

IP5904 数据手册

目录

1. 产品简述	4
1.1 产品特性	4
1.2 简介及系统框图	5
1.3 引脚说明	6
2. 电气参数	7
2.1 极限参数	7
2.2 推荐工作条件	7
2.3 电气特性	7
3. MCU 与存储器	9
3.1 概述	9
3.1.1 MCU 指令集	9
3.2 程序存储空间	12
3.3 数据存储空间	12
3.4 特殊功能寄存器	12
4. 系统控制模块	15
4.1 正常工作状态与低功耗	15
4.1.1 进入低功耗寄存器	15
4.2 唤醒源	15
4.2.1 唤醒源查询寄存器	15
4.3 唤醒配置的相关寄存器	16
5. 中断	18
5.1 中断向量表	18
5.2 中断置起的相关寄存器	18
5.3 中断源配置相关寄存器	20
5.4 中断优先级	23
6. 系统时钟	24
6.1 高频时钟	24
6.2 低频时钟	24
6.3 时钟配置相关寄存器	24
7. I/O 端口	26
7.1 I/O 口功能	26
7.2 I/O 端口配置寄存器	26
7.2.1 I/O 端口通用配置及数据寄存器	26
7.2.2 I/O 端口复用配置寄存器	28
8. Key	30
8.1 Key 模块介绍	30
8.2 key 状态相关寄存器	30
9. 定时器	33
9.1 看门狗定时器	33
9.1.1 看门狗定时器相关寄存器	33

9.2 Timer0/Timer1/Timer2 定时器/计数器	34
9.2.1 Timer0 定时器/计数器相关寄存器	35
9.2.2 Timer1 定时器/计数器相关寄存器	36
9.2.3 Timer2 定时器/计数器相关寄存器	38
10. 模数转换 (ADC)	40
10.1 ADC 模块介绍	40
10.2 ADC 各通道计算公式	40
10.3 ADC 相关寄存器汇总	40
10.3.1 ADC 配置寄存器	40
10.3.2 ADC 数据寄存器	42
11. Charger 模块	43
11.1 Charger 模块介绍	43
11.2 Charger 相关寄存器	43
12. 比较器模块	45
12.1 比较器模块介绍	45
12.2 比较器相关寄存器	45
13. PWM 模块	49
13.1 PWM 模块介绍	49
13.1.1 PWM 类型	49
13.1.2 工作模式	49
13.1.3 极性控制	50
13.2 PWM 相关寄存器	50
14. BLN 模块	53
14.1 BLN 模块介绍	53
14.1.1 概述	53
14.1.2 内部结构图	53
14.2 BLN 相关寄存器	53
15. UART 模块	56
15.1 UART 模块介绍	56
15.2 UART 相关寄存器	56
16. 典型应用原理图	59
17. 封装信息	60
18. IC 丝印说明	61
19. 责任及版权声明	62

1. 产品简述

1.1 产品特性

- **MCU**
 - ◇ 兼容 8051 指令集内核
 - ◇ 4K Bytes OTP 型程序存储器
 - ◇ 192 Bytes RAM 通用数据存储器，支持直接寻址、间接寻址等多种寻址方式
 - ◇ 系统时钟：低速 32KHz、高速 16MHz
 - ◇ 3 个 16bit 定时器：Timer0、Timer1、Timer2
 - ◇ 2 个 16bit PWM，复用定时器 Timer1、Timer2
 - ◇ 1 个 8bit PWM，支持同频率的 2 组 PWM（有互补和死区可调功能）或 4 个同频率不同占空比的 PWM 输出
 - ◇ 7 通道 12bit 位宽 ADC：VGP0/VGP1/VGP2/ADC3/VBAT/VIN/IVIN(充电电流)
 - ◇ IO 口资源：支持最多 10 个普通 IO 口
- **充电**
 - ◇ 最大 620mA 线性充电，电流可调节 20mA/Step
 - ◇ 支持 4.20V、4.30V、4.35V 和 4.4V 电池
 - ◇ 支持自动再充电功能
 - ◇ 可反馈充电状态
- **低功耗**
 - ◇ 待机功耗小于 5 μ A
 - ◇ DeepSleep 待机功耗小于 2 μ A
- **多重保护、高可靠性**
 - ◇ 输入过压、过充保护
 - ◇ 整机过温保护
- ESD 4KV
- 封装方式 SOP16L

1.2 简介及系统框图

IP5904 内置兼容 8051 指令集 8-bit MCU，主频时钟最高为 16MHz，内置 192 Bytes RAM 及 4K Bytes OTP ROM 为便携电子设备提供充足的编程空间。

IP5904 支持 10 个 GPIO 口功能，其中 IO4,IO5,IO6,IO7 支持 30mA 大电流驱动能力，其他 IO 支持 20mA 驱动能力。

IP5904 的线性充电提供最大 620mA 充电电流，可灵活配置充电电流。内置输入欠压、过压、过流、IC 过温和输入欠压智能调节充电电流功能，提供安全可靠的充电管理。

IP5904 内置 100uA 恒流源输出及 12bit ADC，可实现自定义 NTC 功能。

IP5904 内置 2 个 16bit 的 PWM（跟 Timer1 和 Timer2 复用）和 1 个 8bit 的 PWM。

IP5904 的高集成度与丰富功能，使其在应用时仅需极少的外围器件，并有效减小整体方案的尺寸，降低 BOM 成本，为小型电子设备提供高集成度解决方案。

图 1 为系统框架图。

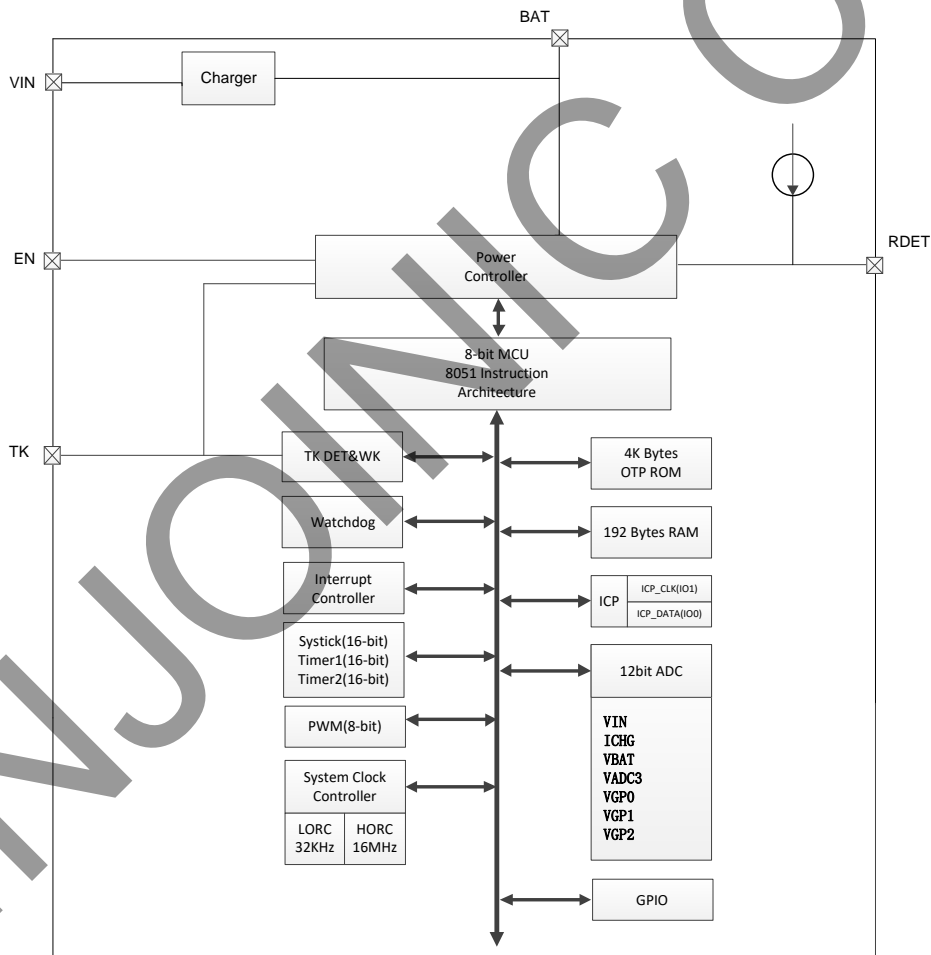


图 1 IP5904 内部系统框图

1.3 引脚说明

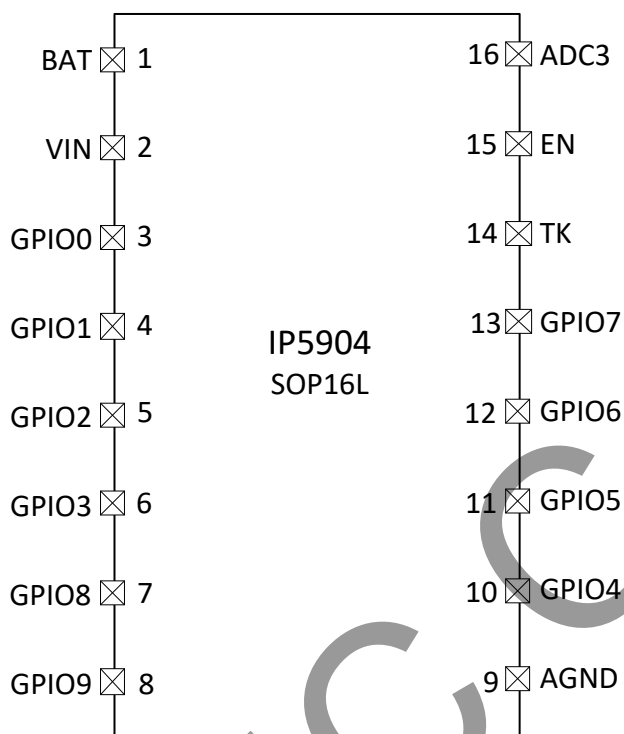


图 2 IP5904 引脚图

引脚编号	引脚名字	引脚功能		
		F1	F2	F3
1	BAT	电池正端		
2	VIN	充电输入端		
3	GPIO0	通用 IO0	ADCGP0	
4	GPIO1	通用 IO1	ADCGP1	T2_PWM
5	GPIO2	通用 IO2	ADCGP2	SIRQ
6	GPIO3	通用 IO3	SIRQ	T1_PWM
7	GPIO8	通用 IO8	UART_TX	
8	GPIO9	通用 IO9	UART_RX	
9	GND	接地		
10	GPIO4	通用 IO4	PWM0	
11	GPIO5	通用 IO5	PWM1	
12	GPIO6	通用 IO6	PWM2	
13	GPIO7	通用 IO7	PWM3	
14	TK	通用输入口 TK		
15	EN	按键功能		
16	ADC3	插拔检测	ADC	

2. 电气参数

2.1 极限参数

参数	符号	值	单位
引脚耐压范围	VIN/BAT/TK/GPIO/EN/ADC3	-0.3 ~ 8	V
结温范围	T _J	-40 ~ 150	°C
存储温度范围	T _{stg}	-60 ~ 150	°C
热阻（结温到环境）	θ _{JA}	70	°C/W
人体模型（HBM）	ESD	4	KV

*高于绝对最大额定值部分所列数值的应力有可能对器件造成永久性的损害，在任何绝对最大额定值条件下暴露的时间过长都有可能影响器件的可靠性和使用寿命。

2.2 推荐工作条件

参数	符号	最小值	典型值	最大值	单位
输入电压	VIN	4.5	5	6.0	V
工作环境温度	T _A	-20	--	85	°C

*超出这些工作条件，器件工作特性不能保证。

2.3 电气特性

除特别说明，T_A=25°C，V_{IN}=5V，V_{BAT}=3.7V

参数	符号	测试条件	最小值	典型值	最大值	单位
充电系统						
输入电压	V _{IN}		4.5	5	6	V
输入过压阈值	上升电压		5.8	6	6.2	V
输入过压保护迟滞				200		mV
恒流充电电流	I _{CC}	设置充电电流为 620mA			620	mA
CV 恒压充电电压	V _{CV}	CV _{4.2V}	4.16	4.20	4.24	V
		CV _{4.30V}	4.28	4.30	4.34	V
		CV _{4.35V}	4.33	4.35	4.4	V
		CV _{4.4V}	4.38	4.40	4.44	V
充满停充检测电压	V _{SV}			V _{CV} -0.05		V

