P20PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] P2nPU – P2 口上拉电阻选择 (n=7-0)

0:P2n 上拉电阻选择 100K

1:P2n 上拉电阻选择 40K

3A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PUP3	P37PU	P36PU	P35PU	P34PU	P33PU	P32PU	P31PU	P30PU
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] P3nPU – P3 口上拉电阻选择 (n=7-0)

0:P3n 上拉电阻选择 100K

1:P3n 上拉电阻选择 40K

5.3 端口模式控制

P0, P1, P2, P3 口可以作为通用 IO 口，当设置为输入口时，也可以矩型键盘口，通过设置 DKWP0, DKWP1, DKWP2, DKWP3 使其在休眠模式下可以自动进行端口扫描。

24	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DKWP0	P07DKW	P06DKW	P05DKW	P04DKW	P03DKW	P02DKW	P01DKW	P00DKW
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] P0nDKW – P0 口模式选择 (n=7-0)，当 PBIE 设置为 1 时，可以引起中断唤醒

1: P0n 端口作为扫描键盘口

0: P0n 端口作为通用 IO 口

2C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DKWP1	-	-	-	P14DKW	P13DKW	P12DKW	P11DKW	P10DKW
R/W	-	-	-	R/W	R/W	R/W	R/W	R/W
初始值	-	-	-	0	0	0	0	0

BIT[4:0] P0nDKW – P1 口模式选择 (n=4-0)，当 PBIE 设置为 1 时，可以引起中断唤醒

1: P1n 端口作为扫描键盘口

0: P1n 端口作为通用 IO 口

34	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DKWP2	P27DKW	P26DKW	P25DKW	P24DKW	P23DKW	P22DKW	P21DKW	P20DKW
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] P2nDKW – P2 口模式选择 (n=7-0)

1: P2n 端口作为扫描键盘口，当 PBIE 设置为 1 时，可以引起中断唤醒

0: P2n 端口作为通用 IO 口

3C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DKWP3	P37DKW	P36DKW	P35DKW	P34DKW	P33DKW	P32DKW	P31DKW	P30DKW
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] P3nDKW – P3 口模式选择 (n=7-0)

1: P3n 端口作为扫描键盘口，当 PBIE 设置为 1 时，可以引起中断唤醒

0: P3n 端口作为通用 IO 口

5.4 端口模式控制

P0, P1, P2, P3 口可以作为通用 IO 口，当设置为 OD 门输出，就是不能输出高电平，通过设置 ODP0, ODP1, ODP2, ODP3 使其在输出模式下，按 Open-Drain 的方式输出。

23	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ODP0	P07OD	P06OD	P05OD	P04OD	P03OD	P02OD	P01OD	P00OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] P0nOD – P0 口模式选择 (n=7-0)

1: P0n 端口作为开漏输出方式

0: P0n 端口作为通用 IO 口

2B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ODP1	-	-	P15OD	P14OD	P13OD	P12OD	P11OD	P10OD
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	-	-	1	0	0	0	0	0

BIT[5:0] P1nOD – P1 口模式选择 (n=5-0)

1: P1n 端口作为开漏输出方式

0: P1n 端口作为通用 IO 口

33	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ODP2	P27OD	P26OD	P25OD	P24OD	P23OD	P22OD	P21OD	P20OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] P2nOD – P2 口模式选择 (n=7-0)

1: P2n 端口作为开漏输出方式

0: P2n 端口作为通用 IO 口

3B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ODP3	P37OD	P36OD	P35OD	P34OD	P33OD	P32OD	P31OD	P30OD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] P3nOD – P3 口模式选择 (n=7-0)

1: P3n 端口作为开漏输出方式

0: P3n 端口作为通用 IO 口

5.5 端口模式控制

P0, P1, P2, P3 口可以作为通用 IO 口, 也可以设置为对应的 SEG 或 COM 功能输出, 通过设置 SEGP0, SEGP1, SEGP2, COMP3 使其在输出模式下, 按设置的 IO 口的第 2 种方式输出。

25	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LCDSP0	P07SEG	P06SEG	P05SEG	P04SEG	P03SEG	P02SEG	P01SEG	P00SEG
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] P0nSEG – P0 口模式选择 (n=7-0)

1: P0n 端口作为 SEG 输出方式(P00SEG=SEG4,P01SEG=SEG5,..... P07SEG=SEG11)

0: P0n 端口作为通用 IO 口

2D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LCDSP1	-	-	-	P14SEG	P13SEG	P12SEG	P11SEG	-
R/W	-	-	-	R/W	R/W	R/W	R/W	-
初始值	-	-	-	0	0	0	0	-

BIT[4:1] P1nSEG – P1 口模式选择 (n=4-1)

1: P1n 端口作为 SEG 输出方式(P11SEG=SEG0,P12SEG=SEG1,P13SEG=SEG2,P14SEG=SEG3)

0: P1n 端口作为通用 IO 口

35	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LCDSP2	P27SEG	P26SEG	P25SEG	P24SEG	P23SEG	P22SEG	P21SEG	P20SEG
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] P2nSEG – P2 口模式选择 (n=7-0)

1: P2n 端口作为 SEG 输出方式(P20SEG=SEG12,P21SEG=SEG13,..... P27SEG=SEG19)

0: P2n 端口作为通用 IO 口

3D	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
----	-------	-------	-------	-------	-------	-------	-------	-------

LCDSP3	P37COM	P36COM	P35COM	P34COM	P33COM	P32COM	P31COM	P30COM
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] **P3nCOM** – P3 口模式选择 (n=7-0)

1: P3n 端口作为 COM/SEG 输出方式

P37COM=COM0.

P36COM=COM1,

P35COM=COM2,

P34COM=COM3,

P33COM=COM4/SEG23 (具体是做 SEG 还是 COM, 由 LCDCR0 寄存器来控制)

P32COM=COM5/SEG22 (具体是做 SEG 还是 COM, 由 LCDCR0 寄存器来控制)

P31COM=COM6/SEG21 (具体是做 SEG 还是 COM, 由 LCDCR0 寄存器来控制)

P30COM=COM7/SEG20 (具体是做 SEG 还是 COM, 由 LCDCR0 寄存器来控制)

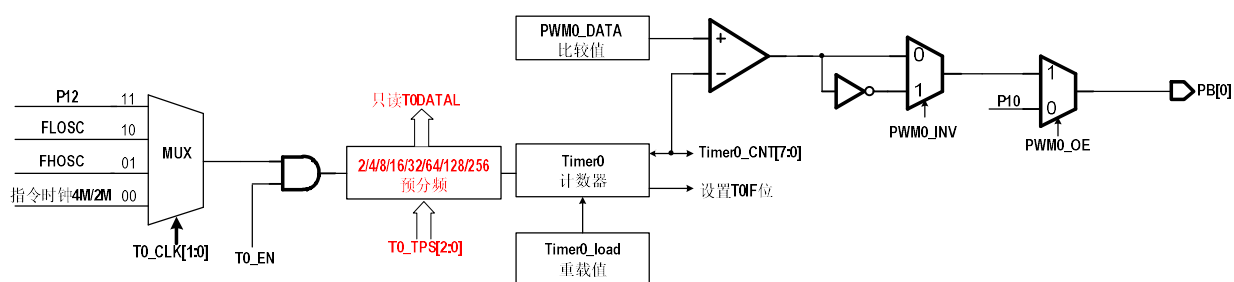
0: P3n 端口作为通用 IO 口

6 定时器

6.1 定时器 T0

定时/计数器 T0 包含 1 个可编程预分频器, 控制寄存器、重载寄存器及比较寄存器

- 可通过预分频比设置频率
- 通过重载寄存器设置周期
- 通过比较寄存器设置 PWM 占空比 (仅 PWM 模式)
- BUZ 功能
- 溢出中断功能
- 溢出唤醒功能



注 1: FH OSC 指系统高频时钟源; FLOSC 指系统内部或外部 32768 晶振低频时钟源;

注 2: 当定时器选择高频时钟源 FHOSC 且时钟控制模块 OSCM 寄存器的 HFEN=1 时, 定时器在低频工作模式或休眠模式时可继续工作, 溢出可唤醒休眠模式; 若 HFEN=0 时, 定时器在低频工作模式或休眠模式下将停止工作

注 3: 当定时器选择低频时钟源 FLOSC 且时钟控制模块 OSCM 寄存器的 LFEN=1 时, 定时器在休眠模式时可继续工作, 溢出可唤醒休眠模式; 若 LFEN=0 时, 定时器在休眠模式下将停止工作

TO_CLK 可选择 T0 的时钟源, TO_TPS 可选择 T0 的预分频比, 所选中的时钟源通过预分频器后产生

TOCNT 的时钟。

当 TOCNT 递加到 0 时，此时产生 T0 溢出中断请求标志 TOIF 置 1，重载寄存器值自动置入 TOCNT，PWMDATA 的值写入缓冲器 TODATABUF 用于新的占空比波形生成，BUZ0 信号反相。

通过 T0_TPS 可选择时钟源的分频比，可选择范围为 2-256 分频，对 TOCNT 的写操作将使预分频器清零，分频比保持不变。

当 PWM0OE=1 时，将输出 PWM 波形，当 TOCNT 计数到与 TODATA 相等时，PWM0 输出置 1；当 TOCNT 计数溢出时，PWM0 输出清 0，PWM0 占空比的计算如下：

PWM0 高电平时间 = (TODATA) * TOCNT 计数时钟周期

PWM0 周期 (T0 的溢出周期) = (TOLOAD+1) * TOCNT 的计数周期

PWM0 占空比 = (TODATA / (TOLOAD+1))

当 BUZ0OE=1 且 PWM0OE=0 时，输出 BUZ0 信号，BUZ0 信号的输出频率为 T0 溢出频率的 2 分频。

与定时器T0相关的寄存器说明如下

0x11	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TOCR	TO_EN	PWM1INV	PWM0INV	T0_CLK1	T0_CLK0	T0_TPS2	T0_TPS1	T0_TPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] TO_EN – T0 使能控制

0: 关闭 T0

1: 启动 T0

BIT[6] PWM1INV – PWM1 输出极性选择 (这里这一位是控制 T1 定时器(除 PWM5 外)的输出)

0: PWM1 原始波形输出

1: PWM1 输出原始波形的反向波形信号

BIT[5] PWM0INV – PWM0 输出极性选择

0: PWM0 原始波形输出

1: PWM0 输出原始波形的反向波形信号

BIT[4:3] T0_CLK[1:0] – T0 时钟源选择

T0PTS[1:0]	T0 时钟源
00	FHOSC
01	FCPU
10	FLOSC
11	P12

BIT[2:0] T0_TPS[2:0] – T0 预分频倍数选择

TPS2	T0TPS1	T0TPS0	TOCNT
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128
1	1	1	256

0x10	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
------	-------	-------	-------	-------	-------	-------	-------	-------

T0CNT	T0CNT7	T0CNT6	T0CNT5	T0CNT4	T0CNT3	T0CNT2	T0CNT1	T0CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T0CNT[7:0]** – T0CNT 的值，这是一个读写寄存器。

重要：在对 T0CNT 重写数据之后，建议立即对 T0IF 进行清零；

0x12	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0LOAD	T0LOAD7	T0LOAD6	T0LOAD5	T0LOAD4	T0LOAD3	T0LOAD2	T0LOAD1	T0LOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] **T0LOAD[7:0]** – T0LOAD 的值，这是一个读写寄存器，用于设置 T0 重载值。

0x13	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0DATA	T0DATA7	T0DATA6	T0DATA5	T0DATA4	T0DATA3	T0DATA2	T0DATA1	T0DATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

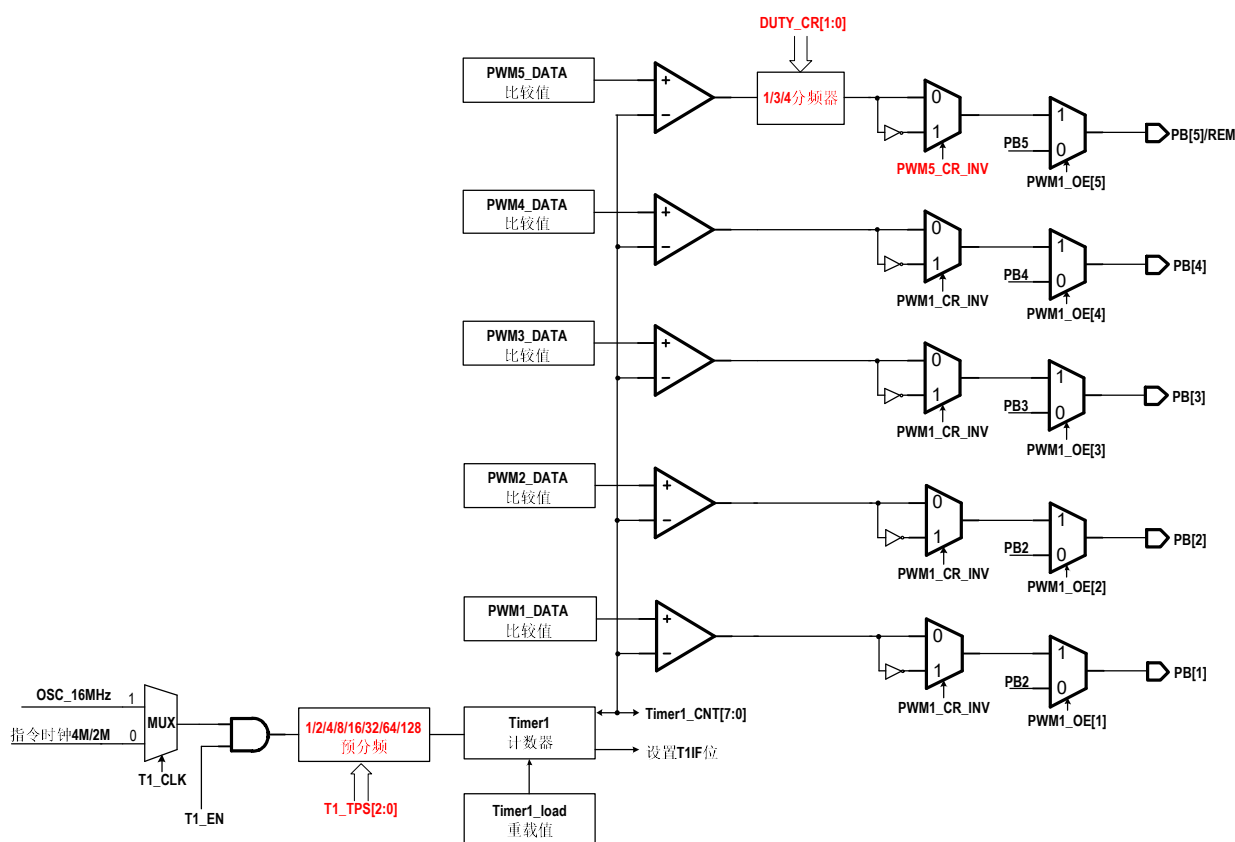
BIT[7:0] **T0DATA[7:0]** – 这是一个读写寄存器，用于设置 PWM0 低电平时间。

注：当 T0EN=0 时，写 T0LOAD 将自动加载到 T0CNT；当 T0EN=1 时，写 T0LOAD 时不自动加载到 T0CNT，在 T0 溢出时自动加载到 T0CNT

0x09	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T0DATAL	T0DL7	T0DL6	T0DL5	T0DL4	T0DL3	T0DL2	T0DL1	T0DL0
R/W	R	R	R	R	R	R	R	R
初始值	0	0	0	0	0	0	0	0