充满后回充电压	V_{RC}			$V_{CV}-0.1$		V
涓流充电电流	I_{TK}	VIN=5V,		$1/10 I_{CC}$		mA
控制系统						
电池待机电流	I_{STB_STOP}	VIN=0V,VBAT=3.7V ,STOP 模式		5		uA
	I_{STB_DS}	VIN=0V,VBAT=3.7V ,DeepSleep 模式		2		uA
普通 IO 口驱动电流	I_{Gpio}	普通 IO 口强驱动电流		20		mA
强 IO 口驱动电流	I_{HGpio}	IO4,IO5,IO6,IO7 口强驱动电流		30		mA
热关断温度	T_{OTP}	上升温度	130	140	150	°C
热关断迟滞	ΔT_{OTP}		30	40	50	°C

*上述数据基于 demo 程序测试，多个参数规格可软件配置

3. MCU 与存储器

3.1 概述

3.1.1 MCU 指令集

IP5904 内置 8-bit MCU，其指令集如下。

算术操作指令

Mnemonic	Description	Code	Bytes	Cycles
ADD A,Rn	Add register to accumulator	28-2F	1	1
ADD A,direct	Add direct byte to accumulator	25	2	2
ADD A,@Ri	Add indirect RAM to accumulator	26-27	1	2
ADD A,#data	Add immediate data to accumulator	24	2	2
ADDC A,Rn	Add register to accumulator with carry flag	38-3F	1	1
ADDC A,direct	Add direct byte to A with carry flag	35	2	2
ADDC A,@Ri	Add indirect RAM to A with carry flag	36-37	1	2
ADDC A,#data	Add immediate data to A with carry flag	34	2	2
SUBB A,Rn	Subtract register from A with borrow	98-9F	1	1
SUBB A,direct	Subtract direct byte from A with borrow	95	2	2
SUBB A,@Ri	Subtract indirect RAM from A with borrow	96-97	1	2
SUBB A,#data	Subtract immediate data from A with borrow	94	2	2
INC A	Increment accumulator	04	1	1
INC Rn	Increment register	08-0F	1	2
INC direct	Increment direct byte	05	2	3
INC @Ri	Increment indirect RAM	06-07	1	3
INC DPTR	Increment data pointer	A3	1	1
DEC A	Decrement accumulator	14	1	1
DEC Rn	Decrement register	18-1F	1	2
DEC direct	Decrement direct byte	15	2	3
DEC @Ri	Decrement indirect RAM	16-17	1	3
MUL AB	Multiply A and B	A4	1	5
DIV	Divide A by B	84	1	5
DA A	Decimal adjust accumulator	D4	1	1

Mnemonic	Description	Code	Bytes	Cycles
ANL A,Rn	AND register to accumulator	58-5F	1	1
ANL A,direct	AND direct byte to accumulator	55	2	2
ANL A,@Ri	AND indirect RAM to accumulator	56-57	1	2
ANL A,#data	AND immediate data to accumulator	54	2	2
ANL direct,A	AND accumulator to direct byte	52	2	3
ANL direct,#data	AND immediate data to direct byte	53	3	4
ORL A,Rn	OR register to accumulator	48-4F	1	1
ORL A,direct	OR direct byte to accumulator	45	2	2
ORL A,@Ri	OR indirect RAM to accumulator	46-47	1	2
ORL A,#data	OR immediate data to accumulator	44	2	2
ORL direct,A	OR accumulator to direct byte	42	2	3
ORL direct,#data	OR immediate data to direct byte	43	3	4
XRL A,Rn	Exclusive OR register to accumulator	68-6F	1	1
XRL A,direct	Exclusive OR direct byte to accumulator	65	2	2
XRL A,@Ri	Exclusive OR indirect RAM to accumulator	66-67	1	2
XRL A,#data	Exclusive OR immediate data to accumulator	64	2	2
XRL direct,A	Exclusive OR accumulator to direct byte	62	2	3
XRL direct,#data	Exclusive OR immediate data to direct byte	63	3	4
CLR A	Clear accumulator	E4	1	1
CPL A	Complement accumulator	F4	1	1
RL A	Rotate accumulator left	23	1	1
RLC A	Rotate accumulator left through carry	33	1	1
RR A	Rotate accumulator right	03	1	1
RRC A	Rotate accumulator right through carry	13	1	1
SWAP A	Swap nibbles within the accumulator	C4	1	1

数据传输指令

Mnemonic	Description	Code	Bytes	Cycles
MOV A,Rn	Move register to accumulator	E8-EF	1	1
MOV A,direct	Move direct byte to accumulator	E5	2	2
MOV A,@Ri	Move indirect RAM to accumulator	E6-E7	1	2
MOV A,#data	Move immediate data to accumulator	74	2	2
MOV Rn,A	Move accumulator to register	F8-FF	1	2
MOV Rn,direct	Move direct byte to register	A8-AF	2	4
MOV Rn,#data	Move immediate data to register	78-7F	2	2
MOV direct,A	Move accumulator to direct byte	F5	2	3
MOV direct,Rn	Move register to direct byte	88-8F	2	3
MOV direct1,direct2	Move direct byte to direct byte	85	3	4
MOV direct,@Ri	Move indirect RAM to direct byte	86-87	2	4
MOV direct,#data	Move immediate data to direct byte	75	3	3
MOV @Ri,A	Move accumulator to indirect RAM	F6-F7	1	3
MOV @Ri,direct	Move direct byte to indirect RAM	A6-A7	2	5
MOV @Ri,#data	Move immediate data to indirect RAM	76-77	2	3
MOV DPTR,#data16	Load data pointer with a 16-bit constant	90	3	3
MOVC A,@A+DPTR	Move code byte relative to DPTR to accumulator	93	1	3
MOVC A,@A+PC	Move code byte relative to PC to accumulator	83	1	3
MOVX A,@Ri	Move external RAM (8-bit addr.) to A	E2-E3	1	3-10
MOVX A,@DPTR	Move external RAM (16-bit addr.) to A	E0	1	3-10
MOVX @Ri,A	Move A to external RAM (8-bit addr.)	F2-F3	1	4-11
MOVX @DPTR,A	Move A to external RAM (16-bit addr.)	F0	1	4-11
PUSH direct	Push direct byte onto stack	C0	2	4
POP direct	Pop direct byte from stack	D0	2	3
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 indir. RAM with A	D6-D7	1	3

位操作指令

Mnemonic	Description	Code	Bytes	Cycles
CLR C	Clear carry flag	C3	1	1
CLR bit	Clear direct bit	C2	2	3
SETB C	Set carry flag	D3	1	1
SETB bit	Set direct bit	D2	2	3
CPL C	Complement carry flag	B3	1	1
CPL bit	Complement direct bit	B2	2	3
ANL C,bit	AND direct bit to carry flag	82	2	2
ANL C,/bit	AND complement of direct bit to carry	B0	2	2
ORL C,bit	OR direct bit to carry flag	72	2	2
ORL C,/bit	OR complement of direct bit to carry	A0	2	2
MOV C,bit	Move direct bit to carry flag	A2	2	2
MOV bit,C	Move carry flag to direct bit	92	2	3

Mnemonic	Description	Code	Bytes	Cycles
ACALL addr11	Absolute subroutine call	xxx11	2	6
LCALL addr16	Long subroutine call	12	3	6
RET	from subroutine	22	1	4
RETI	from interrupt	32	1	4
AJMP addr11	Absolute jump	xxx01	2	3
LJMP addr16	Long jump	02	3	4
SJMP rel	Short jump (relative addr.)	80	2	3
JMP @A+DPTR	Jump indirect relative to the DPTR	73	1	2
JZ rel	Jump if accumulator is zero	60	2	3
JNZ rel	Jump if accumulator is not zero	70	2	3
JC rel	Jump if carry flag is set	40	2	3
JNC	Jump if carry flag is not set	50	2	3
JB bit,rel	Jump if direct bit is set	20	3	4
JNB bit,rel	Jump if direct bit is not set	30	3	4
JBC bit,direct rel	Jump if direct bit is set and clear bit	10	3	4
CJNE A,direct rel	Compare direct byte to A and jump if not equal	B5	3	4
CJNE A,#data rel	Compare immediate to A and jump if not equal	B4	3	4
CJNE Rn,#data rel	Compare immed. to reg. and jump if not equal	B8-BF	3	4
CJNE @Ri,#data rel	Compare immed. to ind. and jump if not equal	B6-B7	3	4
DJNZ Rn,rel	Decrement register and jump if not zero	D8-DF	2	3
DJNZ direct,rel	Decrement direct byte and jump if not zero	D5	3	4
NOP	No operation	00	1	1

3.2 程序存储空间

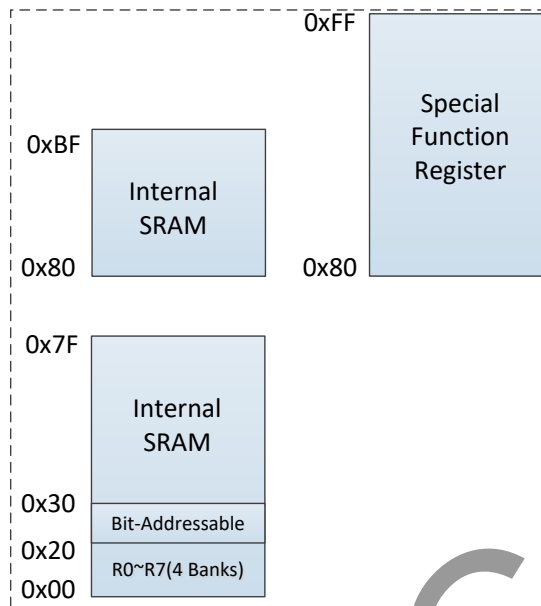
- IP5904 内置 4K OTP 程序存储空间，用来存放用户代码及储存程序。
- 程序存储空间由程序计数器来寻址，也包含数据、表格、中断入口等。
- 程序存储空间内的某些地址用作诸如复位和中断入口等特殊用途。

3.3 数据存储空间

- 内部 SRAM 空间共 192Bytes，地址 00h~BFh。
- 其中低 128bytes（地址 0x00~0x7f）支持直接和间接寻址方式，高 64bytes（地址 0x80~0xbf）只支持间接寻址方式。

3.4 特殊功能寄存器

- SFR 特殊寄存器空间地址占据数据空间高 128bytes（地址 0x80~0xff）只支持直接寻址方式。



关键字	对应的存储空间及范围
code	ROM 空间, 共占用 0x0000~0x5FBF 地址
data	片内 RAM, 直接寻址, 低 128 字节
bdata	片内 RAM, 位寻址 0x20~0x2F, 可字节访问
idata	片内 RAM, 简介选址, 256 字节, 与@Ri 对应
pdata	片外 RAM, 分页寻址的 256 字节(P2 不变), P2 改变可寻址 64KB 全空间, 与 MOVX@Ri 对应
bit	片内 RAM 位寻址区, 位地址 0x00~0x7F, 128 位

图 3 C51 存储区与存储空间的对应关系

3.5 烧录口 ICP

程序烧录口包含 2 根信号线: ICP_DATA(复用 GPIO0)和 ICP_CLK(复用 GPIO1)。

上电的时候, 若检测 GPIO0/GPIO1 均为高电平并持续 8ms, 则将 GPIO0/GPIO1 切换为 ICP 接口, 否则 GPIO0/GPIO1 当做正常功能使用。

3.6 Register

CPU 的内部寄存器有如下几个。

序号	寄存器符号	寄存器地址	寄存器名称和描述
1	SP	81h	Stack Pointer
2	DPL	82h	Data Pointer Low 0
3	DVM	83h	Data Pointer High 0
4	DPL1	84h	Data Pointer Low 1
5	DVM1	85h	Data Pointer High 1
6	PCON	87h	MCU Power Control
7	CKCON	8Eh	MCU Clock Control
8	DPS	92h	Data Pointer Select Register
9	PSW	D0h	Program status word
10	ACC	E0h	Accumulator
11	B	F0h	B Register

4. 系统控制模块

4.1 正常工作状态与低功耗

- 1、芯片正常工作状态下，所有时钟均使能。
- 2、芯片的休眠模式即为低功耗模式。
- 3、执行 STOP 指令可使系统进入低功耗模式。

4.1.1 进入低功耗寄存器

STOP_EN(0x92)

Offset = 0x92, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7:1			R/W	0
0	stop_enable	写 1 进 stop 态。	W	0

4.2 唤醒源

唤醒 IP5904 有两种情况，一种是芯片首次上电，这种情况下 IC 需要激活唤醒，激活 IP5904 的方式有插入 VIN 及 EN 按下。另一种是已经激活的 IC 再次进入 standby 休眠状态后可以通过唤醒源唤醒 CPU，程序将重新从启动文件开始执行。

IP5904 系统的 Wakeup 信号源包括：

- 1、VIN 插入；
- 2、EN 按下（包括短按/长按/上升沿/下降沿等，任选一种）；
- 3、ADC3 检测唤醒
- 4、Watchdog
- 5、SIRQ

唤醒 IP5904 后，用户可以通过只读寄存器 DEBUG0(0x88)对应的 bit 位来查询唤醒方式。

4.2.1 唤醒源查询寄存器

DEBUG0(0x88)

Offset = 0x88, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7		Reserved	R	0
6	Wk_flag[6]	VINOK 唤醒	R	0
5	Wk_flag[5]	SIRQ 唤醒	R	0
4	Wk_flag[4]	Wdg 复位唤醒	R	0
3	Wk_flag[3]	Enpinwk 唤醒	R	0
2		Reserved	R	0
1	Wk_flag[1]	ADC3 唤醒	R	0
0		Reserved	R	0

4.3 唤醒配置的相关寄存器

Wakeup_CTL0(0x93)—INFO_SVCCOK

Offset = 0x93, Default=0xC5

Bit(s)	Name	Description	R/W	Reset
7	KEY_Double_EN	1: enable,连续按键时,第二次以后的按键都算双击,不出现单击 0: disable,连续按键时,奇数次按键为单击,偶数次为双击	R/W	1
6:4	WKUPEN_Mode	按键唤醒选择: 000: Disable 001: 抬起来 010: 按下去 011: 上升沿+下降沿 100: 短按 101: 长按 110: 短按,且1s内没有第二次短按才唤醒,也即1秒内只有单击才唤醒,双击不唤醒。 111: key配置为高电平复位时,上升沿唤醒使能	R/W	0
3	Batlow_Off_EN	Batlow 关机信号是否有效 1: Enable 0: Disable	R/W	0
2	Onofflong_RST_EN	超长按复位使能: 1: Enable 0: Disable	R/W	0
1	Batlow_RST_EN	Batlow 是否 reset 系统使能: 0: disable 1: enable	R/W	0
0	Wdg_RST_WK_EN	配置 Watchdog 发出 RST 后的唤醒使能: 1: Enable 0: Disable	R/W	0

Wakeup_CTL1(0x94)

Offset = 0x94, Default = 0x03

Bit(s)	Name	Description	R/W	Reset
7	Sirq 唤醒使能	1: Enable 0: Disable	R/W	0
6:4	SIRQ0_WAKEUP_SET	Sirq1 唤醒选择: 000: disable 001: 上升沿 010: 下降沿 011: 上升/下降沿 100: 低电平 101: 高电平	R/W	000
3			R/W	0
2	Adc3_Insert_en	ADC3 插拔检测唤醒使能	R/W	0
1	Long_key_timerst	0:2s 1:3s	R/W	1
0	Key_debounce	0:16ms 1:32ms	R/W	1

5. 中断

IP5904 系统的中断源包括：

- 1、ADC3 拔插检测；
- 2、EN（包括长按、双击、短按、按下、抬起）；
- 3、Watchdog；
- 4、电池低电 Batlow

硬件检测到电池低电自动关机进 Standby。Batlow 的电压有 2.7/2.8/2.9/3.0V 四挡可选，同时有 0.1V 的迟滞范围，防止误触发；

- 5、VIN 过压 VINOV，欠压 VINUV，插入 VIN_IN，拔出 VIN_OUT
- 6、UART
- 7、BLN1/BLN0
- 8、SIRQ
- 9、计时器 Timer0、Timer1、Timer2
- 10、PWM

5.1 中断向量表

中断序号	中断名称	中断向量
0		
1	ADC3	0x0b
2	EN	0x13
3	Watchdog	0x1b
4	Batlow_irq	0x23
5	VINUV/VINOV/VININ/VINOUT	0x2b
6	UART	0x33
7	BLN1/BLN0	0x3b
8	SIRQ	0x43
9	Timer0	0x4b
10	Timer1	0x53
11	Timer2	0x5b
12	PWM	0x63

5.2 中断置起的相关寄存器

IRQ_PEND0(0xD7)

Offset= 0xD7, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7		Reserved	R/W	0
6	ADC3	ADC3 插拔检测标志（写 1 清除） 1: 标志置起 0: 标志未置起	R/W	0
5		Reserved	R/W	0
4	KEY_Long	长按键：KEY 持续长按超过 2S 小于 8S（写 1 清除）	R/W	0

		1: 标志置起 0: 标志未置起		
3	KEY_Double	双击: KEY 在 1S 内出现 2 次短按 (写 1 清除) 1: 标志置起 0: 标志未置起	R/W	0
2	KEY_Short	短按键: KEY 在 60mS~1S 以内出现 1 次短按 (写 1 清除) 1: 标志置起 0: 标志未置起	R/W	0
1	KEY_UP	按键抬起来: KEY 由低电转高, 持续 60mS (写 1 清除) 1: 标志置起 0: 标志未置起	R/W	0
0	KEY_Down	按键按下去: KEY 由高电转低, 持续 60mS (写 1 清除) 1: 标志置起 0: 标志未置起	R/W	0

IRQ_PEND1(0xD8)

Offset = 0xD8, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7	Watchdog	Watchdog 中断标志 (写 1 清除) 1: 标志置起 0: 标志未置起	R/W	0
6		Reserved	R/W	0
5	Batlow	Batlow 中断标志 (写 1 清除) 1: 标志置起 0: 标志未置起	R/W	0
4		Reserved	R/W	0
3	VINUV	VIN 欠压标志 (写 1 清除) 可选寄存器配置的欠压阈值 4.7/4.6/4.25 1: 标志置起 0: 标志未置起	R/W	0
2	VINOV	VIN 过压标志 (写 1 清除) 1: 标志置起 0: 标志未置起	R/W	0
1	VIN_OUT	VIN 拔出标志 (写 1 清除) vinok 掉下去的阈值, 3.6V/3.1V 可选 1: 标志置起 0: 标志未置起	R/W	0
0	VIN_IN	VIN 插入 (写 1 清除) 1: 标志置起 0: 标志未置起	R/W	0

IRQ_PEND2(0xD9)

Offset = 0xD9, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7		Reserved	R/W	0
6	BLN1	BLN1_pending 标志 (写 1 清除) 1: 标志置起 0: 标志未置起	R/W	0
5	BLN0	BLN0_pending 标志 (写 1 清除) 1: 标志置起 0: 标志未置起	R/W	0
4	SIRQ	SIRQ 中断标志 (写 1 清除) 1: 标志置起 0: 标志未置起	R/W	0
3	PWM	PWM 中断标志 (写 1 清除) 1: 标志置起 0: 标志未置起	R/W	0
2	Timer2	Timer2 中断标志 (写 1 清除) 1: 标志置起 0: 标志未置起	R/W	0
1	Timer1	Timer1 中断标志 (写 1 清除) 1: 标志置起 0: 标志未置起	R/W	0
0		Reserved	R/W	0

5.3 中断源配置相关寄存器

GIE(0xF1)

Offset = 0xF1, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7:1	-	Reserved	R	0x00
0	GIE	系统中断使能 1: Enable 0: Disable	R/W	0

IRQ_EN0(0xDA)

Offset = 0xDA, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7		Reserved	R/W	0
6	ADC3_IRQ_EN	ADC3 插拔检测使能 1: Enable 0: Disable	R/W	0
5		Reserved	R/W	0
4	KEY_Long_IRQ_EN	KEY 长按中断使能	R/W	0

		1: Enable 0: Disable		
3	KEY_Double_IRQ_EN	KEY 双击中断使能 1: Enable 0: Disable	R/W	0
2	KEY_Short_IRQ_EN	KEY 短按中断使能 1: Enable 0: Disable	R/W	0
1	KEY_UP_IRQ_EN	KEY 抬起来中断使能 1: Enable 0: Disable	R/W	0
0	KEY_Down_IRQ_EN	KEY 按下中断使能 1: Enable 0: Disable	R/W	0

IRQ_EN1(0xDB)

Offset = 0xDB, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7	Watchdog_IRQ_EN	Watchdog 中断使能 1: Enable 0: Disable	R/W	0
6		Reserved	R/W	0
5	Batlow_IRQ_EN	电池低电中断使能 1: Enable 0: Disable	R/W	0
4		Reserved	R/W	0
3	VINUV_IRQ_EN	VIN 欠压中断使能 1: Enable 0: Disable	R/W	0
2	VINOV_IRQ_EN	VIN 过压中断使能 1: Enable 0: Disable	R/W	0
1	VIN_OUT_IRQ_EN	VIN 拔出中断使能 1: Enable 0: Disable	R/W	0
0	VIN_IN_IRQ_EN	VIN 插入中断使能 1: Enable 0: Disable	R/W	0

IRQ_EN2(0xDC)

Offset = 0xDC, default=0x00

Bit(s)	Name	Description	R/W	Reset
7		Reserved	R/W	0
6	BLN1_IRQ_EN	BLN1 中断使能 1: Enable 0: Disable	R/W	0
5	BLN0_IRQ_EN	BLN0 中断使能 1: Enable 0: Disable	R/W	0
4	SIRQ_IRQ_EN	SIRQ 中断使能 1: Enable 0: Disable	R/W	0
3	PWM_IRQ_EN	PWM 中断使能 1: Enable 0: Disable	R/W	0
2	Timer2_IRQ_EN	Timer2 中断使能 1: Enable 0: Disable	R/W	0
1	Timer1_IRQ_EN	Timer1 中断使能 1: Enable 0: Disable	R/W	0
0		Reserved	R/W	0

QIRX(0xD5)

Offset = 0xD5, Default=0x10

Bit(s)	Name	Description	R/W	Reset
7:5	SIRQ0_sel	Sirq1 中断模式选择: 000: reserved 001: 上升沿 010: 下降沿 011: 上升/下降沿 100: 低电平 101: 高电平 110: reserved 111: reserved	R/W	000
6:0		Reserved	R/W	0

5.4 中断优先级

IP5904 内核共有 4 个中断优先级，其配置如下：

ip1.x	ip0.x	Priority Level
0	0	Level0 (lowest)
0	1	Level1
1	0	Level2
1	1	Level3 (highest)

Bits	Group	
IP1.0, IP0.0		UART
IP1.1, IP0.1	Timer0	ADC3
IP1.2, IP0.2	Timer1	BLN1/BLN0
IP1.3, IP0.3	Timer2	PWM
IP1.4, IP0.4	SIRQ	Watchdog
IP1.5, IP0.5	Batlow_irq	EN/ VINUV/VINOV/VININ/VINOUT

INT_PRO0(0x90)

Offset = 0x90, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7	-		R/W	0
6	-		R/W	0
5	IP0.5	中断优先级低位，与 IP1.5 组合决定中断优先级	R/W	0
4	IP0.4	中断优先级低位，与 IP1.4 组合决定中断优先级	R/W	0
3	IP0.3	中断优先级低位，与 IP1.3 组合决定中断优先级	R/W	0
2	IP0.2	中断优先级低位，与 IP1.2 组合决定中断优先级	R/W	0
1	IP0.1	中断优先级低位，与 IP1.1 组合决定中断优先级	R/W	0
0	IP0.0	中断优先级低位，与 IP1.0 组合决定中断优先级	W	0