T0 也可以看作是一个 16 位的计数器，其中当 T0_TPS[2:0]=111 的时候，选择 256 分频，当开始计数后，可以通过读取寄存器 T0DATAL 的值，来获取 T0 低 8 位的计数值；当设置 T0_EN=0 时，T0DATAL 也将被复位成 0；

6.2 定时器 T1



与定时器 T1 相关的寄存器说明如下

0x15	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CR	T1EN	DUTY	DUTY	PWM5_INV	T1_CLK	T2_TPS2	T1_TPS1	T1_TPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] T1EN – T1 使能控制

0: 关闭 T1

1: 启动 T1

BIT[6:5] DUTY – 输出波形 DUTY 选择

00: 无分频, 输出 PWM5_DATA 设置的占空比波形

01: 无分频, 输出 PWM5_DATA 设置的占空比波形

10: PWM5_DATA 输出经过 3 分频后输出 1/3duty 波形

11: PWM5_DATA 输出经过 4 分频后输出 1/4duty 波形

BIT[4] PWM5_INV – T1 输出波形极性选择

0: 选择 PWM5 输出原始波形

1: 选择 PWM5 输出原始波形的反向波形

BIT[3] T1_CLK – T1 时钟源选择

T1PTS	T1 时钟源
0	FCPU
1	FHOSC

BIT[2:0] T1_TPS[2: 0] – T1 预分频倍数选择

TPS2	TPS1	TPS0	T1CNT
0	0	0	1
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

0x14	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1CNT	T1CNT7	T1CNT6	T1CNT5	T1CNT4	T1CNT3	T1CNT2	T1CNT1	T1CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T1CNT[7: 0] – T1CNT 的值，这是一个读写寄存器。

重要：在对 T1CNT 重写数据之后，建议立即对 T1IF 进行清零；

0x16	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T1LOAD	T1LOAD7	T1LOAD6	T1LOAD5	T1LOAD4	T1LOAD3	T1LOAD2	T1LOAD1	T1LOAD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	1	1	1	1	1	1	1	1

BIT[7:0] T1LOAD[7: 0] – T1LOAD 的值，这是一个读写寄存器，用于设置 T1 重载值。

0x17	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1DATA	T1DATA7	T1DATA6	T1DATA5	T1DATA4	T1DATA3	T1DATA2	T1DATA1	T1DATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] PWM1DATA[7:0] – 这是一个读写寄存器，用于设置 PWM1 低电平时间。

0x18	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM2DATA	T1DATA7	T1DATA6	T1DATA5	T1DATA4	T1DATA3	T1DATA2	T1DATA1	T1DATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] PWM2DATA[7:0] – 这是一个读写寄存器，用于设置 PWM2 低电平时间。

0x19	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM3DATA	T1DATA7	T1DATA6	T1DATA5	T1DATA4	T1DATA3	T1DATA2	T1DATA1	T1DATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] PWM3DATA[7:0] – 这是一个读写寄存器，用于设置 PWM3 低电平时间。

0x1A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM4D ATA	<i>T1DATA7</i>	<i>T1DATA6</i>	<i>T1DATA5</i>	<i>T1DATA4</i>	<i>T1DATA3</i>	<i>T1DATA2</i>	<i>T1DATA1</i>	<i>T1DATA0</i>
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] PWM4DATA[7:0] – 这是一个读写寄存器，用于设置 PWM4 低电平时间。

0x1B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM5D ATA	<i>T1DATA7</i>	<i>T1DATA6</i>	<i>T1DATA5</i>	<i>T1DATA4</i>	<i>T1DATA3</i>	<i>T1DATA2</i>	<i>T1DATA1</i>	<i>T1DATA0</i>
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7:0] PWM5DATA[7:0] – 这是一个读写寄存器，用于设置 PWM5 低电平时间。

注：当 **T1EN=0** 时，写 **T1LOAD** 将自动加载到 **T1CNT**；当 **T1EN=1** 时，写 **T1LOAD** 时不自动加载到 **T1CNT**，在 **T1** 溢出时自动加载到 **T1CNT**

0x0A	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMOE	-	-	<i>PWM5_OE</i>	<i>PWM4_OE</i>	<i>PWM3_OE</i>	<i>PWM2_OE</i>	<i>PWM1_OE</i>	<i>PWM0_OE</i>
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

当 **PWM1_OE** 相应的位为 1 时，设定对应的 P11 口的输出为 PWM 输出设定信号；例如，设置 **PWM5_OE=1**，则 P15/REM 口输出对应的 PWM 信号，具体的 PWM 信号的波形情况由上述 T1 定时器的设置为准；

7 键盘扫描模块

7.1 键盘扫描介绍

WL8032 的 P0, P1, P2, P3 均可单独分别设置为键盘扫描输入上拉模式，当系统进入休眠/HOLD 模式时，硬件自动设置相应端口为输入上拉状态。键盘扫描将自动比对当前输入口的上拉是不是被拉低，当被拉低则自动开启键盘中断。

7.2 键盘扫描操作步骤

- ✧ S1: 设置 DKWP 将相应的端口设置为键盘扫描模式
- ✧ S2: 设置 PUP 开启相应的端口的上拉电阻
- ✧ S3: 开启 PBIE 中断使能
- ✧ S4: 执行 STOP 指令
- ✧ S5: 等待端口唤醒后，系统将执行 STOP 后指令
- ✧ S6: 设置 DKWP 将相应的端口设置为通用 IO 口

注: 上述操作步骤是使能芯片硬件唤醒功能，芯片将只能由端口(设置为键盘扫描模式)的下降沿触发唤醒，此方法可使芯片支持矩形键盘。

7.3 键盘扫描相关寄存器

0x07	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DKW	–	–	Vbias_SPEED	LSEL[1]	LSEL[0]	REM_SPEED	DRV[1]	DRV[0]
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
初始值	–	–	0	0	0	0	0	0

BIT[5] Vbias_SPEED – 分压产生器加快产生选择

0: 不加速

1: 加速（可能会导致功耗略大）

BIT[4:3] LSEL[1:0]– IO 管脚驱动电流选择（可调输出电流管脚仅包括 P0X, P1X, P2X 口，不包括 P3X 的端口，P3X 共计 8 个端口做 IO 输出时的输出电流不能通过 LSEL 调节）

00: 2mA

01: 4mA

10: 8mA

11: 16mA

BIT[2] REM-SPEED– REM 口的驱动速度选择

0: 不加快驱动速度

1: 加快驱动速度

BIT[1:0] DRV[1:0]– IROUT 驱动电流选择

00: 220mA

01: 330mA

10: 450mA

11: 500mA

8 低电压检测 (LVD)

WL8032 内嵌低电压检测模块，通过设置 LVDEN 可以开启或屏蔽该功能，LVDS 选择检测电压值，可设置 2.0V-3.6V 等不同电压，当 VDD 电压低于设置电压时 LVD 置 1，否则 LVD 清 0。

0x04	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LVD CR	WDTE	X	LVDEN	LVDCK	LVDS3	LVDS2	LVDS1	LVDS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

此部分说明在前面的系统章节有介绍

BIT[5] LVDEN – LVD 功能使能位

- 1: 使能 LVD 功能
- 0: 不使能 LVD 功能

BIT[4] LVDCK –LVD 比较器工作开始

- 1: 开始 LVD 检测（需要先把 LVDEN 使能打开）
- 0: LVD 比较器不工作

LVD 功能内部包含一个比较器，比较器平时不工作，只有 LVDCK 的上升沿瞬间才开始工作，通常比较器的工作时间只有 100nS，并且只比较一次后就锁存该比较后的数据；

BIT[3:0] LVDS[3:0] – LVD 检测电压选择

LVDS[3:0]	LVD 检测电压设置
0000	2.1V
0001	2.2V
0010	2.3V
0011	2.4V
0100	2.5V
0101	2.6V
0110	2.7V
0111	2.8V
1000	2.9V
1001	3.0V
1010	3.1V
1011	3.2V
1100	3.3V
1101	3.4V
1110	3.5V
1111	3.6V

9 液晶显示控制及驱动

该模块由 LCD 控制寄存器、LCD 端口选择寄存器、以及 LCD 数据寄存器组成。

- 最大支持 8*20 的 LCD 驱动
- 可设置快速充电模式降低功耗
- 可调节帧频

➤ 可设置 B 类波形

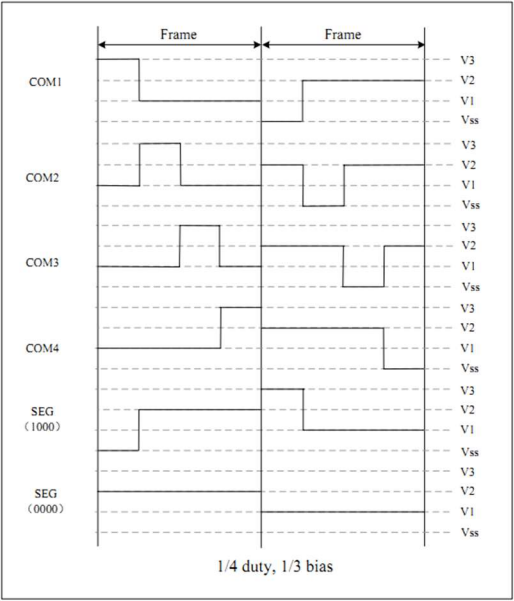


图 3-12 LCD 的 1/4duty 1/3bias 电源系统的时钟 (B 波形)

9.1 LCD 相关寄存器

0x0B	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LDCR0	RES_EN	Bias	SET_COM1	SET_COM0	FLCD1	FLCD0	LCD_CKS	DDisplay
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] RES_EN – 上拉电阻使能选择位(一般情况下请设置为 1)

0: 关闭所有上拉电阻功能

1: 开启上拉电阻功能(上拉电阻的选择还需要其他的控制开关)

BIT[6] LCDBias – LCD 驱动偏压 Bias 选择位(一般情况下请设置为 0)

0: LCD偏压产生1/3偏压选择

1: LCD偏压产生1/4偏压选择

BIT[5:4] SET_COM[1:0] – LCD端口时序设置选择位

LCD的COM口个数时序选择, 注意, 这里还要配合P3X的COM端口的配置, 这里只是设置对应的时序也就是在一个完整的COM时序内分成几个等份:

SET_COM [1:0]	选择的 COM 数	备注
00	4 个 COM	SGE20-23 有效
01	5 个 COM	SGE20-22 有效
10	6 个 COM	SGE20-21 有效
11	8 个 COM	SGE20-23 无效

BIT[3:2] FLCD[1:0] – LCD 帧宽选择位, LCD 的时钟源分别是内置的低频振荡器或者是外置的 32768 的振荡器产生, 时钟源可以通过 LCD_CKS 来进行对应的选择, FLCD 可以用来选择对应的预分频

00: 选择LCD时钟的256分频(当选择4个COM时)或128分频(不是4个COM时)

01: 选择LCD时钟的192分频(当选择4个COM时)或96分频(不是4个COM时)

10: 选择LCD时钟的160分频(当选择4个COM时)或80分频(不是4个COM时)

11: 选择LCD时钟的128分频(当选择4个COM时)或64分频(不是4个COM时)

FLCD[1:0]	低频时钟(Lirc=36K)	外置 32.768K	用途
00	140Hz/281Hz	128Hz/256Hz	通常针对 4 个 COM 使用
01	375Hz	341Hz	通常针对 5 个 COM 使用
10	450Hz	409.6Hz	通常针对 6 个 COM 使用
11	562.5Hz	512Hz	通常针对 8 个 COM 使用

在设置了以上分频后, LCD的刷新率还由选择的COM个数有关, 比如, 选择4个COM, 同时选择 FLCD=00,则实际的刷新率时140/4=35Hz; 刷新率越慢, 则更省电;

BIT[1] LCD_CKS – LCD 时钟选择位(不要使用, 请设置为 0)

0: 内置36K作为LCD时钟

1: 外置低频振荡器32768作为LCD时钟

BIT[0] LCDDisplay – LCD 显示方式选择位(不要使用, 请设置为 0)

0: 不提高动态显示

1: 开启动态显示模式

LCD 的开启详细的请参考我们的例程，以下为大致功能解释

0x0C	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LCDCR1	LCDEN	COM_ENB	SEG_ENB	LCDVBEN	XOSCSPEED	XOSC_EN	Vbias[1]	Vbias[0]
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

BIT[7] LCDEN– LCD 电压选择位（LCD 的开启总开关）

0:LCD关闭

1:LCD开启

BIT[6:5] COM 或 SEG 使能，当没有开启使能时，COM 或 SEG 将无信号输出

00: COM 和 SEG 同时有效

01: COM 有效，SEG 无效

10: COM 无效，SEG 有效

11: COM 和 SEG 都无效

BIT[4] LCDVEN– LCD 单独的电压模块使能位

0:LCD的偏压产生电路关闭

1:LCD的偏压产生电路开启

BIT[3] XTOSC_SPEED– 外置 32768 晶振加速开启位

0:外置晶振起振功能开启，功耗略大

1:外置晶振功能关闭

BIT[2] XTOSC_EN– 外置 32768 晶振使能位(不要使用，请设置为 0)

0:外置的32768晶振辅助振荡电路关闭

1:外置的32768晶振振荡电路开启，需要外接32768晶振

BIT[1:0] Vbias [1:0]– LCD电压偏置功耗选择位

Vbias[1:0]	加速开启比例
00	1/32
01	1/16
10	1/8
11	1/4

加速偏压选择将需要更大的功耗，其中设置 Vbias=11 需要的功耗最大；

注：电阻型 LCD 电阻值偏小，可以得到较好的显示效果，但电流相对会大一些，不适合低功耗的应用。而电阻值选择 3M，虽然可以达到较低的功耗，但 LCD 显示效果会变得差一些。

所以提供了兼顾低功耗和显示效果的显示模式：快速充电模式（即并接加速电阻）。在显示数据刷新时刻选择加速电阻，提供较大的驱动电流，在数据保持期间选择大的偏置电阻，提供较小的驱动电流。

快速充电时间可以通过 LCDSP 位配置为 LCD COM 周期的 1/4、1/5、1/6 或者 1/8

注：快速开启状态为 IO 口输出为 CMOS 电平，其中低电平驱动 30mA，高电平驱动 10mA。当输出状态变化时，上升或者下降沿口为 ns 级变化。由于上升下降沿变化太快，对电源干扰大，但无静态功耗。

慢速开启状态为 IO 口输出为 NMOS 开漏电平，其中低电平驱动 30mA，高电平为内部 40K 上拉电阻。当输出状态变化时，上升或者下降沿口为 1us 级变化。由于上升下降沿缓变，对电源干扰小，但有 15uA 静态功耗，进入静态时需要将其设置为快速开启状态。

9.2 LCD 数据寄存器

数据存储中有一部分区域是专门为 LCD 的显示数据而保留，即显示存储区 LCDRAM。

单片机内部显示驱动电路会自动读取任何写入此处的数据并据此产生 LCD 驱动信号。因此任何写入 LCD 数据寄存器的数据，会立即映射 LCD 端口上。

0x40-0x57	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
<i>LCDDSn</i>	COM7 SEGn	COM6 SEGn	COM5 SEGn	COM4 SEGn	COM3 SEGn	COM2 SEGn	COM1 SEGn	COM0 SEGn
<i>R/W</i>	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
<i>初始值</i>	0	0	0	0	0	0	0	0

BIT[7:0] COMmSEGn – LCD 数据寄存器(m=7-0,n=23-0)

当不使用 SEG20—SEG23 时，需要把 SEG20—SEG23 对应的显示数据设置为 00，这样可以节省显示功耗；

更形象的关于本芯片的显示存储器的排布如下图

管脚	P11	P12	P13	P14	P00	P01	P02	P03	P04	P05	P06	P07	P20	P21	P22	P23	P24	P25	P26	P27	P30	P31	P32	P33
SEG 编号	Seg0	Seg1	Seg2	Seg3	Seg4	Seg5	Seg6	Seg7	Seg8	Seg9	Seg10	Seg11	Seg12	Seg13	Seg14	Seg15	Seg16	Seg17	Seg18	Seg19	Seg20	Seg21	Seg22	Seg23
地址	0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F	0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57
COM7	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
COM6	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
COM5	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
COM4	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
COM3	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
COM2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
COM1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
COM0	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

上图中，由于 COM 和 SEG 互用，所以图中阴影部分的显示存储器是不可用的；

图中设置对应点点亮，则对应在该显示存储器对应位写 1，如果不需要点亮，则写 0；

10 中断

WL8032 的中断有外中断（PBINT）、定时器中断（T0）。外部中断、定时器中断可被 CPU 状态寄存器 MCR 的 GIE 位屏蔽。

中断响应过程如下：

- ✧ 当发生中断请求时，CPU 将相关下一条要执行的指令的地址压栈保存（累加器 A 和状态寄存器需要软件保护），对中断屏蔽位 GIE 清 0，禁止中断响应。与复位不同，硬件中断不停止当前指令的执行，而是暂时挂起中断直到当前指令执行完成。
- ✧ CPU 执行中断时，程序跳到中断向量 0008H 地址开始执行中断代码，中断代码应该先保存累加器 A 和状态寄存器，然后判断是哪一个中断响应。
- ✧ 执行中断内容后应该恢复累加器 A 和状态寄存器，然后执行 RETIE 返回主程序。这时，从堆栈取出 PC 的值，然后从中断发生时的那条指令的后一条指令继续执行。

WL8032 的中断向量地址是 0008H。

10.1 外中断

WL8032 有 1 路外部中断源，当外部中断触发时，外部中断标志（PBIF）将被置 1，若中断总使能位 GIE 为 1 且外部中断使能位（PBIE）为 1，则产生外部中断。

10.2 定时器中断

定时器 T0、T1 在计数溢出时会置位中断标志 T0IF 和 T1IF，若中断总使能位 GIE 为 1 且定时器中断使能位（T0IE 和 T1IE）为 1，则产生定时器中断。

10.3 LVD 标记

当 VDD 电压低于设置电压后会置位标志 LVD，LVD 不会产生中断，则仅可以读取标志。

10.4 中断相关寄存器

中断使能寄存器(地址0x05)

地址 0x05	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTE	GIE	-	-	-	T1IE	-	PB1E	T0IE
R/W	R/W	-	-	-	R/W	-	R/W	R/W
初始值	0	-	-	-	0	-	0	0

BIT[7] GIE – 总中断使能

0:屏蔽所有中断

1:中断源是否产生中断有相应的控制位决定

BIT[6:4] 未用

BIT[3] T1IE – Timer1 计数器溢出中断使能寄存器

0:关闭T1计数器中断

1:开启T1计数器中断

BIT[1] PBIE – 键盘扫描中断控制寄存器

0:关闭键盘扫描中断

1:开启键盘扫描中断

BIT[0] T0IE – Timer0 计数器溢出中断使能寄存器

0:关闭T0计数器中断

1:开启T0计数器中断

中断标志寄存器

0x06	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTF	-	-	-	-	T1IF	-	PBIF	T0IF
R/W	-	-	-	-	R/W	-	R/W	R/W
初始值	-	-	-	-	0	-	0	0

BIT[7:4] 未用

BIT[3] T1IF – 定时器 1 中断标志

0:未发生定时器 1 中断

1:发生定时器 1 中断，需软件清零

BIT[2] 未用

BIT[1] PBIF– 键盘扫描中断标志

0:未发生键盘扫描中断

1:发生键盘扫描中断，需软件清零

BIT[0] T0IF – 定时器 0 中断标志

0:未发生定时器 0 中断

1:发生定时器 0 中断，需软件清零

11 电气参数

11.1 极限参数

参数	符号	值	单位
工作电压	VDD	-0.3~4.0	V
输入电压	VIN	VSS-0.3 ~ VDD+0.3	V
工作温度	TA	-20 ~ 70	°C
储存温度	Tstg	-65 ~ 150	°C
焊接温度及时间	Tsld	260°C持续 10 秒	
流过 VDD 最大电流	IVDDmax	30	mA
流过 GND 最大电流 1 (除 IROUT, P02 外)	IGNDmax1	30	mA
流过 GND 最大电流 2 (IROUT 引脚)	IGNDmax2	持续驱动 200	mA
流过 GND 最大电流 3 (IROUT 引脚)	IGNDmax3	间歇性驱动 500 (大于 5KHz, 占空比 50%)	mA
流过 GND 最大电流 4 (P02 引脚)	IGNDmax4	60	mA

11.2 直流特性参数

所有参数均为 T=25°C, VDD=3V, 除非特殊说明

特性	符号	引脚	条件	最小	典型	最大	单位
工作电压	VDD	VDD	FCPU=0~8MHz	2.2	3	5.5	V
			FCPU=0~4MHz	2.0	3	5.5	V
输入高电平	VIH	所有输入引脚		0.8VDD			V
输入低电平	VIL	所有输入引脚				0.2VDD	V
上拉电阻 1	RPU1	P0 P1	Vin=0.9V, RSEL=0	50	100	200	KΩ
			Vin=0.9V, RSEL=1	20	40	80	
输出高电平驱动电流 1	IOH1	P0 P1(除 P02)	Voh=2.4V	-4	-8	-16	mA
输出低电平驱动电流 1	IOL1	P0 P1 (除 P02)	Vol=0.6V	4	8	16	mA
输出低电平驱动电流 2	IOL2	P02	Vol=0.3V	15	30	60	mA
输出低电平驱动电流 3	IOL3	IROUT	Vol=1.5V,DSEL=00	75	125	175	mA
			Vol=1.5V, DSEL=01	150	250	350	
			Vol=1.5V, DSEL=10	225	375	525	
			Vol=1.5V,DSEL=11	300	500	700	
动态功耗	IDDC	VDD	FCPU=16MHz, VDD=3V		600		uA

			FCPU=内部 LIRC36KHz/2, 打开低功耗		6	12	uA
			FCPU=外部 32768 晶体/2, 打开低功耗		6	12	uA
HOLD 模式 1 功耗	IHOLD1	VDD	高频振荡开启, 执行 STOP 指令 内部高频 16MHz		550		uA
HOLD 模式 2 功耗 1	IHOLD21	VDD	高频振荡关闭, 内部低频振荡开启, LCD 关闭, LVD 关闭, 执行 STOP 指令		2	3	uA
HOLD 模式 2 功耗 2	IHOLD22	VDD	高频振荡关闭, 外部低频振荡开启, LCD 关闭, LVD 关闭, 执行 STOP 指令		3	6	uA
休眠模式功耗	ISTOP	VDD	高频振荡器关闭, 低频振荡器关闭, LCD 关闭, LVD 关闭, 执行 STOP 指令		1.5	3	uA
LCD 模式 1	ILCD1	VDD	高频振荡关闭, 低频振荡开启, LCD 开 启 300K 电阻, LVD 关闭, 执行 STOP 指令		13	26	uA
LCD 模式 2	ILCD2	VDD	高频振荡关闭, 低频振荡开启, LCD 开 启 3M 加速电阻 300K, LVD 关闭, 执行 STOP 指令		6	12	uA
晶体输入对地电容	CG	OSCI			10		pF
晶体输出对地电容	CD	OSCO			10		pF
32768 晶体最低起振 电压	Vsu	VDD		1.8			V
低电压复位电压	VLVR	VDD	VLVRS=11	2.6	2.4	2.2	V
			VLVRS=10	2.4	2.2	2.0	V
			VLVRS=01	2.2	2.0	1.8	V
上电复位电压	VPOR	VDD	VLVRS=00	2.0	1.8	1.6	V
低电压检测电压	VLVD	VDD	LVDSEL=0000	2.0	1.8	1.6	V
			LVDSEL=0001	2.0	1.8	1.6	V
			LVDSEL=0010	2.1	1.9	1.7	V
			LVDSEL=0011	2.2	2.0	1.8	V
			LVDSEL=0100	2.3	2.1	1.9	V
			LVDSEL=0101	2.4	2.2	2.0	V
			LVDSEL=0110	2.5	2.3	2.1	V
			LVDSEL=0111	2.6	2.4	2.2	V
			LVDSEL=1000	2.7	2.5	2.3	V
			LVDSEL=1001	2.8	2.6	2.4	V
			LVDSEL=1010	2.9	2.7	2.5	V
			LVDSEL=1011	3.0	2.8	2.6	V
			LVDSEL=1100	3.1	2.9	2.7	V
			LVDSEL=1101	3.2	3.0	2.8	V
			LVDSEL=1110	3.3	3.1	2.9	V
			LVDSEL=1111	3.4	3.2	3.0	V

11.3 交流电气参数

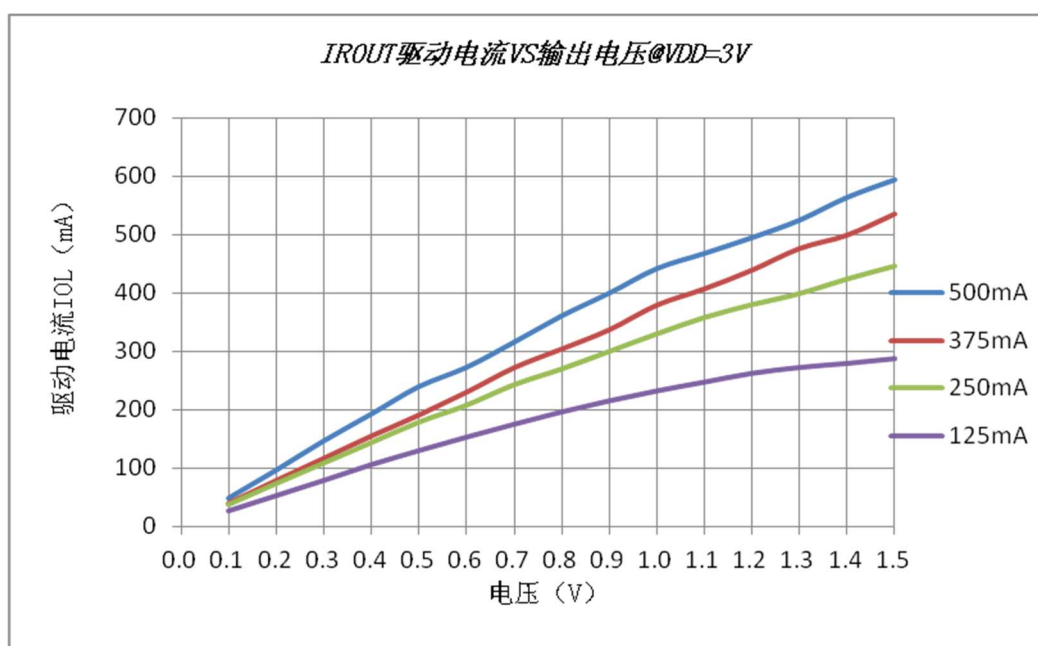
T=25℃

特性	符号	条件	最小	典型	最大	单位
内部高频 RC16M 振荡频率	FHIRC1	T=25℃ VDD=3V	-1%	16	+1%	MHz
		T=-20℃~70℃ VDD=2.0~5.5V	-2%	16	+2%	MHz
内部低频 RC 振荡频率	FLIRC	T=25℃ VDD=3V	-50%	36	+50%	KHz
32768 晶体起振时间	Tsu	T=25℃ VDD=3V			1	S