INT_PRO1(0x91)

Offset = 0x91, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7	-		R/W	0
6	-		R/W	0
5	IP1.5	中断优先级高位，与 IP0.5 组合决定中断优先级	R/W	0
4	IP1.4	中断优先级高位，与 IP0.4 组合决定中断优先级	R/W	0
3	IP1.3	中断优先级高位，与 IP0.3 组合决定中断优先级	R/W	0
2	IP1.2	中断优先级高位，与 IP0.2 组合决定中断优先级	R/W	0
1	IP1.1	中断优先级高位，与 IP0.1 组合决定中断优先级	R/W	0
0	IP1.0	中断优先级高位，与 IP0.0 组合决定中断优先级	W	0

6. 系统时钟

芯片运行所需要的时钟源由振荡器提供，本芯片提供的振荡器有两种：

16Mhz 高频 RC 振荡器 HOSC

32Khz 低频 RC 振荡器 LOSC

在 workon 模式下，HOSC 和 LOSC 均振荡

在 standby 模式下，LOSC 振荡，HOSC 暂停

CPU 的时钟源固定为系统主时钟 HOSC，CPU 的时钟频率 FCLK 可以根据需要灵活选择。振荡器除了作为系统时钟源外，还可以为看门狗定时器、ADC 电路、Timer 等提供所需要的时钟源。

6.1 高频时钟

芯片内置 1 个振荡频率为 16MHz 的高精度 HOSC 振荡器，其分频时钟可用作系统主时钟源。

6.2 低频时钟

芯片内置 1 个振荡频率为 32KHz 的 LOSC 振荡器，LOSC 在工作状态和低功耗状态为相关电路提供时钟。

6.3 时钟配置相关寄存器

CMU_CTL0(0xD1)

Offset = 0xD1, Default=0xC6

Bit(s)	Name	Description	R/W	Reset
7		Reserved	R/W	1
6		Reserved	R/W	1
5:4	Adc_div	00: hosc/4 01: hosc/8 10: hosc/16 11: hosc/32	R/W	00
3:1	Hclk_div	000:1 001:2 010:3 011:4 100:5 101:6 110:7 111:8	R/W	011
0		Reserved	R/W	0

CMU_CTL1(0xD2)

Offset = 0xD2, Default=0xFF

Bit(s)	Name	Description	R/W	Reset
7	IO3_CMP_EN	EN Bit0 : 根据比较器 CMP 输出结果关闭 IO3 输出功能使能开关 1 disable 因为这个寄存器复位值是 1 0 enable	R/W	1
6		Reserved	R/W	1
5	Pwm_clk_en	PWM 时钟使能: 1:enable 0:disable	R/W	1
4	Timer_clk_en	Timer 时钟使能: 1:enable 0:disable	R/W	1
3	Adc_clk_en	adc 时钟使能: 1:enable 0:disable	R/W	1
2	Blk_clk_en	blk 时钟使能: 1:enable 0:disable	R/W	1
1	Uart_clk_en	uart 时钟使能: 1:enable 0:disable	R/W	1
0		Reserved	R/W	1

RMU_CTL1(0xD3)

Offset = 0xD3, Default=0xFF

Bit(s)	Name	Description	R/W	Reset
7	IO3_CMP_POLY	IO3_CMP 功能极性设置 POLY Bit1 1: 上升沿触发保护 这个寄存器复位值是 1 0: 下降沿触发保护	R/W	1
6		Reserved	R/W	1
5	Pwm_reset_en	0:reset 1:normal	R/W	1
4	Timer_reset_en	0:reset 1:normal	R/W	1
3	Adc_reset_en	0:reset 1:normal	R/W	1
2	Blk_reset_en	0:reset 1:normal	R/W	1
1	Uart_reset_en	0:reset 1:normal	R/W	1
0		Reserved	R/W	1

系统唤醒后，软件需要先复位模块再用模块功能

7. I/O 端口

7.1 I/O 口功能

通用 IO 口支持三状态：输入、输出及高阻态；

推挽输出时，有强弱两种驱动电流（20mA/5mA），其中 IO4、IO5、IO6、IO7 支持大驱动电流（30mA）；
 为输入时可配内部弱上拉或者弱下拉（100K）；

如下表所示：

IO0~IO9 口均可用做通用 IO 口；

IO0~IO2 可复用为 ADC。

IO4~IO7 可复用为 PWM，IO1、IO3、IO4、IO5 能复用成定时器 PWM 功能。

IO0、IO1、IO2、IO8、IO9 能复用成 BLN 功能。

IO2、IO3 能复用成 SIRQ 功能。

IO9、IO8 能复用成 CMP 功能，用于比较，IO2 能把 CMP 结果输出出来

序号	通用功能	复用功能			烧录
IO9	GPIO9	UART_RX	BLN1	CMPN	
IO8	GPIO8	UART_TX	BLN0	CMPP	
IO7	GPIO7	UART_RX	PWM3		
IO6	GPIO6	UART_TX	PWM2		
IO5	GPIO5	BLN1	PWM1	T2_PWM	
IO4	GPIO4	BLN0	PWM0	T1_PWM	
IO3	GPIO3	T1_PWM	SIRQ		
IO2	GPIO2	ADC2	SIRQ	CMPO	
IO1	GPIO1	ADC1	UART_RX	T2_PWM	ICP_CLK
IO0	GPIO0	ADC0	UART_TX	BLN0	ICP_DAT

表 1 IP5904 IO 口功能表

7.2 I/O 端口配置寄存器

7.2.1 I/O 端口通用配置及数据寄存器

GPIO_DRV(0x9D)

Offset = 0x9D default=0x00

Bit(s)	Name	Description	R/W	Reset
7:0	DRV_EH	对应 GPIO7~GPIO0 的驱动能力： 0: 弱 1: 强	R/W	0x0

GPIO_PD(0x9B)

Offset = 0x9B default=0x00

Bit(s)	Name	Description	R/W	Reset
7:0	GPIO_PD_EN	对应 GPIO7~GPIO0 的下拉使能	R/W	0x00

		1: Enable 0: Disable		
--	--	-------------------------	--	--

GPIO_PU(0x9C)

Offset = 0x9C default=0x00

Bit(s)	Name	Description	R/W	Reset
7:0	GPIO_PU	对应 GPIO7~GPIO0 的上拉使能 1: Enable 0: Disable	R/W	0x00

GPIO_OE(0x99)

Offset = 0x99 default=0x00

Bit(s)	Name	Description	R/W	Reset
7:0	GPIO_OE	对应 GPIO7~GPIO0 的输出使能	R/W	0x0

GPIO_IE(0x98)

Offset = 0x98 default=0x00

Bit(s)	Name	Description	R/W	Reset
7: 0	GPIO_IE	对应 GPIO7~GPIO0 的输入使能	R/W	0x0

GPIO_DAT(0x9A)

Offset = 0x9A default=0x00

Bit(s)	Name	Description	R/W	Reset
7:0	GPIO_DATA	对应 GPIO7~GPIO0 的输入或输出数据	R/W	0x0

GPIO_DEBUG1(0x9F)

Offset = 0x9F default=0x00

Bit(s)	Name	Description	R/W	Reset
7:6	GPIO_DRV[9:8]	对应 GPIO9~GPIO8 的驱动能力	R/W	0x0
5:4	GPIO_DAT[9:8]	对应 GPIO9~GPIO8 的输入或输出数据	R/W	0x0
3:2	GPIO_OE[9:8]	对应 GPIO9~GPIO8 的输出使能	R/W	0x0
1:0	GPIO_IE[9:8]	对应 GPIO9~GPIO8 的输入使能	R/W	0x0

7.2.2 I/O 端口复用配置寄存器

MFP_CTL0(0x95)

Offset = 0x95 default=0x00

Bit(s)	Name	Description	R/W	Reset
7: 6	GPIO3	00: GPIO3 01: T1_PWM 10: -- 11: SIRQ	R/W	00
5: 4	GPIO2	00: GPIO2 01: ADC2 10: CMPO 11: SIRQ	R/W	00
3: 2	GPIO1	00: GPIO1 01: ADC1 10: T2_PWM 11: UART_RX	R/W	00
1: 0	GPIO0	00: GPIO0 01: ADC0 10: BLN0 11: UART_TX	R/W	00

MFP_CTL1(0x96)

Offset = 0x96 default=0x00

Bit(s)	Name	Description	R/W	Reset
7: 6	GPIO7	00: GPIO7 01: -- 10: PWM3 11: UART_RX	R/W	00
5: 4	GPIO6	00: GPIO6 01: -- 10: PWM2 11: UART_TX	R/W	00
3: 2	GPIO5	00: GPIO5 01: BLN1 10: PWM1 11: T2_PWM	R/W	00
1: 0	GPIO4	00: GPIO4 01: BLN0 10: PWM0 11: T1_PWM	R/W	00

MFP_CTL2(0x97)

Offset = 0x97 default=0x00

Bit(s)	Name	Description	R/W	Reset
7: 6	GPIO_PU[9:8]	对应 GPIO9~GPIO8 的下拉使能	R/W	00
5: 4	GPIO_PD[9:8]	对应 GPIO9~GPIO8 的上拉使能	R/W	00
3: 2	GPIO9	00: GPIO9 01: UART_RX 10: CMPN 11: BLN1	R/W	00
1: 0	GPIO8	00: GPIO8 01: UART_TX 10: CMPP 11: BLN0	R/W	00

8. Key

8.1 Key 模块介绍

接上电池后，芯片处于 DeepSleep 状态，可以通过 EN 下拉，或者插入 Vin 来唤醒芯片。EN 脚为高时才可以进入 DeepSleep 模式。

IP5904 的 EN 脚在系统起来后，会复用做按键功能。如果不想充电激活，可以将按键下拉，按下为高电平，松开为低电平。

按键是否按下通过只读寄存器 DEBUG1 的 bit6 可以查询。

CPU 硬件读到这个 bit 也会触发相应的状态中断标志位，按键动作包括按键动作包括：

- 1、短按：32ms~1s 以内的低电平。此按键动作有唤醒系统功能。短按的 debounce 时间：16ms 或 32ms（默认值）；
- 2、双击：在 1s 时间内出现两次短按。双击可配置为：连续按键时，第二次以后的按键都算双击，不出现单击；或是连续按键时，奇数次按键为单击，偶数次为双击。
- 3、长按：持续出现低电平超过 2s 时间即发出长按动作。长按时间 2s（默认值）或 3s；
- 4、超长按：持续出现低电平超过 8s 时间即发出超长按动作。需要做超长按 8s 复位的功能（默认 disable）；
- 5、按下去：低电平到高电平跳变。此按键动作有唤醒系统功能。按下去持续 32ms（默认值）或 16ms 的时刻发出按键信号。
- 6、抬起来：高电平到低电平跳变。此击按键动作有唤醒系统功能。Key=0 持续超过 32ms（默认值）或 16ms，再抬起来的瞬间发出按键信号。
- 7、上升沿+下降沿：debounce 2ms，此按键动作有唤醒系统功能。

除超长按 8S 外，其它六种按键动作均有相应中断功能，中断以及中断标志请查看中断相关寄存器。

IP5904 的 TK 脚也可以用作按键功能。

按键是否按下通过只读寄存器的 DEBUG2 的 BIT2 来查询，然后通过软件来模拟按键功能，检测按键动作。

注意 TK 的按键功能，只有高电平唤醒功能，TK 脚低电平无法唤醒 CPU。

8.2 key 状态相关寄存器

DEBUG1(0x89)

Offset = 0x89, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7	-	Reserved	R	00
6	ENPINWK	按键状态指示： 1：按下 0：未按下	R	0
5:0	-	Reserved	R	00000

DEBUG2(0x8A)

Offset = 0x8A, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7	-	Reserved	R	00
6	TK_OUT	TK 脚高低电平状态指示: 1: 按下 0: 未按下	R	0
5:0	-	Reserved	R	00000

Wakeup_CTL1(0x94)