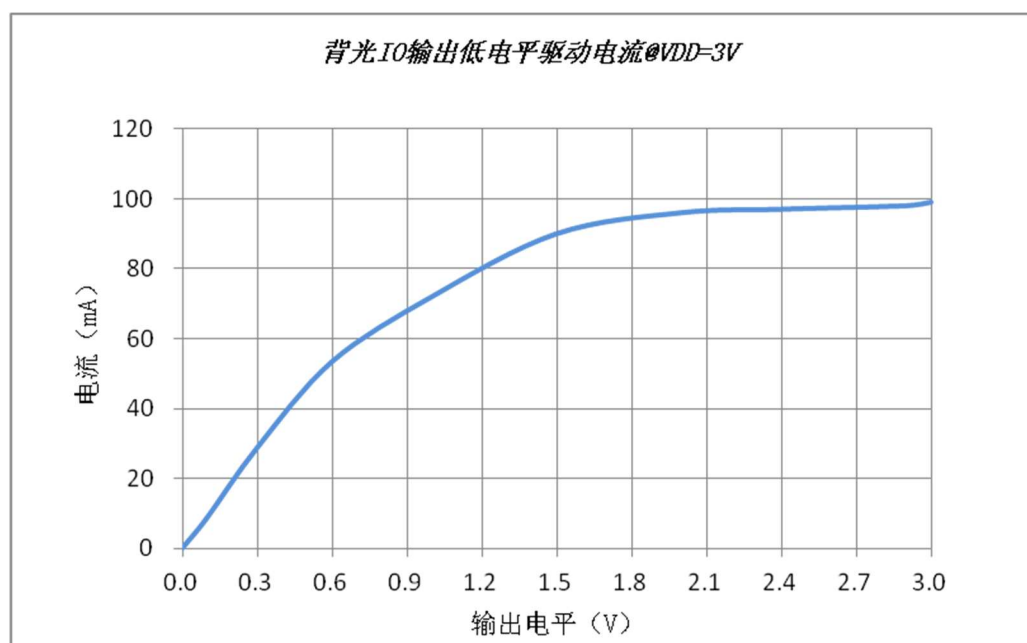
12 特性曲线图

注：本节列出的特性曲线图仅作为设计参考，部分数据可能超出芯片额定的工作条件范围，为保证芯片能正常工作，请严格按照电气特性说明。

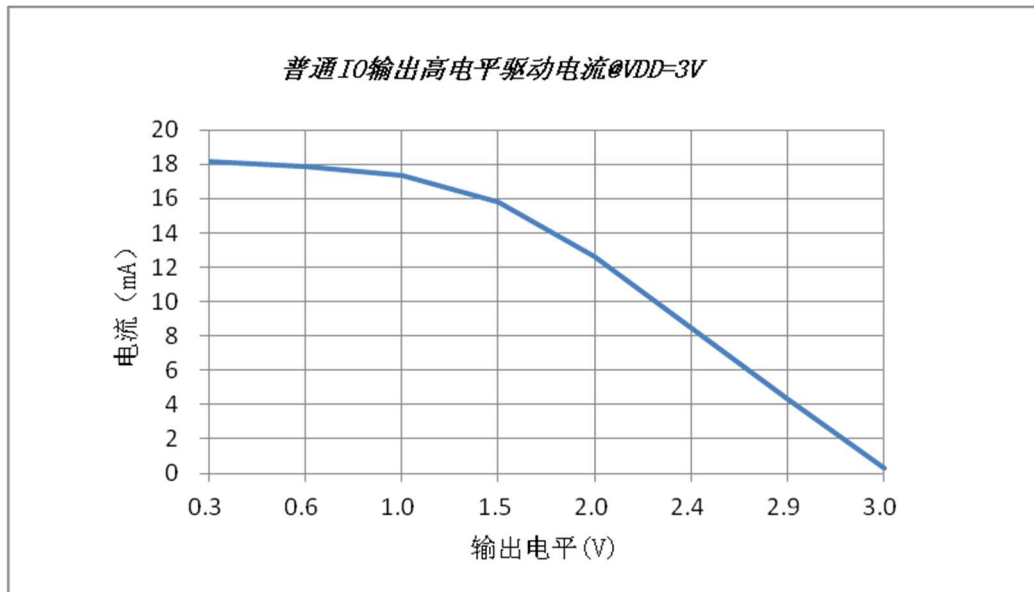
12.1 IROUT 驱动电流 VS 输出电压



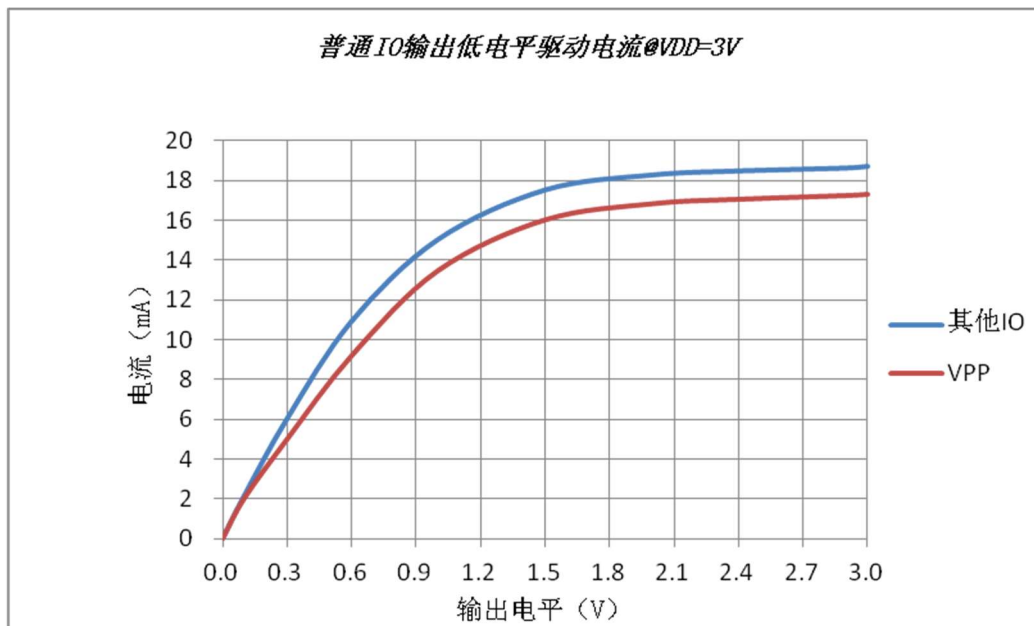
12.2 LCD 背光驱动电流 VS 输出电压



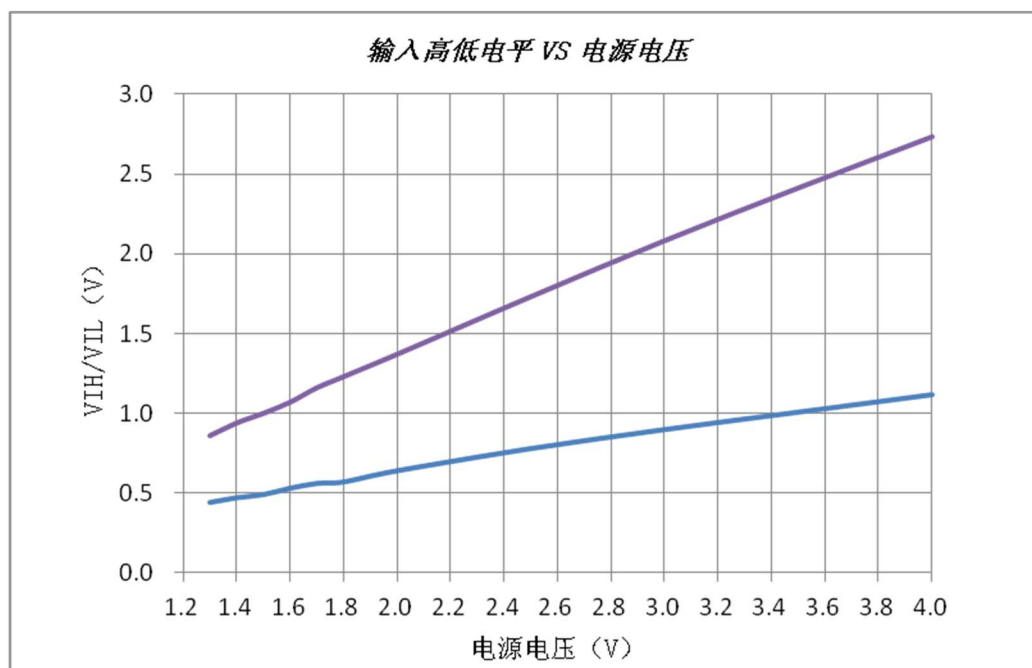
12.3 IO 输出高电平驱动电流 VS 输出电平



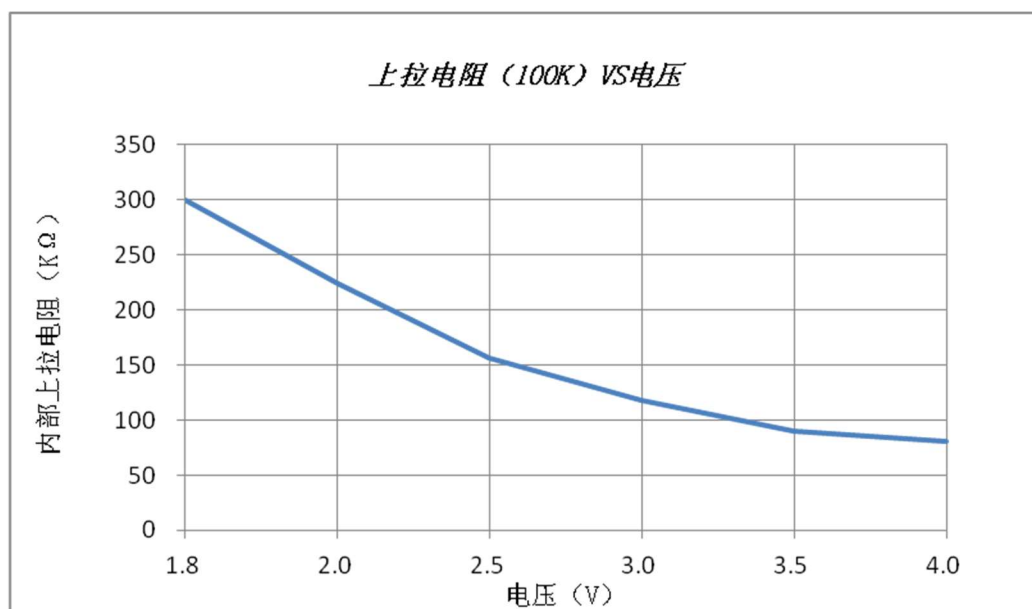
12.4 IO 输出低电平驱动电流 VS 输出电平

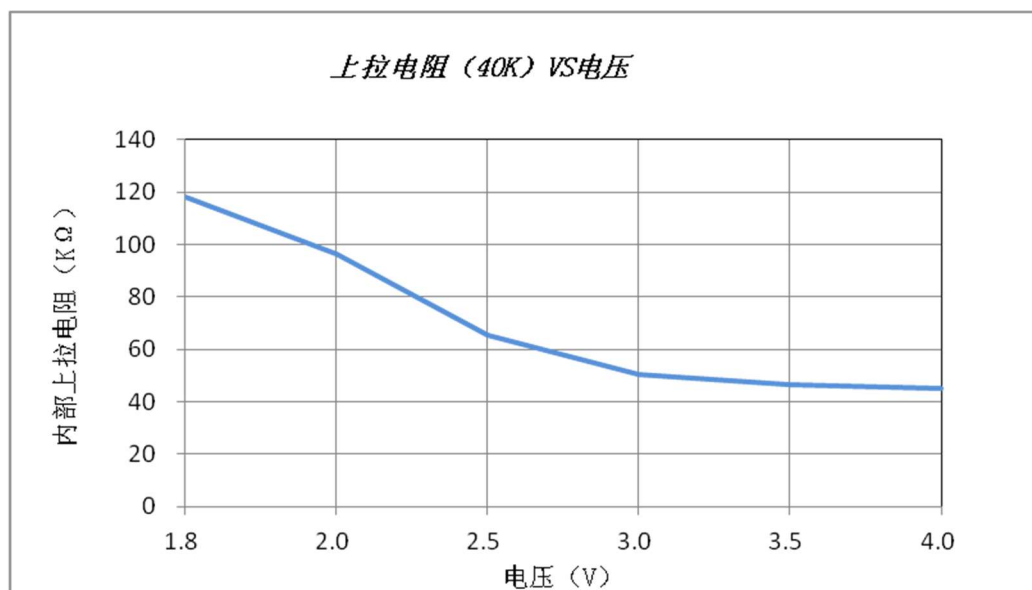


12.5 输入高低电平 VS 电源电压

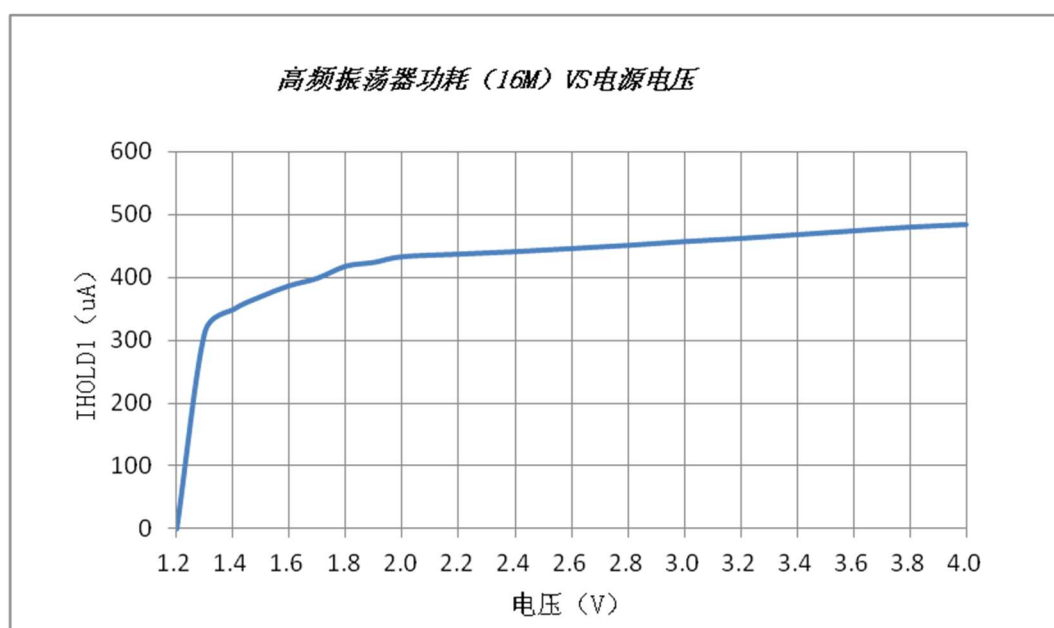


12.6 上拉电阻 VS 电源电压

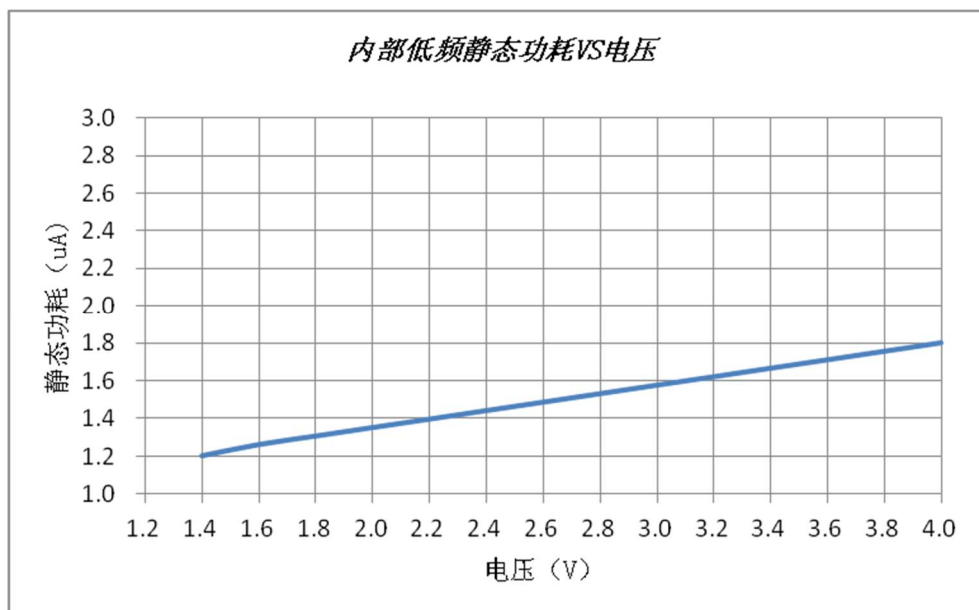




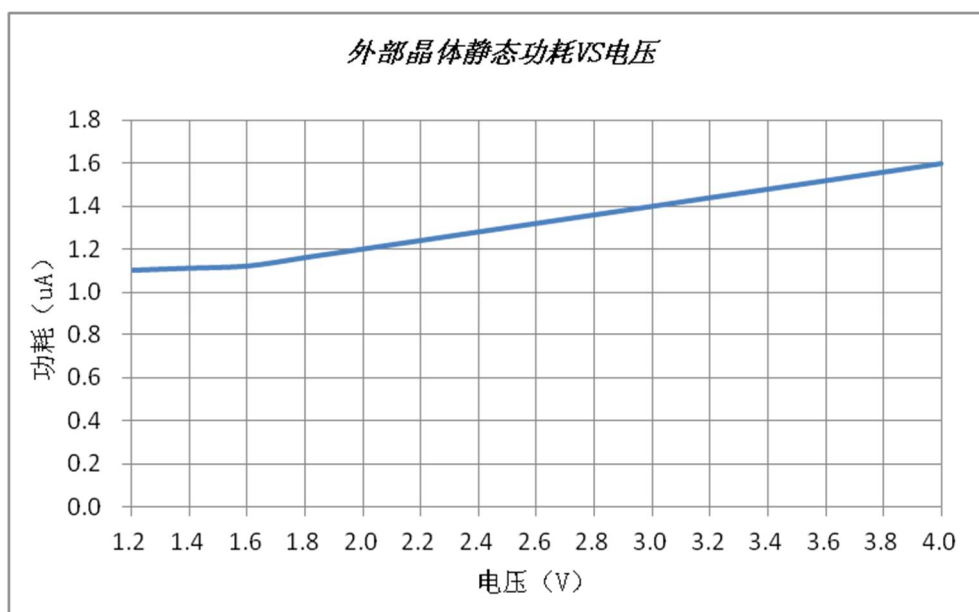
12.7 HOLD 模式1 功耗 VS 电源电压



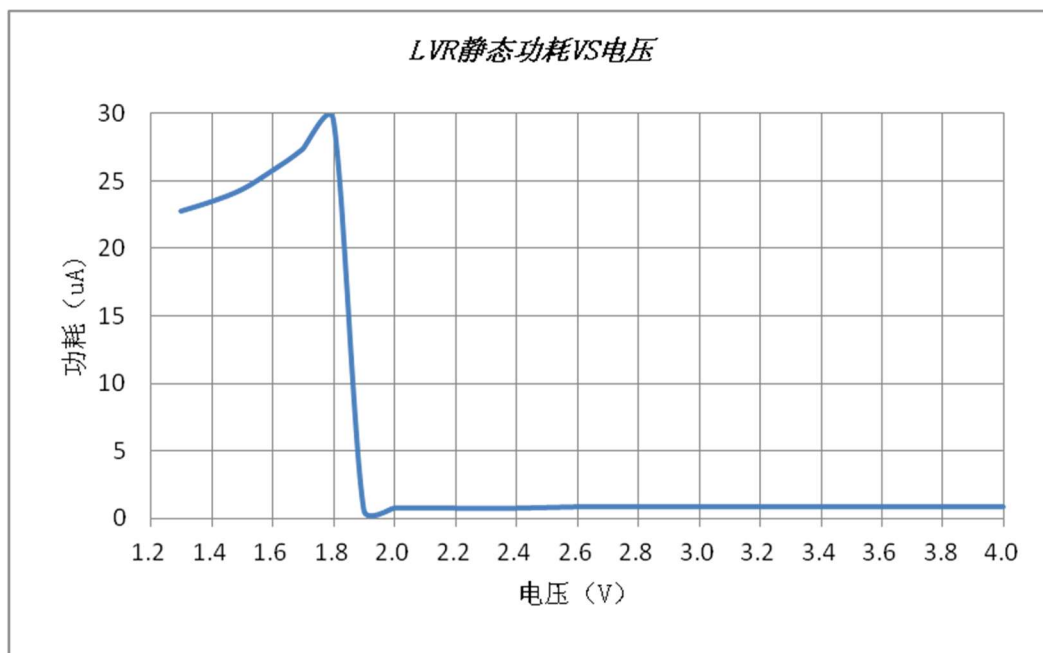
12.8 HOLD 模式 2（内部低频）功耗 VS 电源电压



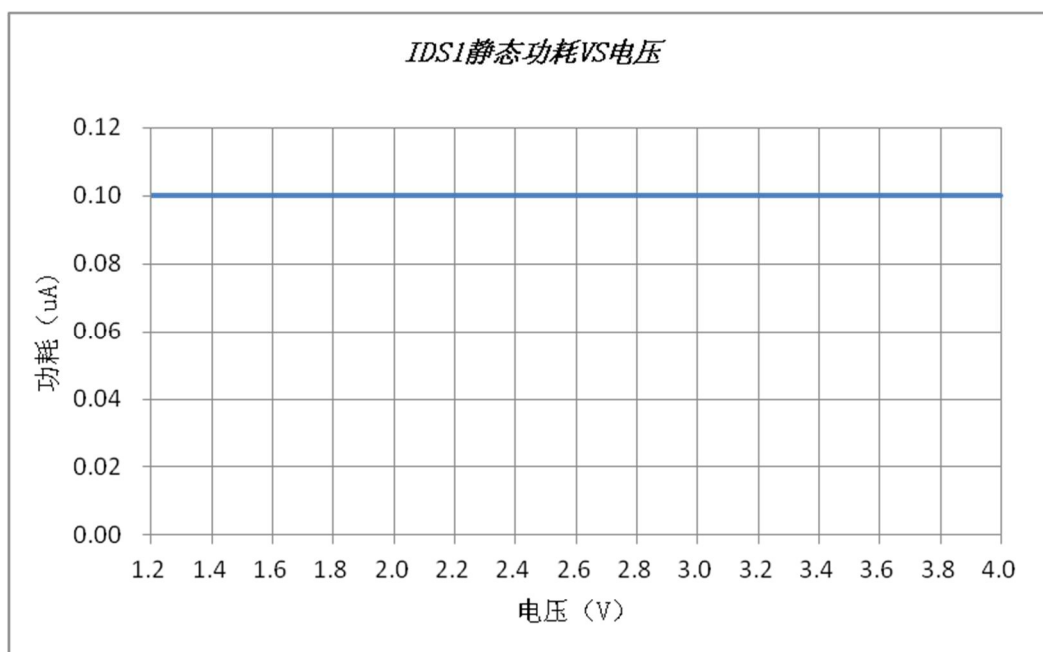
12.9 HOLD 模式 2（外部低频）功耗 VS 电源电压