Offset = 0x94, Default = 0x03

Bit(s)	Name	Description	R/W	Reset
7	Sirq 唤醒使能	1: Enable 0: Disable	R/W	0
6:4	SIRQ0_WAKEUP_SET	Sirq1 唤醒选择: 000: disable 001: 上升沿 010: 下降沿 011: 上升/下降沿 100: 低电平 101: 高电平	R/W	000
3		Reserved	R/W	0
2	ADC3_Insert_wake_en	ADC3 插拔检测唤醒使能	R/W	0
1	Long_key_timerst	0:2s 1:3s	R/W	1
0	Key_debounce	0:16ms 1:32ms	R/W	1

IRQ_EN2(0xDC)

Offset = 0xDC, default=0x00

Bit(s)	Name	Description	R/W	Reset
7		Reserved	R/W	0
6	BLN1_IRQ_EN	BLN1 中断使能 1: Enable 0: Disable	R/W	0
5	BLN0_IRQ_EN	BLN0 中断使能 1: Enable 0: Disable	R/W	0
4	SIRQ_IRQ_EN	SIRQ 中断使能 1: Enable 0: Disable	R/W	0
3	PWM_IRQ_EN	PWM 中断使能 1: Enable 0: Disable	R/W	0
2	Timer2_IRQ_EN	Timer2 中断使能 1: Enable 0: Disable	R/W	0
1	Timer1_IRQ_EN	Timer1 中断使能 1: Enable 0: Disable	R/W	0
0	TK_WAKEUP_EN	TK 脚高电平唤醒功能使能 1: Enable 0: Disable	R/W	0

TKREG0H(0xF6)

Offset = 0xF6, Default = 0x10

Bit(s)	Name	Description	R/W	Reset
7:2		Reserved	R/W	00010 0
1	EN_TKDET	TK 检测使能 1: Enable 0: Disable	R/W	0
0		Reserved	R/W	0

9. 定时器

9.1 看门狗定时器

芯片内建一个看门狗定时器，他可以在发生软件故障的时候，将芯片复位。使能看门狗时，若用户程序异常，清除定时器看门狗失败，则在预定的时间到达后，看门狗会发出中断信号或者直接使系统复位。

IP5904 集成了 WDT 定时器，特性如下：

- 1、定时器时钟源为内部 32Khz RC 时钟。
- 2、8 档 WDT 定时计数器。
- 3、中断功能
- 4、复位功能
- 5、唤醒功能

9.1.1 看门狗定时器相关寄存器

WDCTL(0xCF)

Offset = 0xCF, Default=0xF4

Bit(s)	Name	Description	R/W	Reset
7	WD_EN	Watchdog 使能: 0: Disable 1: Enable	R/W	0x1
6:4	WD_CLK	Watchdog 时间选择(RC_LOSC source, 32K): CLK Watch Dog Length 000 2hours 001 64 s 010 128 ms 011 256 ms 100 1 s 101 2 s 110 4 s 111 8 s 注意：修改 Watchdog 的时间的时候，需要先 Disable WD_EN 再改时间再 Enable WD_EN	R/W	111
3: 2	WD_MODE	Watchdog Signal (IRQ or Reset)选择 11: Reserved 10: 无 reset 功能，只发出唤醒系统信号 01: reset 后进 standby，是否再唤醒系统根据 WK_CTL0 的 bit0 来决定 00: IRQ	R/W	01
1	-	Reserved	R	0
0	-	Reserved	W	0

WD_CLEAR(0x80)

Offset = 0x80, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7:1	-	Reserved	R/W	0x0
0	CLEAR	每次写 1 动作会清 watchdog 计数器。 读取的值为正常写入的值。	R/W	0

9.2 Timer0/Timer1/Timer2 定时器/计数器

芯片支持三个定时器 Timer0、Timer1 和 Timer2，Timer1/2 定时器基本一样，各有一个 pwm 控制器，均支持三种计数模式：

1、Normal Mode: 定时器从 0 开始累加计数，Timer 计时到 Timer_Val 后自动停止。

2、Circle Mode: 定时器从 0 开始累加计数，计数值加到目标值后，发出中断标志，且计数器持续累加，直到累加到最大值 0xFFFF 溢出后又从 0 开始累加，加到目标值后又发出中断标志，如此不断循环。

3、Reload Mode: 定时器从 0 开始累加计数，计数值加到目标值后，发出中断标志，然后计数值清零，从零开始重新计数。

启动 Timer 时，需先配置好 Timer clk/Mode，计数器低 8 位寄存器，最后写入计数器高 8 位寄存器，如果还需要 pwm 功能的话，还需要配置 Timer_pwm，再启动 Timer 计时。

关于在 timer 运行中改变计数值的问题：

1: normal mode: 运行过程中改变数值时计数器不清 0，继续计数到改动后的数值后结束（结果：改动时刻，新的改动值比已经计数到的值大，继续计数到新值结束；改动时刻，新的改动值比已经计数到的值小，会计数到 24'hffff 后，再次从 0 计数到新值后结束）。

2: circle mode: 在本轮计数达到计数器上限（16 位）后开始的下一轮计数过程中，从零开始计数，达到更改的数值后发出中断，然后继续计数到上限，并循环。

3: reload mode: 运行过程中改变数值时计数器不清 0，继续计数到改动后的数值后结束（结果：改动时刻，新的改动值比已经计数到的值大，继续计数到新值结束；改动时刻，新的改动值比已经计数到的值小，会计数到 24'hffff 后，再次从 0 计数到新值后结束）。

4: PWM1/2 的功能是一样的。每个跳变沿都有中断

Timer0 是给系统使用的 systick 定时器，以 32khz 频率计数，每 1ms 计数一个 bit，总宽度为 16bits。每次读 Timer0 的计数器时，需要保证先读 T0_CNT0，再读 T0_CNT1，cpu 在读 T0_CNT 时系统会停止更新计数器值（但是会继续计数），直到 T0_CNT 寄存器全部读完才会更新计数器值。注意不要在中断程序中读取 T0_CNT 寄存器，以免造成错误。为方便软件处理，计数器每次加 1 时置起一个标志位 t0_1ms_fg，计数器每加到 8 的倍数就置起一个标志位 t0_8ms_fg，计数器每加到 1024 的倍数就置起一个标志位 t0_1024ms_fg，计数器溢出时置起一个标志位 t0_overflow_fg，这些标志位都只能由软件清零。

9.2.1 Timer0 定时器/计数器相关寄存器

T0_CTL(0xC0)

Addr=0xC0 default=0x00

Bit(s)	Name	Description	R/W	Reset
7	OVERFLOW_FG	计数器溢出标志位	R/W	0
6	1024ms_FG	Timer0计数溢出标志位, 软件写1清0	R/W	0
5	8ms_FG	8ms中断pending, 软件写1清0	R/W	0
4	1ms_FG	1ms中断pending, 软件写1清0	R/W	0
3	T0_IRQ1024ms_EN	1024ms中断使能	R/W	0
2	T0_IRQ8ms_EN	8ms中断使能	R/W	0
1	T0_IRQ1ms_EN	1ms中断使能	R/W	0
0	T0_EN	Timer0 Enable: 1: Enable 0: Disable	R/W	0

T0_CNT0(0xC1)

Timer0 Count0 Register

Addr=0xC1 default=0x00

Bit(s)	Name	Description	R/W	Reset
7:0	TOCNT0	TOCNT 低八位	R	0x0

每次从 disable 到 enable, TOCNT 自动从 0 开始计时。T0 的 clock 是 32Khz 频率。

T0_CNT1(0xC2)

Timer0 Count1 Register

Addr=0xC2 default=0x00

Bit(s)	Name	Description	R/W	Reset
7:0	TOCNT1	TOCNT 高八位	R	0x0

9.2.2 Timer1 定时器/计数器相关寄存器

T1_CTL(0xC3)

Timer1 Control Register

Addr=0xC3 default=0x00

Bit(s)	Name	Description	R/W	Reset
7	Timer_enable	1: enable 0: disable 写 1 启动 timer, 如果是 normal 模式, 计数完成后硬件自动将 enable 清 0	R/W	0
6:4	Timer1_clk_div	Timer1 clock source divider: 000: HCLK/1 001: HCLK/2 010: HCLK/4 011: HCLK/16 100: HCLK/32 101: HCLK/64 110: HCLK/128 111: HCLK/256	R/W	0x0
3:2	Mode	Timer1 Mode 00: Disable 01: Circle mode 10: Normal mode 11: Reload mode	R/W	00
1	-	Reserved	R/W	0
0	POLARITY	PWM1 polarity control: 0: The Timer1 Value >=T1_PWM, out 1 The Timer1 Value < T1_PWM, out 0 1: The Timer1 Value >=T1_PWM, out 0 The Timer1 Value < T1_PWM, out 1	R/W	0

T1_PL(0xC4)

Timer1 Period Low Register

Addr=0xC4 default=0x00

Bit(s)	Name	Description	R/W	Reset
7:0	T1PL	Timer1 Period, when the Timer count reach T1P, the Timer pending=1 .	R/W	0x0

T1_PH(0xC5)

Timer1 Period High Register

Addr=0xC5 default=0x00

Bit(s)	Name	Description	R/W	Reset
7:0	T1PH	Timer1 Period, when the Timer count reach T1P, the Timer pending=1 .	R/W	0x0

T1_PWML(0xC6)

Timer1 PWM Low Register

Addr=0xC6 default=0x00

Bit(s)	Name	Description	R/W	Reset
7:0	T1PWML	Timer1PWM Low for PWM inversion, $(T1PWMH \ll 8 + T1PWML) / (T1PH \ll 8 + T1PL)$ is PWM's duty cycle	R/W	0x0

T1_PWMH(0xC7)

Timer1 PWM High Register

Addr=0xC7 default=0x00

Bit(s)	Name	Description	R/W	Reset
7:0	T1PWMH	Timer1PWM High for PWM inversion, $(T1PWMH \ll 8 + T1PWML) / (T1PH \ll 8 + T1PL)$ is PWM's duty cycle	R/W	0x0

T1PWM_EDGE(0xC8)

Timer1 edge pending

Addr=0xC8 default=0x00

Bit(s)	Name	Description	R/W	Reset
7:2	-	-	R	0x0
1	T1PWM_NEG	t1pwm negedge pending 1->0,写1清0	RW	0
0	T1PWM_POS	t1pwm posedge pending 0->1,写1清0	RW	0

9.2.3 Timer2 定时器/计数器相关寄存器

T2_CTL(0xC9)

Timer2 Control Register

Addr=0xC9 default=0x00

Bit(s)	Name	Description	R/W	Reset
7	Timer_enable	1: enable 0: disable 写 1 启动 timer, 如果是 normal 模式, 计数完成后硬件自动将 enable 清 0	R/W	0
6:4	Timer2_clk_div	Timer2 clock source divider: 000: HCLK/1 001: HCLK/2 010: HCLK/4 011: HCLK/16 100: HCLK/32 101: HCLK/64 110: HCLK/128 111: HCLK/256	R/W	0x0
3:2	Mode	Timer2 Mode 00: Disable 01: Circle mode 10: Normal mode 11: Reload mode	R/W	00
1	-	Reserved	R/W	0
0	POLARITY	PWM2 polarity control: 0: The Timer2 Value >=T2_PWM, out 1 The Timer2 Value < T2_PWM, out 0 1: The Timer2 Value >=T2_PWM, out 0 The Timer2 Value < T2_PWM, out 1	R/W	0

T2_PL(0xCA)

Timer2 Period Low Register

Addr=0xCA default=0x00

Bit(s)	Name	Description	R/W	Reset
7:0	T2PL	Timer2 Period, when the Timer count reach T2P, the Timer pending=1 .	R/W	0x0

T2_PH(0xCB)

Timer2 Period High Register

Addr=0xCB default=0x00

Bit(s)	Name	Description	R/W	Reset
7:0	T2PH	Timer2 Period, when the Timer count reach T2P, the Timer pending=1 .	R/W	0x0

T2_PWML(0xCC)