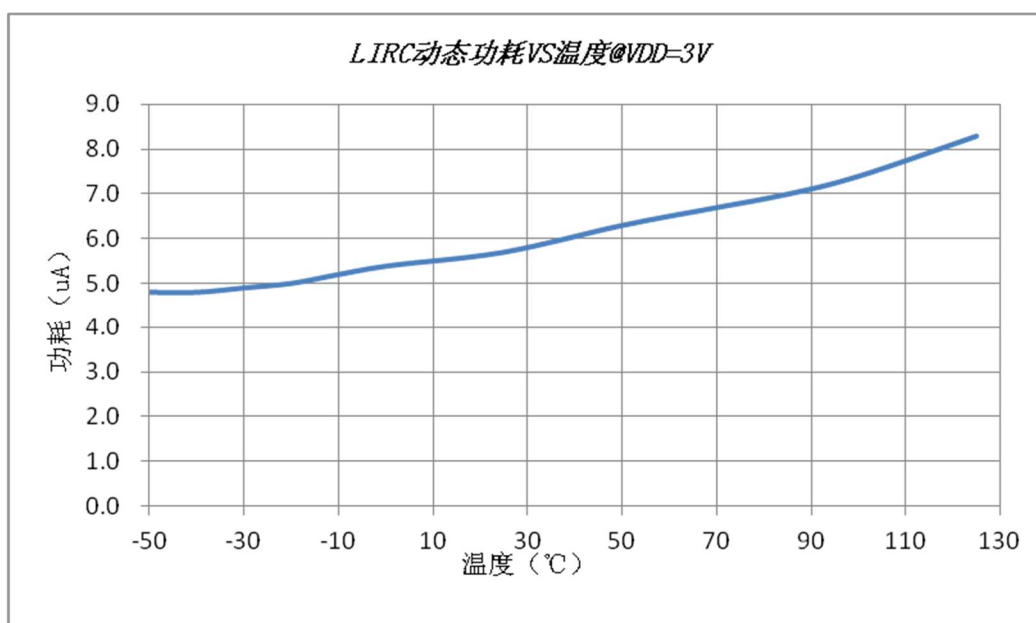
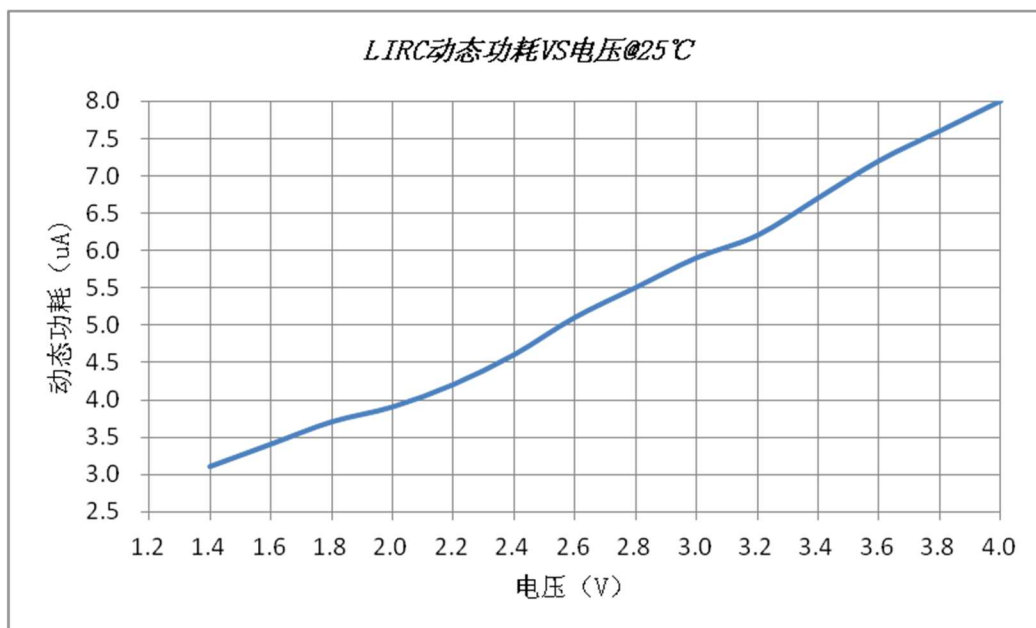
12.10 休眠模式功耗 (LVR=1.8V) VS 电源电压



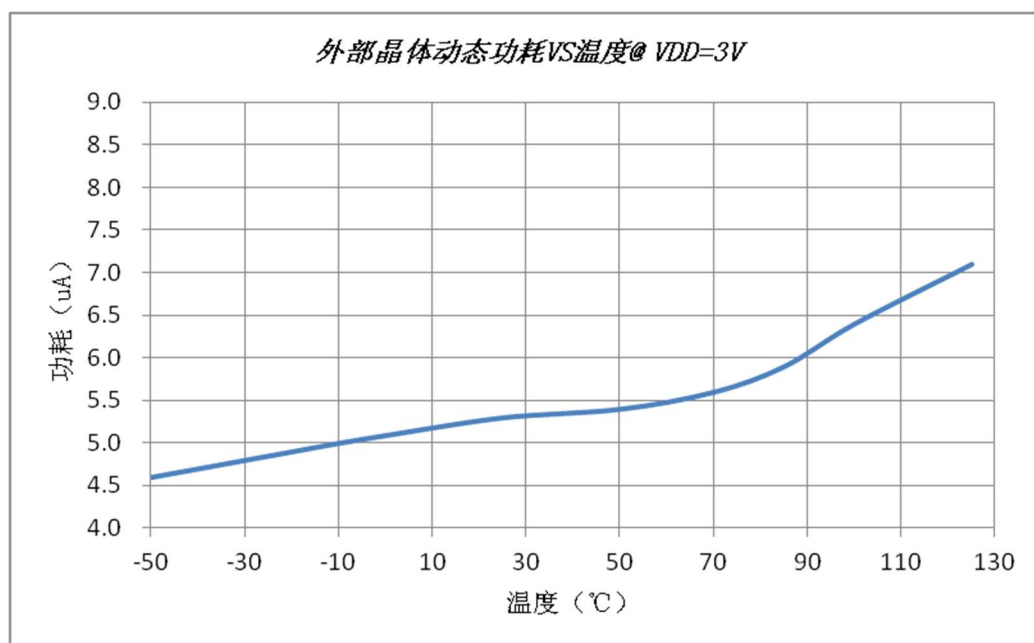
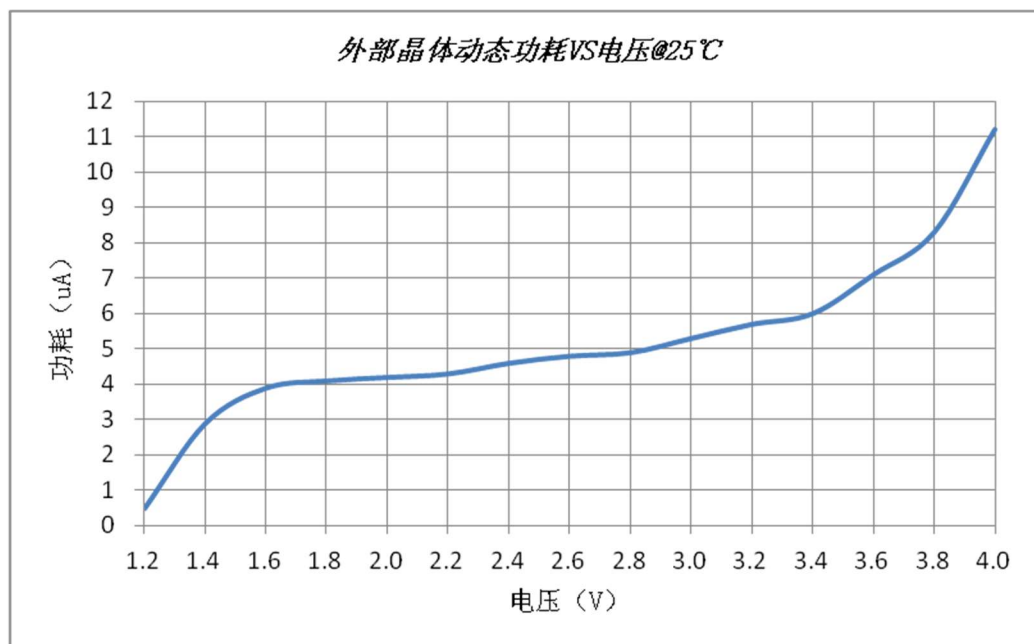
12.11 休眠模式功耗 (LVR=1.2V) VS 电源电压



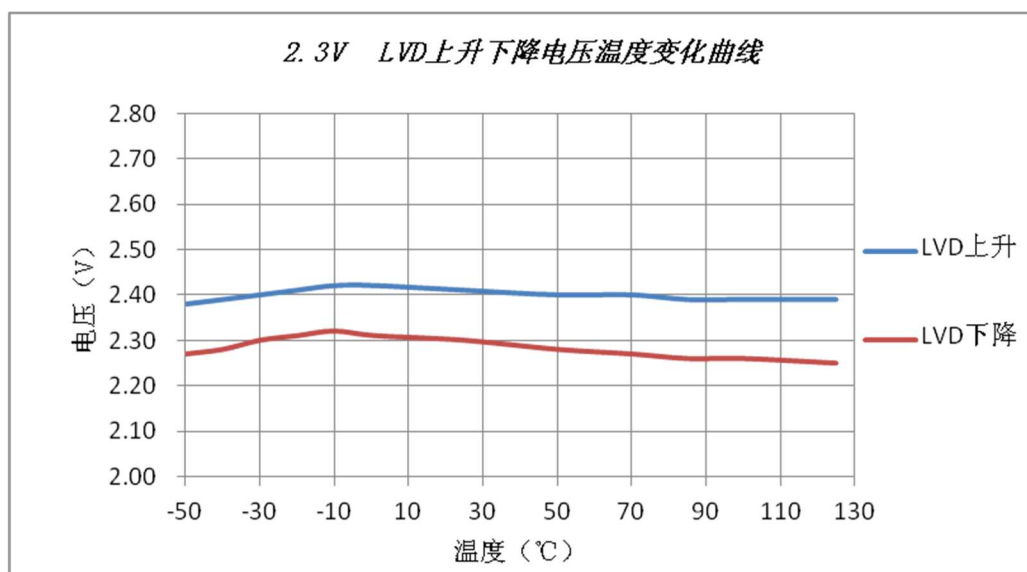
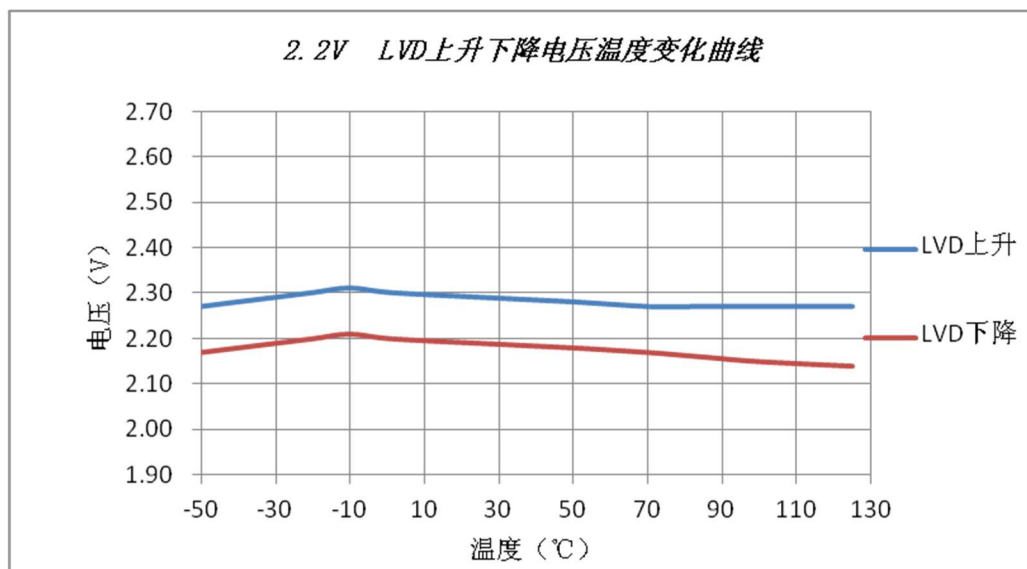
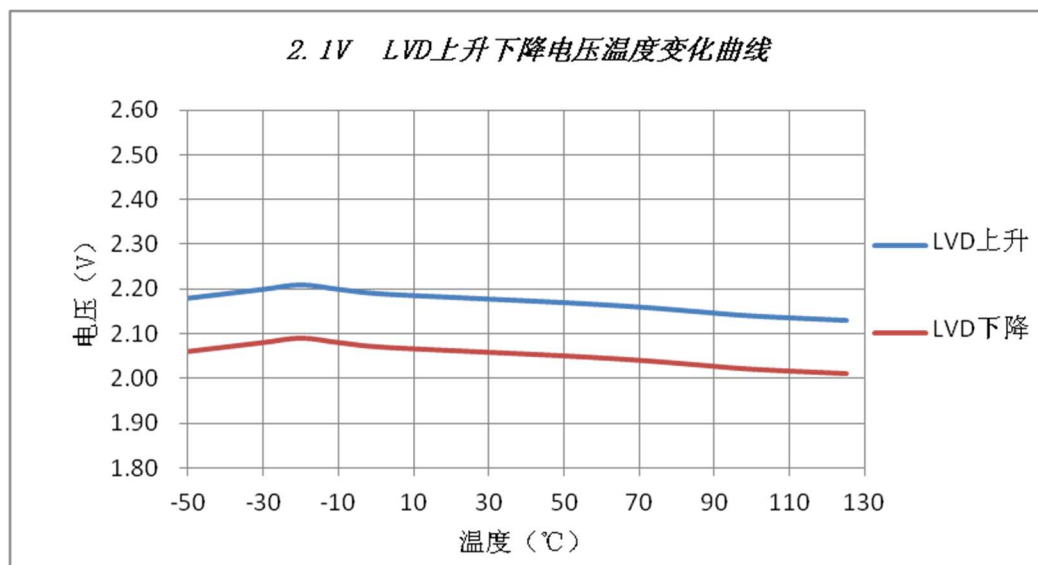
12.12 LIRC (内部低频) 动态功耗 VS 电源电压 VS 温度