Timer2 PWM Low Register

Addr=0xCC default=0x00

Bit(s)	Name	Description	R/W	Reset
7:0	T2PWML	Timer2PWM Low for PWM inversion, $(T2PWMH \ll 8 + T2PWML) / (T2PH \ll 8 + T2PL)$ is PWM's duty cycle	R/W	0x0

T2_PWMH(0xCD)

Timer2 PWM High Register

Addr=0xCD default=0x00

Bit(s)	Name	Description	R/W	Reset
7:0	T2PWMH	Timer2PWM High for PWM inversion, $(T2PWMH \ll 8 + T2PWML) / (T2PH \ll 8 + T2PL)$ is PWM's duty cycle	R/W	0x0

T2PWM_EDGE(0xCE)

Timer2 edge pending

Addr=0xCE default=0x00

Bit(s)	Name	Description	R/W	Reset
7:2	-	-	R	0x0
1	T2PWM_NEG	t2pwm negedge pending 1->0,写1清0	R/W	0
0	T2PWM_POS	t2pwm posedge pending 0->1,写1清0	R/W	0

10. 模数转换（ADC）

10.1 ADC 模块介绍

IP5904 集成一个 12 bit ADC，用来检测 7 路输入信号。

IP5904 的 ADC 只保留 1 组转换后的数据，MCU 通过轮询的方式来进行 ADC 检测，检测时，先使能对应的 ADC 通道，等转换完成后再次读取数据。

ADC 7 通道分别为 Vin 电压、IBAT 线充电流、VBAT 电压、ADC3 电压、ADC0、ADC1、ADC2。其中 ADC0、ADC1、ADC2 分别是 IO0、IO1 和 IO2 的复用功能，IC 可从 ADC0、ADC1 和 ADC2 放出 100uA 恒流源。

ADC 的时钟可配置为：Hosc/32（0.5MHz）、Hosc/16（1.0MHz）默认、Hosc/8（2.0MHz）、Hosc/4（4.0MHz）。

ADC 能支持轮询模式或固定单路模式。

ADC 平均次数：8、16、32、64。

10.2 ADC 各通道计算公式

ADC 只保留一组转换后的数据，MCU 通过轮询的方式来进行 ADC 检测，检测时，先使能对应的 ADC 通道，等待 ADC 转换完成后再次读取数据。

不同 ADC 通道采样到的数据与模拟量对应的公式不同，

电压模式：ADC0,ADC1,ADC2,ADC3 = (ADC0_DAT<11:0> -648)mV

电流模式：ADC0,ADC1,ADC2,ADC3 = (ADC0_DAT<11:0> -2048)mV

VBAT=(ADC0_DAT<11:0> +376)mV

当 VIN > VBAT（即 Vin_OK == 1）时：

$VIN = (2 * ADC0_DAT<11:0> + 752) mV$

当 VIN < VBAT（即 Vin_OK == 0）时：

$VIN = (ADC0_DAT<11:0> + 376) mV$

IBAT=(ADC0_DAT<11:0> -259.2)mA

10.3 ADC 相关寄存器汇总

10.3.1 ADC 配置寄存器

ADC_EN(0xA0)

Offset = 0xA0, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7		Reserved	R/W	0
6	IBAT	1:enable 0:disable	R/W	0
5	VIN	1:enable 0:disable	R/W	0
4	VBAT	1:enable 0:disable	R/W	0
3	ADC3	1:enable 0:disable	R/W	0
2	ADC2	1:enable 0:disable	R/W	0
1	ADC1	1:enable 0:disable	R/W	0

0	ADC0	1:enable 0:disable	R/W	0
---	------	--------------------	-----	---

ADC_CTL (0xA1)

Addr = 0xA1 Default=0x00

Bit(s)	Name	Description	R/W	Reset
7:5	-	-	R	0
4	DONE_PENDING	单周期转换完成中断 pending, 写 1 清 0	R/W	0
3	DONE_IRQ_EN	单周期转换完成中断使能: 1: enable 0: disable	R/W	0
2	CK_EDGE_SEL	数字采样数据沿选择: 0: posedge 1: negedge	R/W	0
1:0	AVERAGE	ADC 平均次数: (每个通道先丢弃 ADC 前 2 个数据) 11:64 10:32 01:16 00:8	R/W	00

ADCREG2H(0xFD)

Addr = 0xFD Default=0x01

Bit(s)	Name	Description	R/W	Reset
7	-	Reserved	R	00
6:5	CSGAIN<1:0>	电流放大增益选择 00: X1 01: X8 10: X16 11: X64	R/W	00
4:2	ENIPU100U<2:0>	GPIO 上拉电流使能 0: disable 1: enable 对应 ADC 使能后才能放出电流	R/W	000
1:0	-	Reserved	R/W	01

ADCREG3H(0xFE)

Addr = 0xFE Default=0x00

Bit(s)	Name	Description	R/W	Reset
7:4	-	Reserved	R/W	0000
3:0	ENCM<3:0>	ADC3/GPIO2/GPIO1/GPIO0 通路模式选择 0: 电压模式 1: 电流模式	R/W	0000

10.3.2 ADC 数据寄存器

ADC_DATL(0xA2)

Addr = 0xA2, Default_value = 0x00

Bit(s)	Name	Description	R/W	Reset
7:0	ADC0_DAT<7:0>	ADC 低 8bit	R	0x00

ADC_DATH (0xA3)

Addr = 0xA3, Default_value = 0x00

Bit(s)	Name	Description	R/W	Reset
7:4			R	0x00
3:0	ADC0_DAT<11:8>	ADC 高 4bit	R	0x00

11. Charger 模块

11.1 Charger 模块介绍

IP5904 集成一个线性充电 Charger 模块，最大充电电流 500mA, 充电电流大小 20mA/step 可调。

VIN 过压 ($V_{in} > 5.6V$) 时，硬件自动关闭 VIN 到 BAT 之间的充电电路，直到 VINOV 标志变为 0 才恢复充电。VIN 欠压阈值有 4.35V/4.5V 两挡可调，同样当 VIN 电压低于欠压阈值时，CPU 也会自动关闭充电。

当 BAT 未接电池时，直接插入 VIN，Chager 电路会进入预充电状态，BAT 端电压会上升至 CV 电平（电池充满电压），以保障此时不会触发 Batlow。

IP5904 采用 0.1 CC 涓流充电；进入涓流充电需要用软件设置 TKING_EN 为 1，然后设置 TKING 为 1。

11.2 Charger 相关寄存器

CHG_CTL0(0x8C)

Offset = 0x8C, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7	-	-	R	0
6	TKING_EN	TKING 设置使能： 0: Disable 1: Enable	R/W	0
5	TKING	TKING 设置位： 0: 退出涓流充电模式 1: 进入涓流充电模式	R/W	0
4		Reserved	R/W	0
3		Reserved	R/W	0
2	ENCHG	CHG 使能 0: disable 1: enable	R/W	0
1:0		Reserved	R/W	0

DEBUG1(0x89)

Offset = 0x89, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7	-	Reserved	R	00
6	ENPINWK	按键状态指示： 1: 按下 0: 未按下	R	0
5:2	-	Reserved	R	000
1	VINOV	VINOV 中断信号，模拟送过来的电平信号	R	0
0		Reserved	R	000

CHGREG1H(0xF3)

Offset = 0xF3, Default=0x7D

Bit(s)	Name	Description	R/W	Reset
7..3	ISET<4:0>	CC 环充电电流选择: 00000 0mA ... 01111 300mA ... 11001 500mA ... 11111 620mA Icc=ISET*20mA 20mA/step	RW	01111
2		Reserved	R/W	1
1		Reserved	R/W	0
0	-	Reserved	R/W	1

CHGREG2H(0xF4)

Offset = 0xF4, Default=0xAF

Bit(s)	Name	Description	R/W	Reset
7: 6		Reserved	R/W	10
5: 2	R_VCHG<3:0>	CV 环电压选择 0000 3.65V 1000 4.05V 0001 3.7V 1001 4.1V 0010 3.75V 1010 4.15V 0011 3.8V 1011 4.2V 0100 3.85V 1100 4.25V 0101 3.9V 1101 4.3V 0110 3.95V 1110 4.35V 0111 4V 1111 4.4V	R/W	1011
1:0	-	Reserved	R/W	0

12. 比较器模块

12.1 比较器模块介绍

IP5904 集成 1 个比较器，比较器输入 CMPP 和 CMPN 和 IO 复用，要求比较结果输出有变化时，需要产生中断，通知 MCU 做处理，用于实现过流保护的功能：

比较器的输出状态可直接通过寄存器读取，也可以产生中断请求。CMP 输出中断支持上升沿、下降沿可选择。

选用 IO3 做驱动电机时，使能模拟 CMP 功能，如果 CMP_RES 跳转为 1，说明输出有过流，会把 IO3 输出关闭，硬件触发保护，防止电流过大损坏点击，如果不使能模拟 CMP 功能，则不会触发关闭 IO3 输出功能。（注意 IO3 不管是配置成 IO 输出或者是 PWM 输出，在使能上述功能后，触发保护后都能关闭 IO3 的 OE 使能）。

在 IO 翻转时，可以通过 CHG_CTL0[7]来配置 CMP 的延时设置，防止误触发 CMP 保护功能。

12.2 比较器相关寄存器

CMU_CTL1(0xD2)

Offset = 0xD2, Default=0xFF

Bit(s)	Name	Description	R/W	Reset
7	IO3_CMP_EN	EN Bit0：根据比较器 CMP 输出结果关闭 IO3 输出功能使能开关 1 disable 因为这个寄存器复位值是 1 0 enable	R/W	1
6		Reserved	R/W	1
5	Pwm_clk_en	PWM 时钟使能：1:enable 0:disable	R/W	1
4	Timer_clk_en	Timer 时钟使能：1:enable 0:disable	R/W	1
3	Adc_clk_en	adc 时钟使能：1:enable 0:disable	R/W	1
2	Bln_clk_en	bln 时钟使能：1:enable 0:disable	R/W	1
1	Uart_clk_en	uart 时钟使能：1:enable 0:disable	R/W	1
0		Reserved	R/W	1

RMU_CTL1(0xD3)

Bit(s)	Name	Description	R/W	Reset
7	IO3_CMP_POLY	IO3_CMP 功能极性设置 POLY Bit1 1：上升沿触发保护 这个寄存器复位值是 1 0：下降沿触发保护	R/W	1
6		Reserved	R/W	1
5	Pwm_reset_en	0:reset 1:normal	R/W	1
4	Timer_reset_en	0:reset 1:normal	R/W	1
3	Adc_reset_en	0:reset 1:normal	R/W	1
2	Bln_reset_en	0:reset 1:normal	R/W	1
1	Uart_reset_en	0:reset 1:normal	R/W	1

0		Reserved	R/W	1
---	--	----------	-----	---

系统唤醒后，软件需要先复位模块再用模块功能

CMPREG (0xFF)

Offset = 0xFF, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7	engpio	比较器通路 GPIO8/9 的选择	R/W	0
6		Reserved	R/W	0
5	enadc3	比较器通路 BAT/ADC3 的选择	R/W	0
4: 3		Reserved	R/W	00
2	selcmpinv	比较器结果取反 0: 不取反 1: 取反	R/W	0
1	selgpout	比较器结果从 GPIO2 放出 0: 不放出 1: 放出	R/W	0
0	encmp	CMP 使能功能 1: 使能 0: 不使能	R/W	0

CMP_CTL(0x8F)

Offset = 0x8F, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7	CMP_CNT_CLR	CMP 上升沿或下降沿计数清 0	R/W	0
6	CMP_RES	比较器输出状态 CMP Result	R	0
5	CMPN_PENDING	比较器下降沿中断标志位。	R/W	0
4	CMPP_PENDING	比较器上升沿中断标志位。	R/W	0
3	CMPN_IRQ_EN	CMP 下降沿中断使能	R/W	0
2	CMPP_IRQ_EN	CMP 上升沿中断使能	R/W	0
1	CMPN_CNT_EN	CMP 下降沿计数使能	R/W	0

ADCREG2H(0xFD)