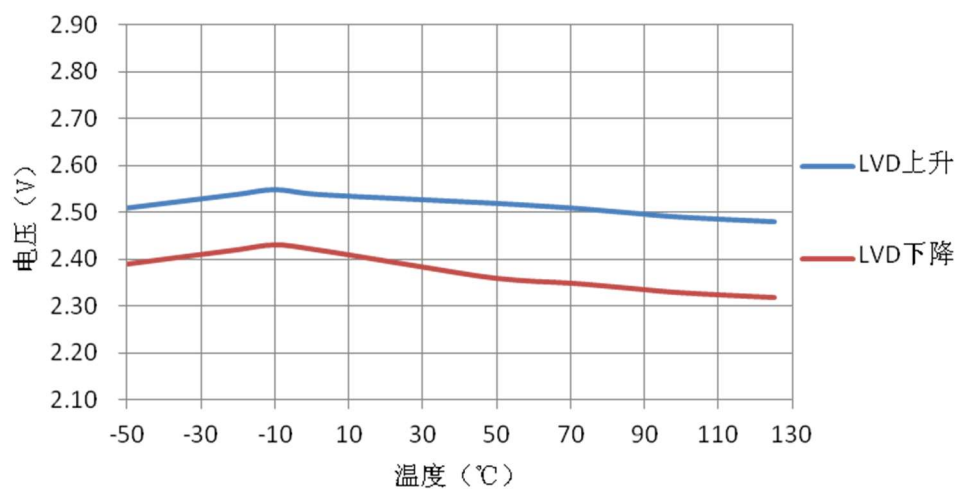
12.13 外部低频动态功耗 VS 电源电压 VS 温度



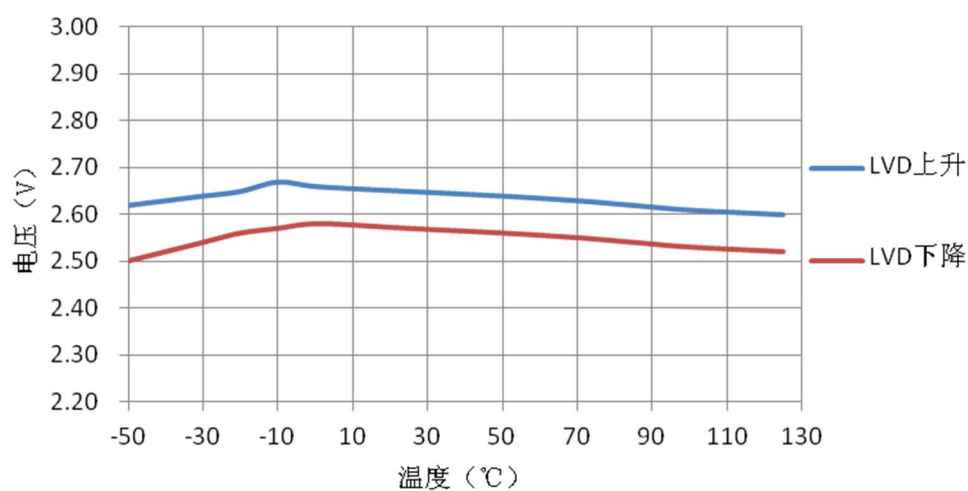
12.14 LVD 电压 VS 温度



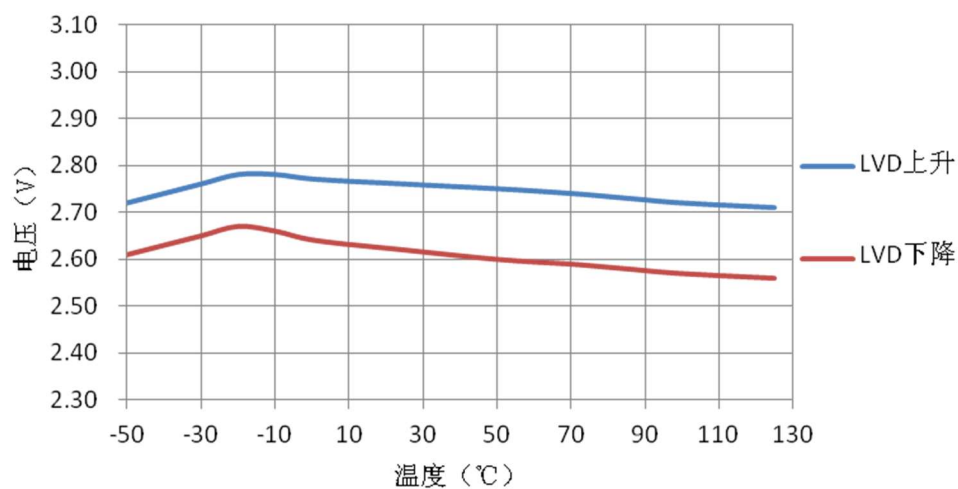
2. 4V LVD上升下降电压温度变化曲线



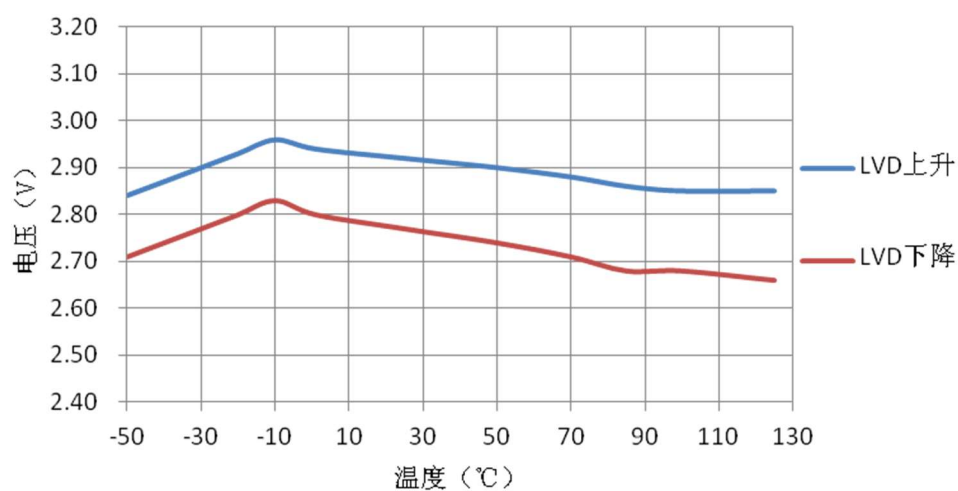
2. 5V LVD上升下降电压温度变化曲线



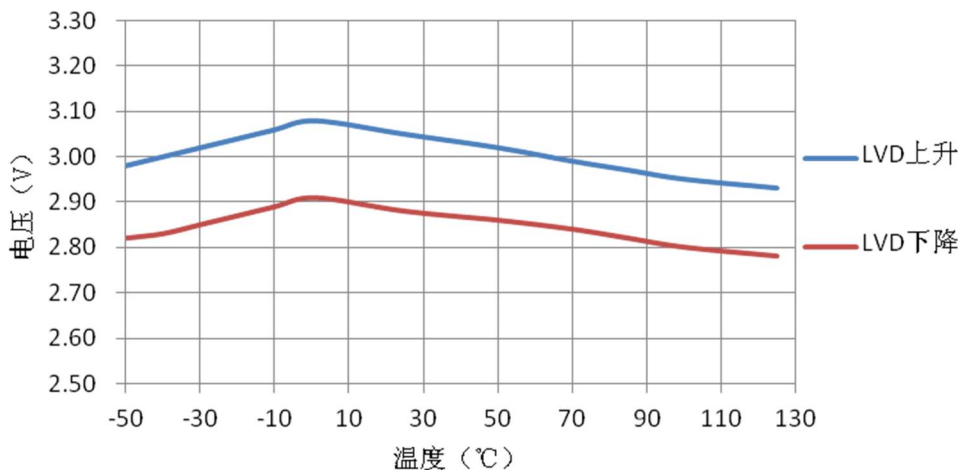
2. 6V LVD上升下降电压温度变化曲线



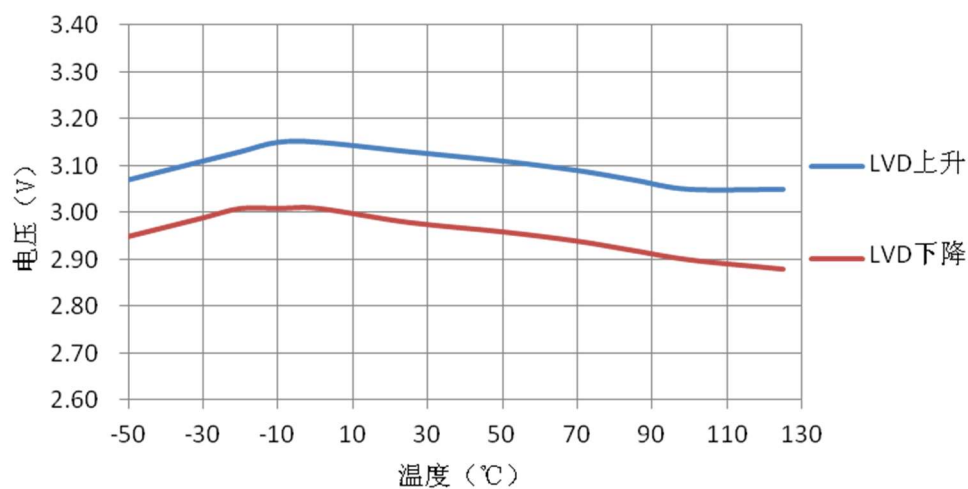
2. 7V LVD上升下降电压温度变化曲线



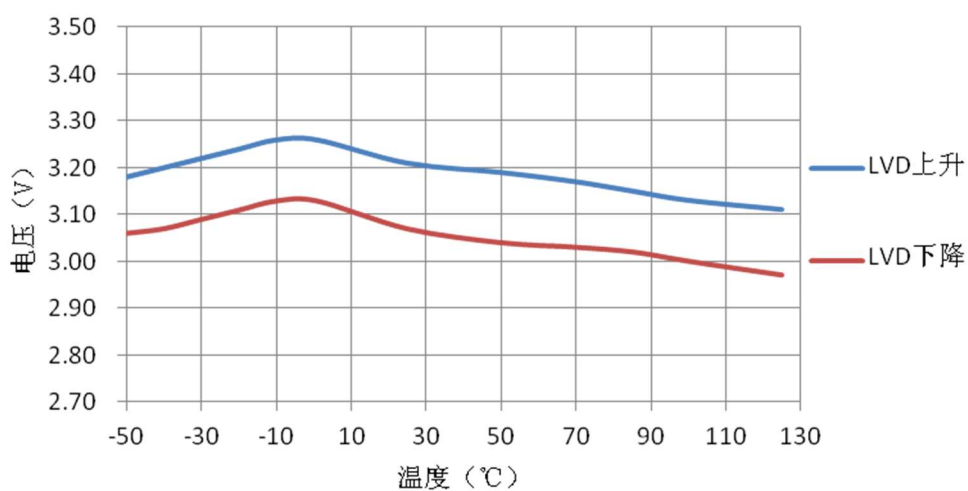
2. 8V LVD上升下降电压温度变化曲线



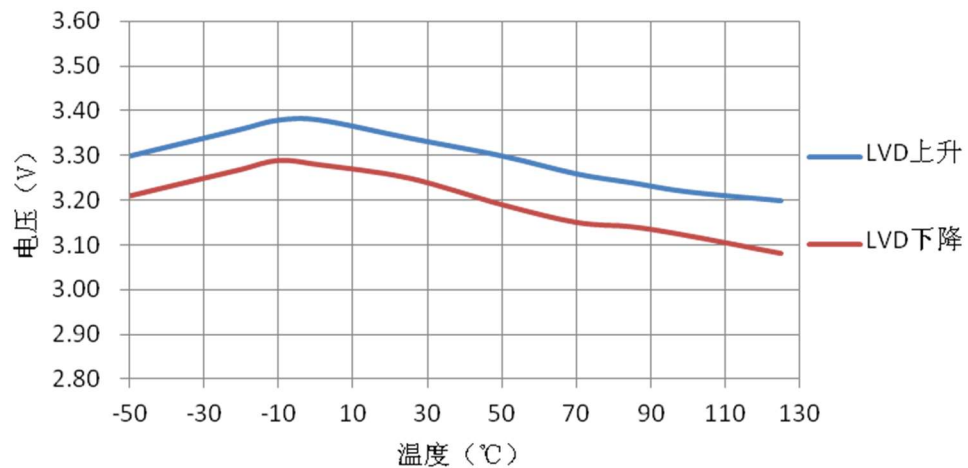
2. 9V LVD上升下降电压温度变化曲线

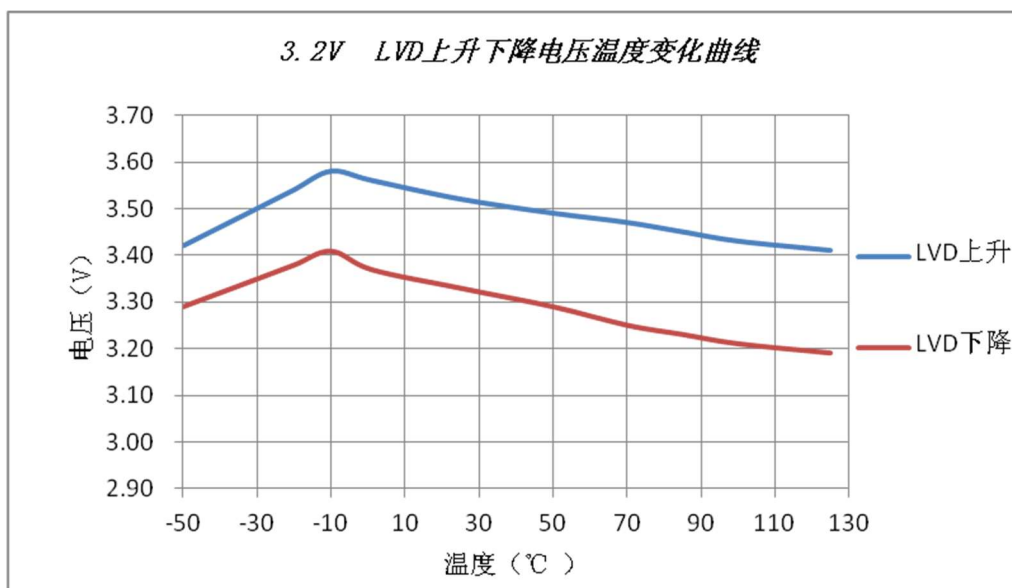


3. 0V LVD上升下降电压温度变化曲线

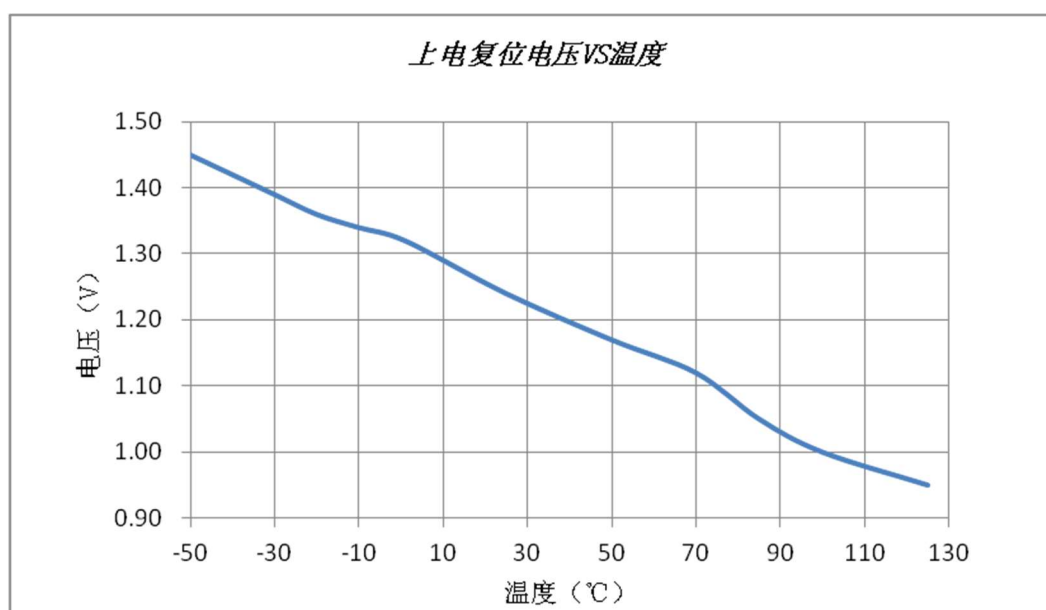


3. 1V LVD上升下降电压温度变化曲线

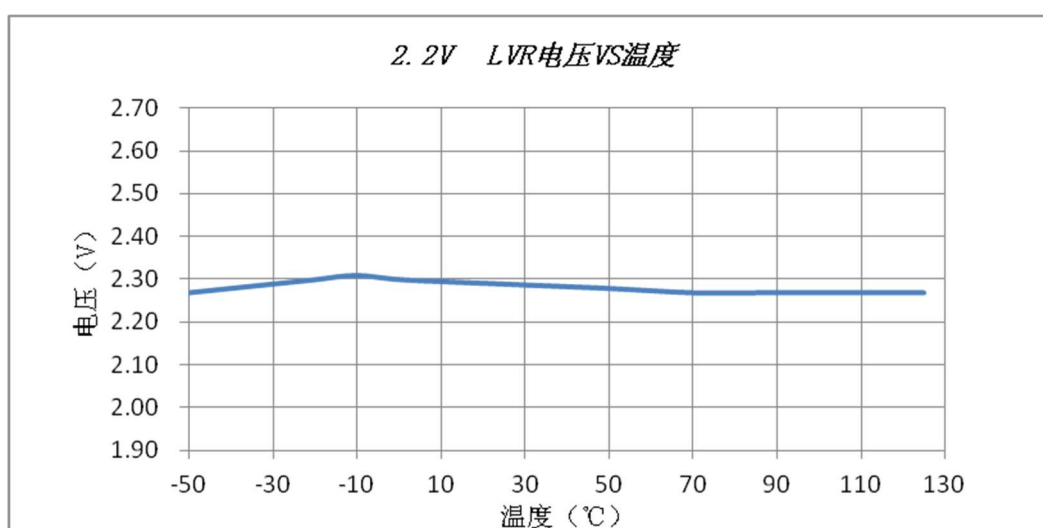
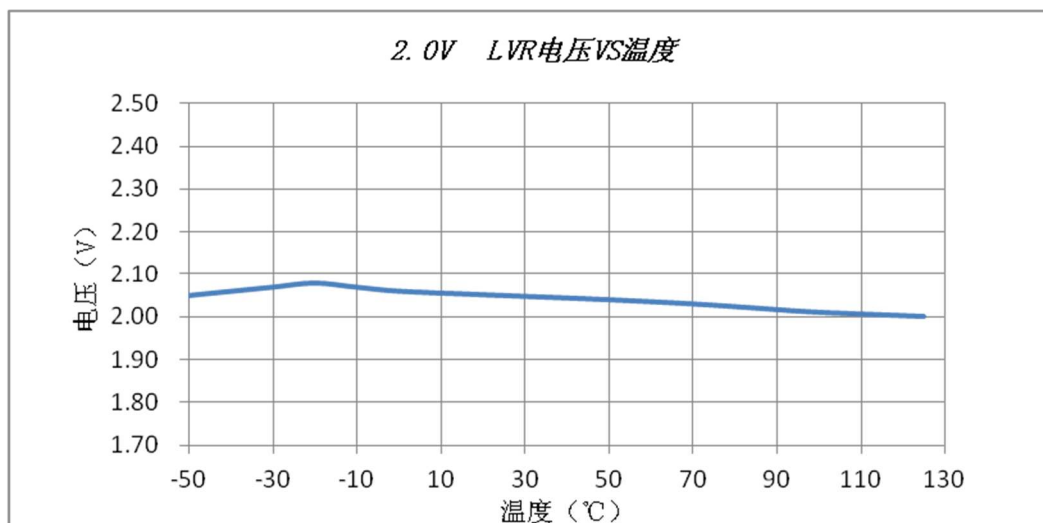
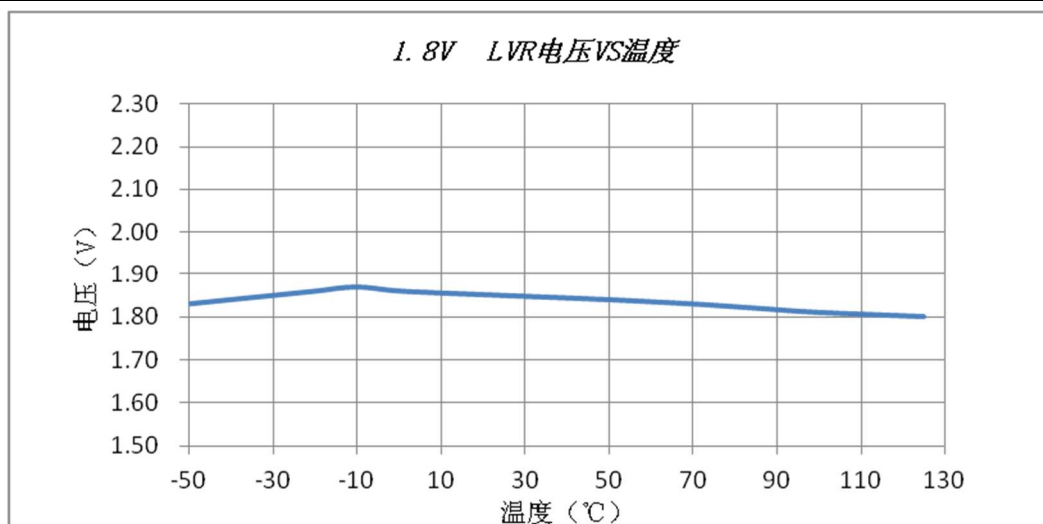