Addr = 0xFD Default=0x01

Bit(s)	Name	Description	R/W	Reset
7	ENGPI01	GPIO1 作为 CMP 负端输入的使能 0: disable 1: enable 应用时要注意 IO1 的 MFP 要配置成输入	R	00
6:5	CSGAIN<1:0>	电流放大增益选择 00: X1 01: X8 10: X16 11: X64	R/W	00
4:2	ENIPU100U<2:0>	GPIO 上拉电流使能 0: disable 1: enable 对应 ADC 使能后才能放出电流	R/W	000
1:0		Reserved	R/W	01

ADCREG3H(0xFE)

Addr = 0xFE Default=0x00

Bit(s)	Name	Description	R/W	Reset
7	ENCMPVR	CMP 负端参考电压输入使能 0: disable 1: enable	R/W	0000
6: 5	ENVRSEL<1:0>	CMP 负端参考电压选择 00: 0.1V 01: 0.2V 10: 0.3V 11: 0.4V		
4	BATSEL<1>	与 ADCREG0H<0>组合为 BATSEL<1:0>, 可以用作比较器的参考电压设置 00: BAT*0.97 01: BAT*0.95 10: BAT*0.93 11: BAT*0.90		
3:0	ENCM<3:0>	ADC3/GPIO2/GPIO1/GPIO0 通路模式选择 0: 电压模式, 与 1.5V 相比, 测试电压 1: 电流模式, 另一端参考为 GND, 测试电流	R/W	0000

CHG_CTL0(0x8C)

Offset = 0x8C, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7	DELAY_TIME	CMP 延迟开启时间设置: 0: 1us 1: 4us	R	0
6	TKING_EN	TKING 设置使能: 0: Disable 1: Enable	R/W	0
5	TKING	TKING 设置位: 0: 退出涓流充电模式 1: 进入涓流充电模式	R/W	0
4		Reserved	R/W	0
3		Reserved	R/W	0
2	ENCHG	CHG 使能 0: disable 1: enable	R/W	0
1:0		Reserved	R/W	0

ADCREG0H(0xFB)

Offset = 0xFB, Default=0x55

Bit(s)	Name	Description	R/W	Reset
7:1		Reserved	R/W	0x0101010
0	BATSEL<0>	与 ADCREG3H<4>组合为 BATSEL<1:0>, 可以 用作比较器的参考电压设置 00: BAT*0.97 01: BAT*0.95 10: BAT*0.93 11: BAT*0.90	R/W	0

BLN_MODE (0xA4)

Offset = 0xA4, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7	CMP_Delay_en	IO3 翻转时, 比较器延时使能功能 0:disable 1:enable	R/W	0
6:5	Low_set_bln0	设置 1000 次 pwm 中前多少次 pwm 高电平减半 Low_set[9:8]	R/W	01
4	BLN1_Mode	BLN1 亮灯模式选择 0: 与 BLN0 同步 1: 与 BLN0 互补	R/W	0
3	BLN0_Lev	BLN0 亮灯电平选择:	R/W	0

	el_Select	1: 低电平亮灯 0: 高电平亮灯		
2:0	BLNO_Mode	Bln0 模式选择 000: disable, 灭灯 001: 连续呼吸模式 010: 单次呼吸从亮状态呼吸到灭呼吸后呼吸到亮, 之后保持亮 011: 单次呼吸从灭状态呼吸到亮呼吸后呼吸到灭, 之后保持灭 100: 渐变亮, 之后保持亮 101: 渐变灭, 之后保持灭 110: 常亮 111: 常灭	R/W	000

13. PWM 模块

13.1 PWM 模块介绍

脉冲宽度调制发生器时钟由 16MHz HOSC 提供。

PWM 周期由 8 位周期寄存器 PWMP 预先设置。所有 PWM 通道共享同一个 8 位周期计数器，周期是相同的。每个 PWM 通道都有独立占空比寄存器分别为 PWM0D、PWM1D、PWM2D、PWM3D。有 4 个占空比寄存器，4 个通道都能产生独立占空比的 PWM 信号，PWM 信号的周期和占空比可通过 8 位计数器跟周期和占空比寄存器比较产生。

请按照如下初始化步骤来产生 PWM 信号。第一步，设定 PWMP 及所有 PWMnD (n=0~3) 寄存器。对 PWMRUN (PWM_CTL[0]) 置 1，开始 8 位向上计数器计数。PWM 信号开始产生，对应管脚输出 PWM 信号。PWMP 及 PWMnD 寄存器可随时被改写，但不会立即更新 PWM 周期和占空比，直到当前周期完成。用于防止产生非完整周期或占空比的 PWM 波形。

13.1.1 PWM 类型

PWM 发生器为边沿计数类型。

8 位计数器设定为单周期模式，从 00H 向上计数，直到与 PWMP 匹配，然后重新从 00H 开始向上计数。

13.1.2 工作模式

PWMn 信号通过 PWM 控制电路后，PWM 模式选择电路将产生不同类型的 PWM 输出模式，总共 4 个通道、2 组，分别是 PWM0~PWM3。支持独立输出模式，互补模式。

独立输出模式：

当 PWMMOD (PWM_CTL[1]) 设定为 0，PWM 为独立输出模式。该模式为默认输出模式。PWM0, PWM1, PWM2, PWM3 独立输出 PWM 信号。

带死区插入的互补模式：

当 PWMMOD = 1，设定为互补模式。在该模式中 PWM0/2 输出信号与独立模式下输出信号相同，但 PWM1/3 输出与 PWM0/2 输出的信号互补。同时忽略 PWM1/3 占空比寄存器 PWMnD (n:1/3)。该模式可使 PWM0/PWM1 形成一对互补的 PWM 输出。同样 PWM2/PWM3 也可以用于互补输出。

互补模式 PWM 输出需要插入“死区时间”用来防止损坏电源开关器件。IP5904 每组 PWM 共享一个 8 位“死区时间”计数器 PDTCNT，用于产生在同组中两通道之间关闭时间，对同组两个 PWM 信号中插入“关闭时间”，同样，在 PDTCNT 定时器溢出，电平从 0 到 1 转换的边沿会加入一段延迟。

注：仅当 PWM 配置为互补模式，“死区时间”控制才有效。

13.1.3 极性控制

每路 PWM 带有独立的极性控制位 PNP0~PNP3。默认正逻辑为高电平有效，即 PWM 输出高电平电源切换开，低电平电源切换关，当对应 PWM 的 PNPn 为 1 时，对应 PWM 输出会按照设定值取反输出到对应的 PWM 管脚。

13.2 PWM 相关寄存器

PWM_CTL(0xB0)

Offset = 0xB0, Default=0x00,PWM 控制寄存器

Bit(s)	Name	Description	R/W	Reset
7:5	DIV	PWM 时钟分频: 000: HCLK/1 001: HCLK /2 010: HCLK /4 011: HCLK /8 100: HCLK /16 101: HCLK /32 110: HCLK /64 111: HCLK /128	R/W	000
4	--	Reserved	R	0
3	PDT23EN	PWM2/3 组死区时间使能位。仅当 PWM2/3 配置为互补模式，死区功能才会生效。 0 = PWM2/ PWM 3 信号无延时 1 = 在 PWM 2/ PWM 3 信号上升沿加入死区时间延时	R/W	0
2	PDT01EN	PWM0/1 组死区时间使能位。仅当 PWM0/1 配置为互补模式，死区功能才会生效。 0 = PWM0/ PWM 1 信号无延时 1 = 在 PWM0 / PWM 1 信号上升沿加入死区时间延时	R/W	0
1	PWMMOD	PWM 模式选择。 0: 独立输出模式 1: 互补模式	R/W	0
0	PWMRUN	PWM 运行使能位。	R/W	0

		0 = PWM 模块空闲 1 = PWM 开始运行		
--	--	------------------------------	--	--

ENPNP(0xB1)

Offset = 0xB1, Default=0x00, PWM 通道使能和极性控制寄存器

Bit(s)	Name	Description	R/W	Reset
7	PWM3_EN	pwm3 enable	R/W	0
6	PWM2_EN	pwm2 enable	R/W	0
5	PWM1_EN	pwm1 enable	R/W	0
4	PWM0_EN	pwm0 enable	R/W	0
3	PNP3	PWM3 负极性输出使能 0 = PWM3 输出按照设定值输出到 PWM3 管脚; 1 = PWM3 输出按照设定值取反输出到 PWM3 管脚	R/W	0
2	PNP2	PWM2 负极性输出使能 0 = PWM2 输出按照设定值输出到 PWM2 管脚; 1 = PWM2 输出按照设定值取反输出到 PWM2 管脚	R/W	0
1	PNP1	PWM1 负极性输出使能 0 = PWM1 输出按照设定值输出到 PWM1 管脚; 1 = PWM1 输出按照设定值取反输出到 PWM1 管脚	R/W	0
0	PNP0	PWM0 负极性输出使能 0 = PWM0 输出按照设定值输出到 PWM0 管脚; 1 = PWM0 输出按照设定值取反输出到 PWM0 管脚	R/W	0

PMWDT(0xB2)

Offset = 0xB2, Default=0x00, PWM 死区时间设置寄存器

Bit(s)	Name	Description	R/W	Reset
7:4	PWM13_DT	PWM1 和 PWM3 上升沿延时设置 该计数器仅当 PWM 设定为互补模式, 且有效死区使能位已设置时有效。 PWM 死区时间= PWM13_DT/ DIV 16MHz	R/W	0x0
3:0	PWM02_DT	PWM0 和 PWM2 上升沿延时设置 该计数器仅当 PWM 设定为互补模式, 且有效死区使能位已设置时有效。 PWM 死区时间= PWM02_DT/ DIV 16MHz	R/W	0x0

PWMP(0xB3)

Offset = 0xB3, Default=0x00, PWM 周期设定寄存器

Bit(s)	Name	Description	R/W	Reset
--------	------	-------------	-----	-------

7: 0	PWMP	PWM 周期	R/W	0
------	------	--------	-----	---

PWM0D(0xB4)

Offset = 0xB4, Default=0x00, PWM0 占空比设定寄存器

Bit(s)	Name	Description	R/W	Reset
7: 0	PWM0D	PWM0 占空比	R/W	0

PWM1D(0xB5)

Offset = 0xB5, Default=0x00, PWM1 占空比设定寄存器

Bit(s)	Name	Description	R/W	Reset
7: 0	PWM1D	PWM1 占空比	R/W	0

PWM2D(0xB6)

Offset = 0xB6, Default=0x00, PWM2 占空比设定寄存器

Bit(s)	Name	Description	R/W	Reset
7: 0	PWM2D	PWM2 占空比	R/W	0

PWM3D(0xB7)

Offset = 0xB7, Default=0x00, PWM3 占空比设定寄存器

Bit(s)	Name	Description	R/W	Reset
7: 0	PWM3D	PWM3 占空比	R/W	0

14. BLN 模块

14.1 BLN 模块介绍

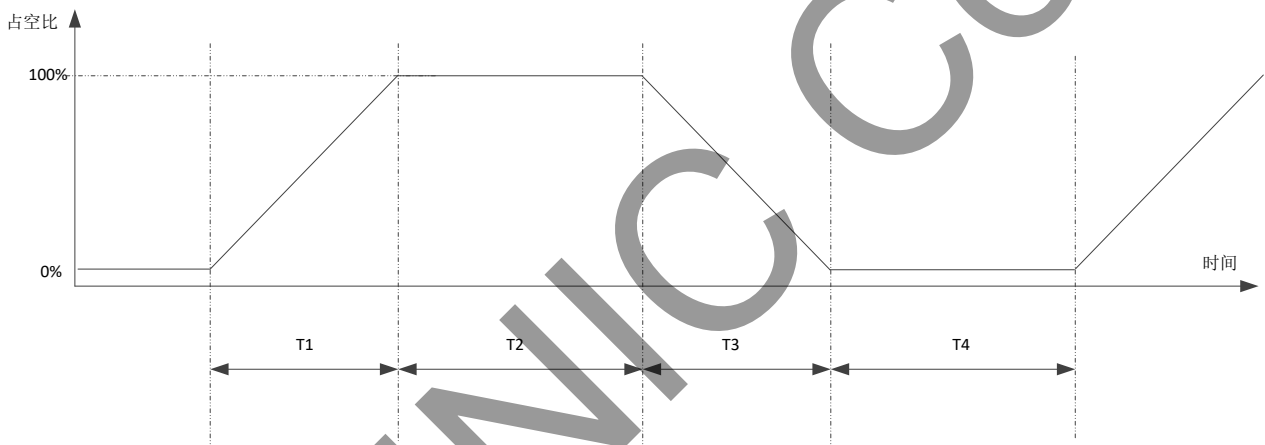
14.1.1 概述

IP5904 支持 2 路呼吸灯，BLN0 和 BLN1。其中 BLN0 是独立的，BLN1 可以选择与 BLN0 同步或者是互补输出。

呼吸灯可以配置为高电平亮灯或低电平亮灯，支持连续呼吸和单次呼吸等模式。

14.1.2 内部结构图

硬件呼吸灯的原理图如下图：



以 BLN0 为例：

- 1、T1 和 T3 阶段为呼吸灯亮度渐变的时间：通过寄存器 BLN0_CTL2[7:4]设置；
- 2、T2 阶段为呼吸灯持续常亮(100%占空比)的时间：通过寄存器 BLN0_CTL2[1:0]和 BLN0_CTL1[7:0]设置；
- 3、T4 阶段为呼吸灯持续常灭(0%占空比)的时间：通过寄存器 BLN0_CTL2[3:2]和 BLN0_CTL3[7:0]设置；