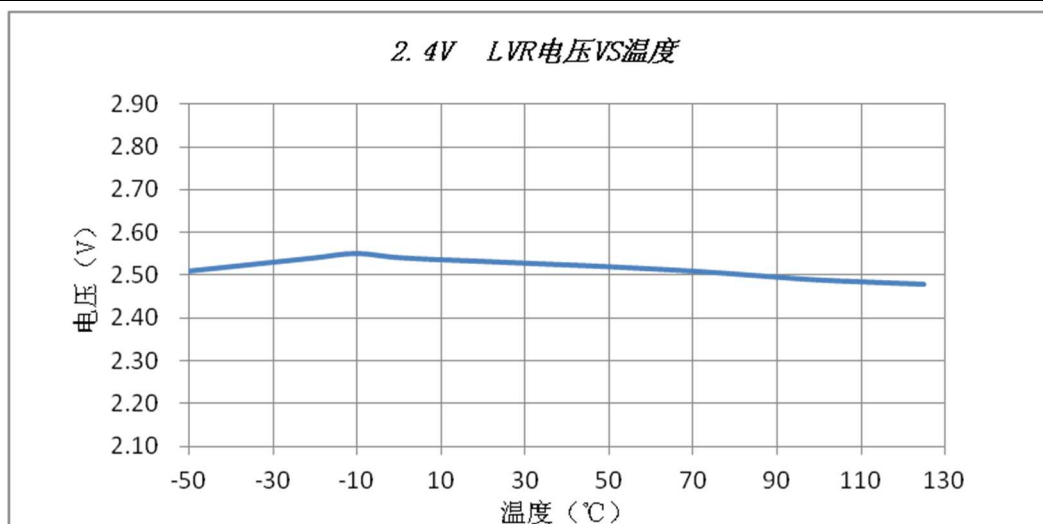


12.15 上电复位电压 VS 温度

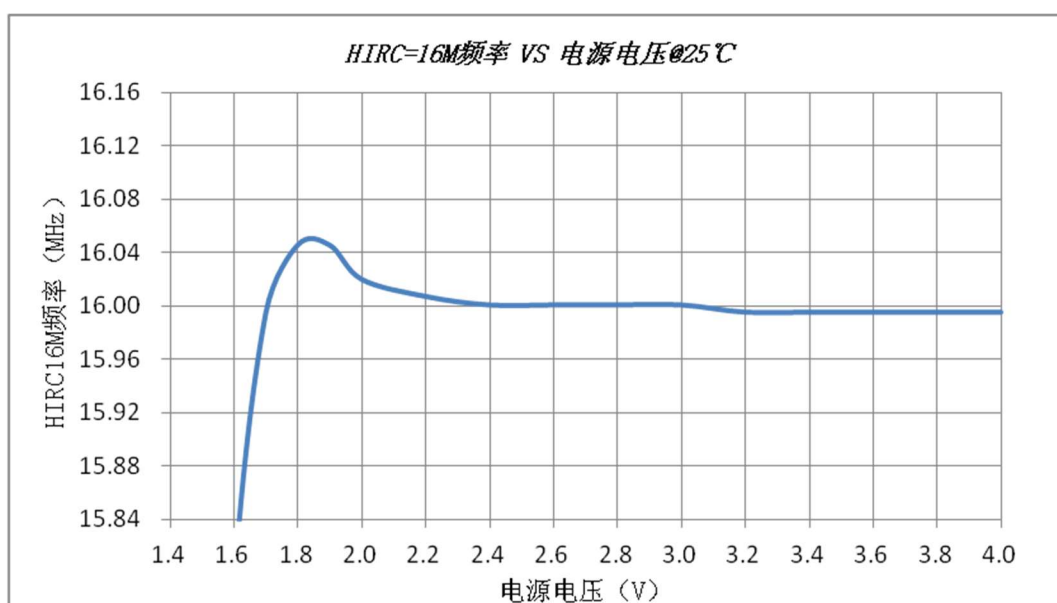


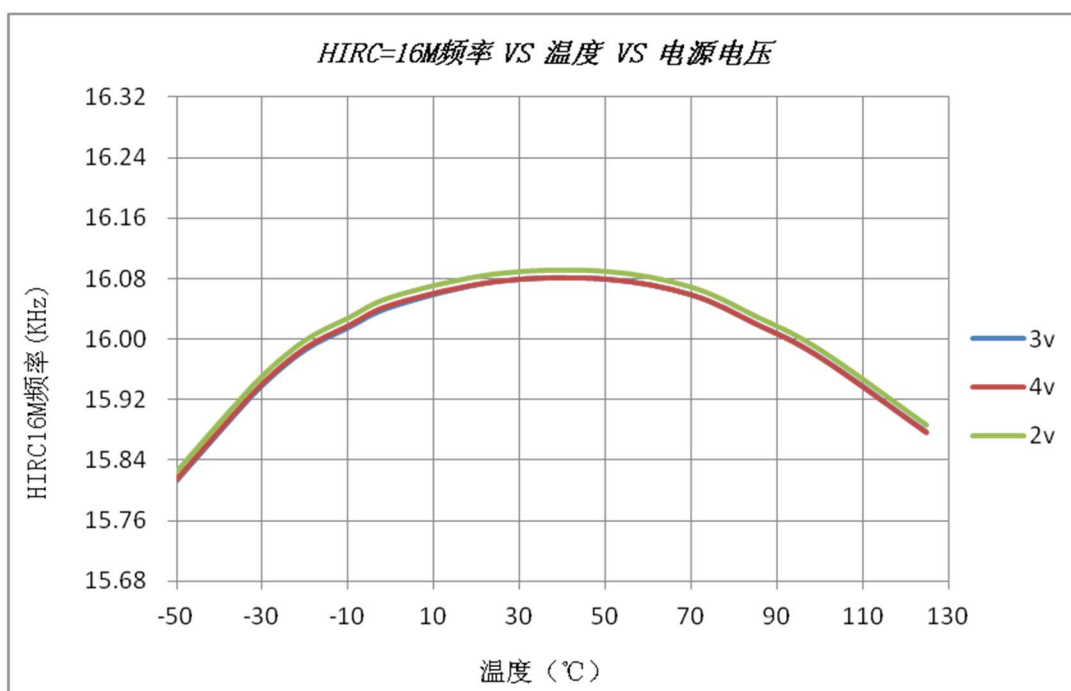
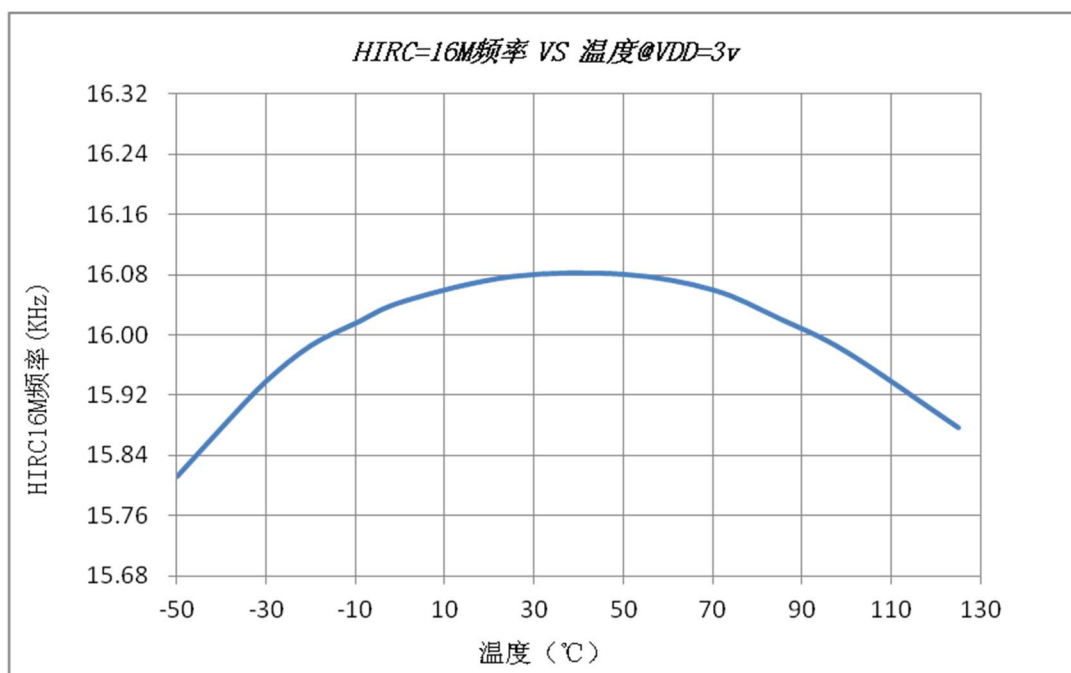
12.16 LVR 电压 VS 温度



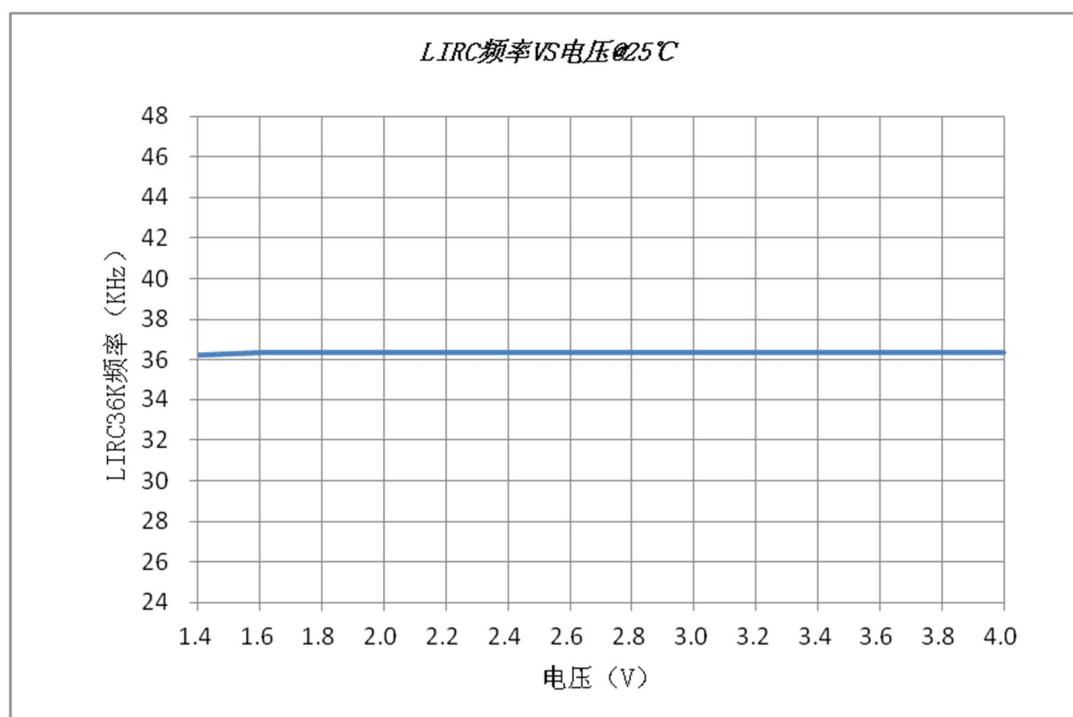
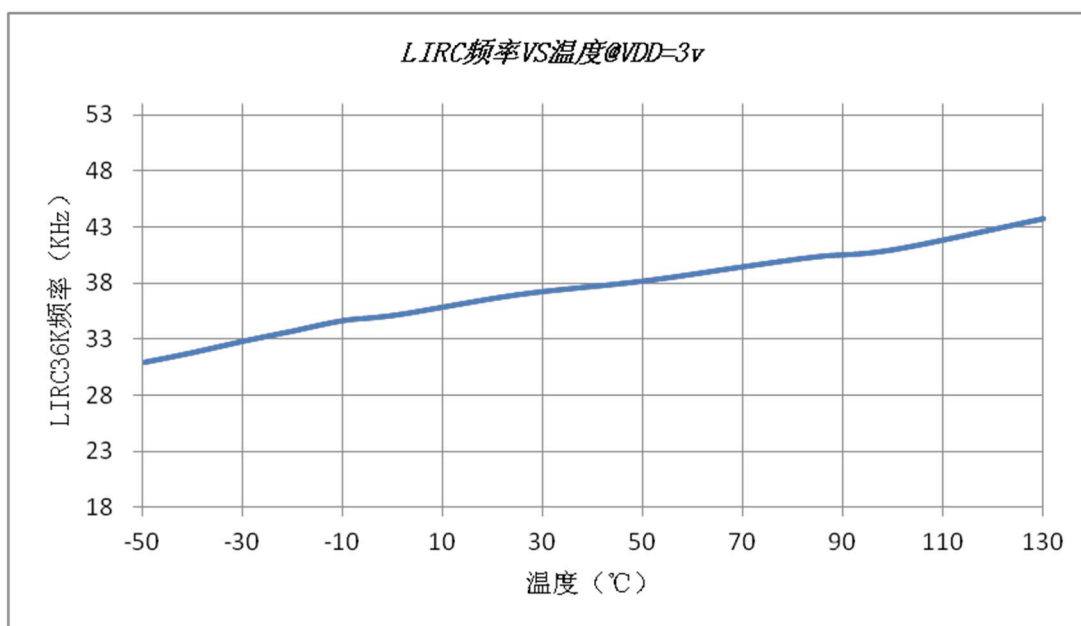


12.17 HIRC=16MHz 频率





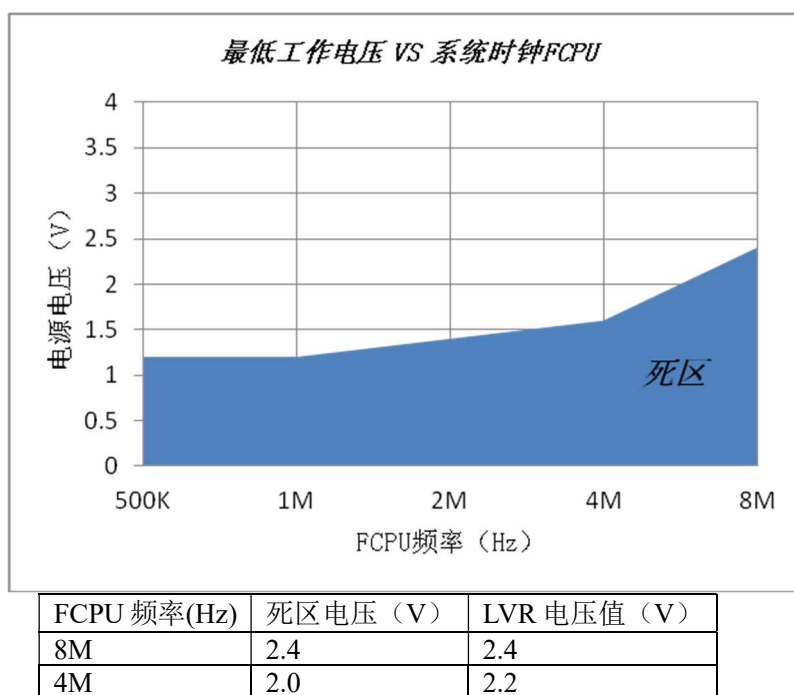
12.18 LIRC=36KHz 频率



12.19 最低工作电压 VS 系统时钟 FCPU 关系图

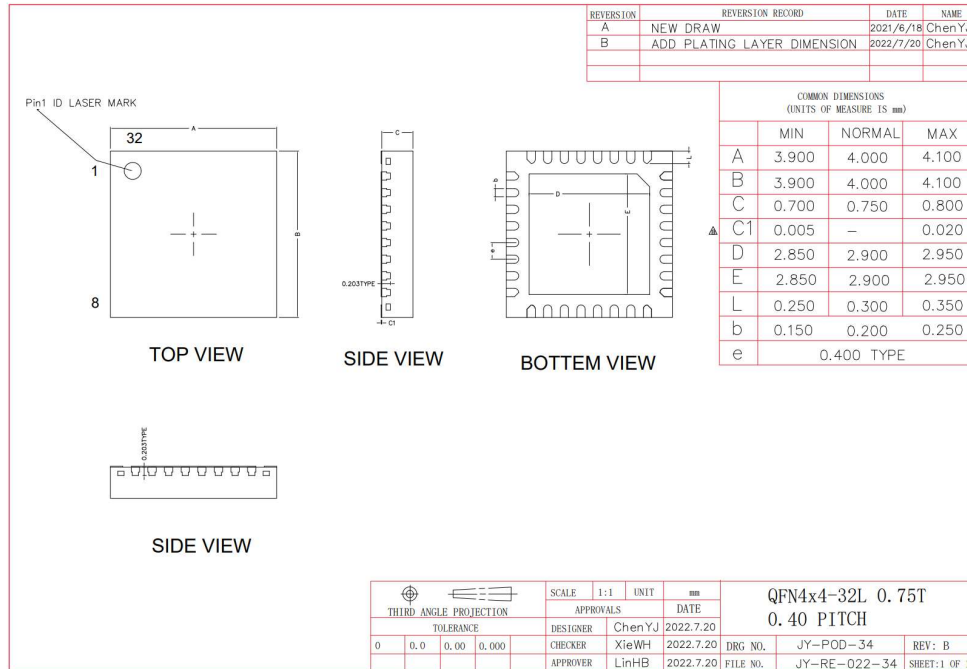
注：系统最低工作电压和系统工作频率 **FCPU** 有关，不同的工作频率 **FCPU** 最低工作电压不同。

如下图所示，当工作频率提高时系统正常工作电压也随之提高，但由于 **POR** 电压固定 (**1.2V@25℃**)，在系统最低工作电压和 **POR** 电压之间就会出现一个不能正常工作的电压区域，此区域系统不能正常工作也不会产生 **POR** 复位，称之为死区，必须根据不同的工作频率设置大于死区电压的 **LVR** 电压避免出现死区。

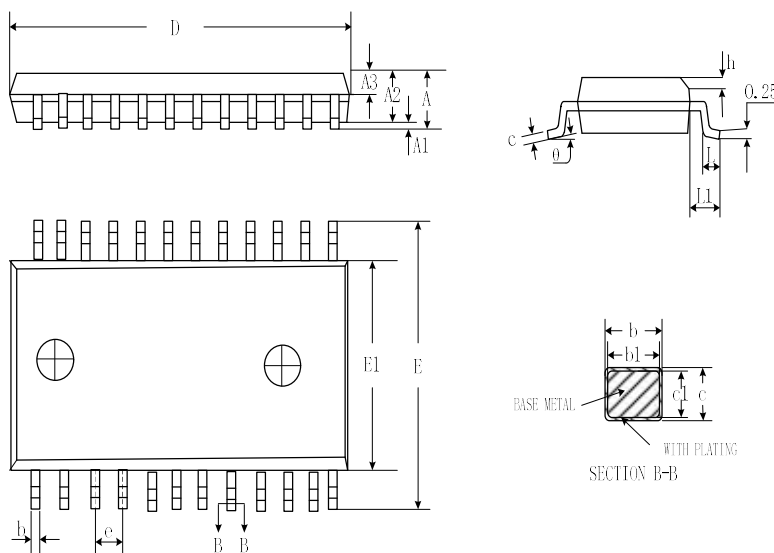


13 封装外形尺寸

LQFP32

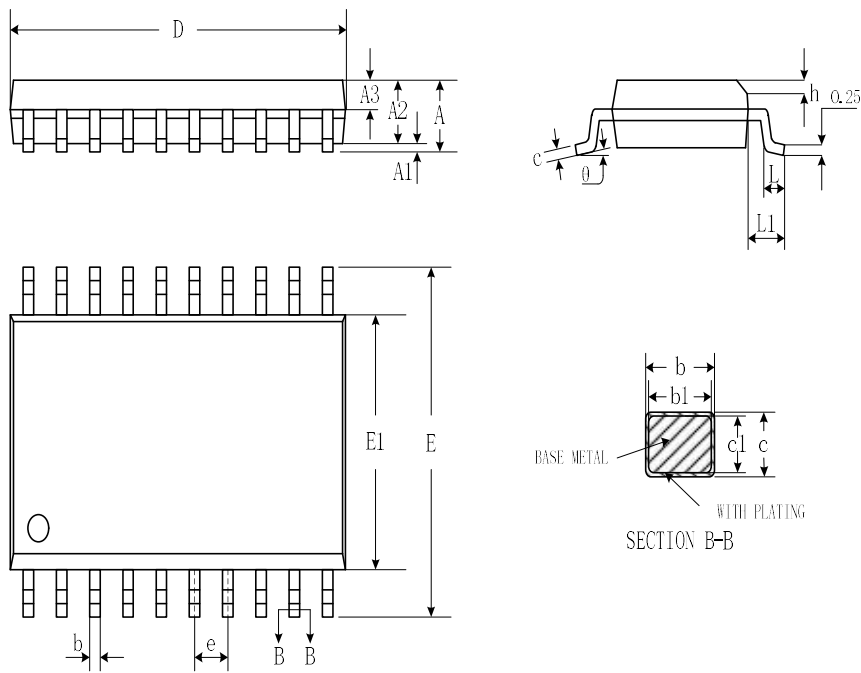


SSOP24



SSOP20

SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	-	-	1.75
A1	0.10	-	0.25
A2	1.30	1.40	1.50
A3	0.60	0.65	0.70
b	0.23	-	0.33
b1	0.22	0.25	0.28
c	0.21	-	0.26
c1	0.19	0.20	0.21
D	8.45	8.65	8.85
E	5.80	6.00	6.20
E1	3.70	3.90	4.10
e	0.635BSC		
h	0.25	-	0.50
L	0.50	-	0.80
L1	1.05BSC		
θ	0°	-	8°



14 版本修订记录

版本号	修订日期	修订内容
V1.0	2022-12-04	新建