14.2 BLN 相关寄存器

BLN_MODE (0xA4)

Offset = 0xA4, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7	CMP_Delay_en	IO3 翻转时，比较器延时使能功能 0:disable 1:enable	R/W	0
6:5	Low_set_bln0	设置 1000 次 pwm 中前多少次 pwm 高电平减半 Low_set[9:8]	R/W	01
4	BLN1_Mode	BLN1 亮灯模式选择 0: 与 BLN0 同步 1: 与 BLN0 互补	R/W	0

3	BLN0_Leve l_Select	BLn0 亮灯电平选择: 1: 低电平亮灯 0: 高电平亮灯	R/W	0
2:0	BLN0_Mo de	BLn0 模式选择 000: disable, 灭灯 001: 连续呼吸模式 010: 单次呼吸从亮状态呼吸到灭呼吸后呼吸到亮, 之后保持亮 011: 单次呼吸从灭状态呼吸到亮呼吸后呼吸到灭, 之后保持灭 100: 渐变亮, 之后保持亮 101: 渐变灭, 之后保持灭 110: 常亮 111: 常灭	R/W	000

BLN0_CTL1(0xA5)

Offset = 0xA5, Default=0xff,

Bit(s)	Name	Description	R/W	Reset
7:0	On_last_time	渐变达到亮后, 持续为亮的时间。 On_last_time[7:0] (Num+1)*1000*该设置值*T_4M_clk	RW	0xff

BLN0_CTL2(0xA6)

Offset = 0xA6, Default=0x77,

Bit(s)	Name	Description	R/W	Reset
7:4	num_bln0-	呼吸灯 0 的从亮变灭或从灭变亮的时间设置: (Num+1)*1000*1000*T_4M_clk BLn0 的控制分频设置	RW	0010
3:2	Off_last_time	渐变达到灭后, 持续为灭的时间。 Off_last_time[9:8] (Num+1)*1000*该设置值*T_4M_clk	RW	11
1:0	On_last_time	渐变达到亮后, 持续为亮的时间。 On_last_time[9:8] (Num+1)*1000*该设置值*T_4M_clk	RW	11

BLN0_CTL3(0xA7)

Offset = 0xA7, Default=0xff,

Bit(s)	Name	Description	R/W	Reset
7:0	Off_last_time	渐变达到灭后, 持续为灭的时间。 Off_last_time[7:0]	RW	0xff

		(Num+1)*1000*该设置值*T_4M_clk		
--	--	----------------------------	--	--

BLN0_CTL4(0xA8)

Offset = 0xA8, Default=0x00,

Bit(s)	Name	Description	R/W	Reset
7:0	Low_set_bln0	设置 1000 次 pwm 中前多少次 pwm 高电平减半 Low_set[7:0]	RW	0x90

BLN0_CTL5(0xA9)

Offset = 0xA9, Default=0x00,

Bit(s)	Name	Description	R/W	Reset
7:0	off_set_bln0	LED 灭态的 PWM 占空比(off_set_bln0/1000)	RW	1'hA

15. UART 模块

15.1 UART 模块介绍

IP5904 支持 1 组全双工的通用异步接收发送器 UART，它采用串行收发方式与外部设备进行数据传输，可以和其它具有异步接收发送器的外部设备进行通讯。

支持 1 个独立 Uart。

支持两种工作模式

- 异步接收器
- 异步发送器

传输波特率配置

- 最低 1200 bps
- 最高 115200 bps

15.2 UART 相关寄存器

UART0_CTL0 (0xEB)

Offset = 0xEB, Default=0x00

Bit(s)	Name	Description	R/W	Reset
7	STOPSEL	STOP Select. 0: 1 stop bit 1: 2 stop bits	RW	00
6		Reserved	RW	0
5:4	DATSEL	Data Length Select. 00: 8 bits 01: 7 bits 10: 6 bits 11: 5 bits	RW	0
3	RX_IE	RX FIFO IRQ Enable When RX FIFO No Empty, IRQ=1	R/W	0
2	TX_IE	TX FIFO IRQ Enable When TX FIFO Empty, IRQ=1	R/W	0
1	RX_EN	UART RX enable 0: disable 1: enable	RW	0
0	TX_EN	UART TX enable 0: disable 1: enable	RW	0

UART0_CTL1(0xEC)

Offset = 0xEC, Default=0x03,

Bit(s)	Name	Description	R/W	Reset
7:6	PSEL	Parity Select 00: no parity 01: no parity 10: Odd 11: Even	RW	00
5	TXRX_LB	UART TX/RX loopback enable 0: disable 1: enable	RW	0
4	-			
3	MLSB	MSB/LSB 选择: 0: LSB 1: MSB	RW	0
2:0	BAUDSET	波特率选择: 000: 1200 bps 001: 2400 bps 010: 4800 bps 011: 9600 bps 100: 19200 bps 101: 38400 bps 110: 14400 bps 111: 115200 bps	RW	011

UART0_STATE(0xED)

Offset = 0xED, Default=0x50,

Bit(s)	Name	Description	R/W	Reset
7	TXFULL	TX FIFO Full. 1: Full 0: No Full	R	0
6	TXEMP	TX FIFO Empty. 1: Empty 0: No Empty	R	1
5	RXFULL	RX FIFO full 0: no full 1: full	R	0
4	RXEMP	RX FIFO Empty. 1: Empty 0: No Empty	R	1
3	UART_BUSY	UART transmitting status bit. The bit is automatically clear when uart finish	R	0

		transmitted data; and set in transmitting status automatically; 0: uart idle status 1: uart transmitting status		
2	UART_ERR	UART received error bit This bit is set to 1 when RX data disobey UART parity checkout or pulse length is less than half baud_set and stopbit_error. Reset rx_fifo or disable uart_rx will clear this bit.	RW	0
1	TXFIFOERR	TX FIFO Error. 0: No Error 1: Error This bit set when UART TX FIFO is written overflow; Writing 1 to the bit will clear the bit and reset the TX FIFO.	RW	0
0	RXFIFOERR	RX FIFO Error. 0: No Error 1: Error This bit set when UART RX FIFO is written overflow; Writing 1 to the bit will clear the bit and reset the RX FIFO.	RW	0

UART0_TXDAT (0xEE)

Offset = 0xEE, Default=0x00,

Bit(s)	Name	Description	R/W	Reset
7:0	TXDAT	Transfer Data. The depth of FIFO is 8bit×8 levels	W	0

UART0_RXDAT (0xEF)

Offset = 0xEF, Default=0x00,

Bit(s)	Name	Description	R/W	Reset
7:0	RXDAT	Received Data. The depth of FIFO is 8bit×8 levels	R	0

从 Rx buffer 中读数据，需要用 idata 变量缓存一下，不能直接用 data 变量缓存，否则可能会出现时序错误。

16. 典型应用原理图

IP5904 只需要电容、电阻，即可实现完整功能的充电方案。

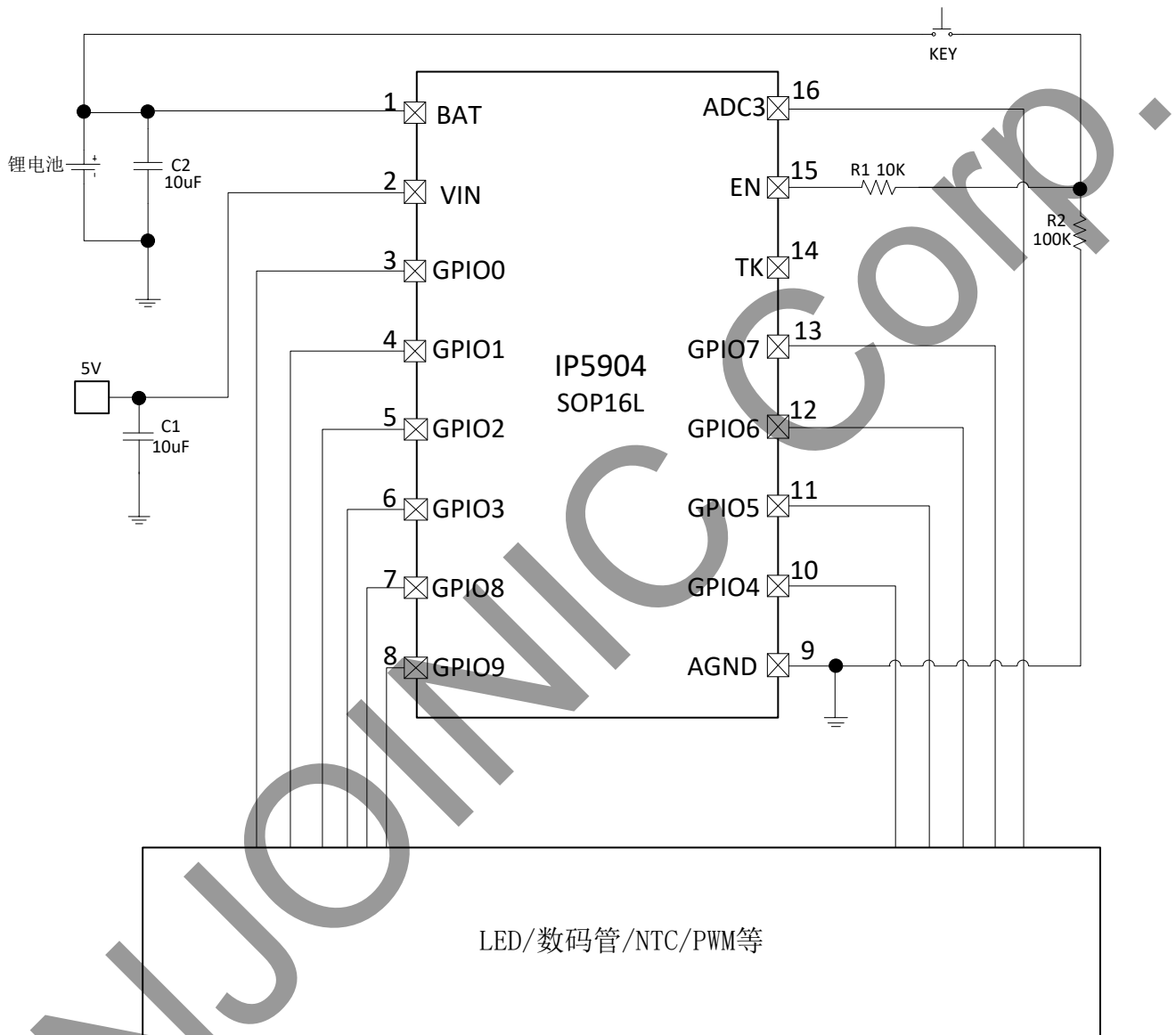


图 4 IP5904 典型应用原理图

17. 封装信息

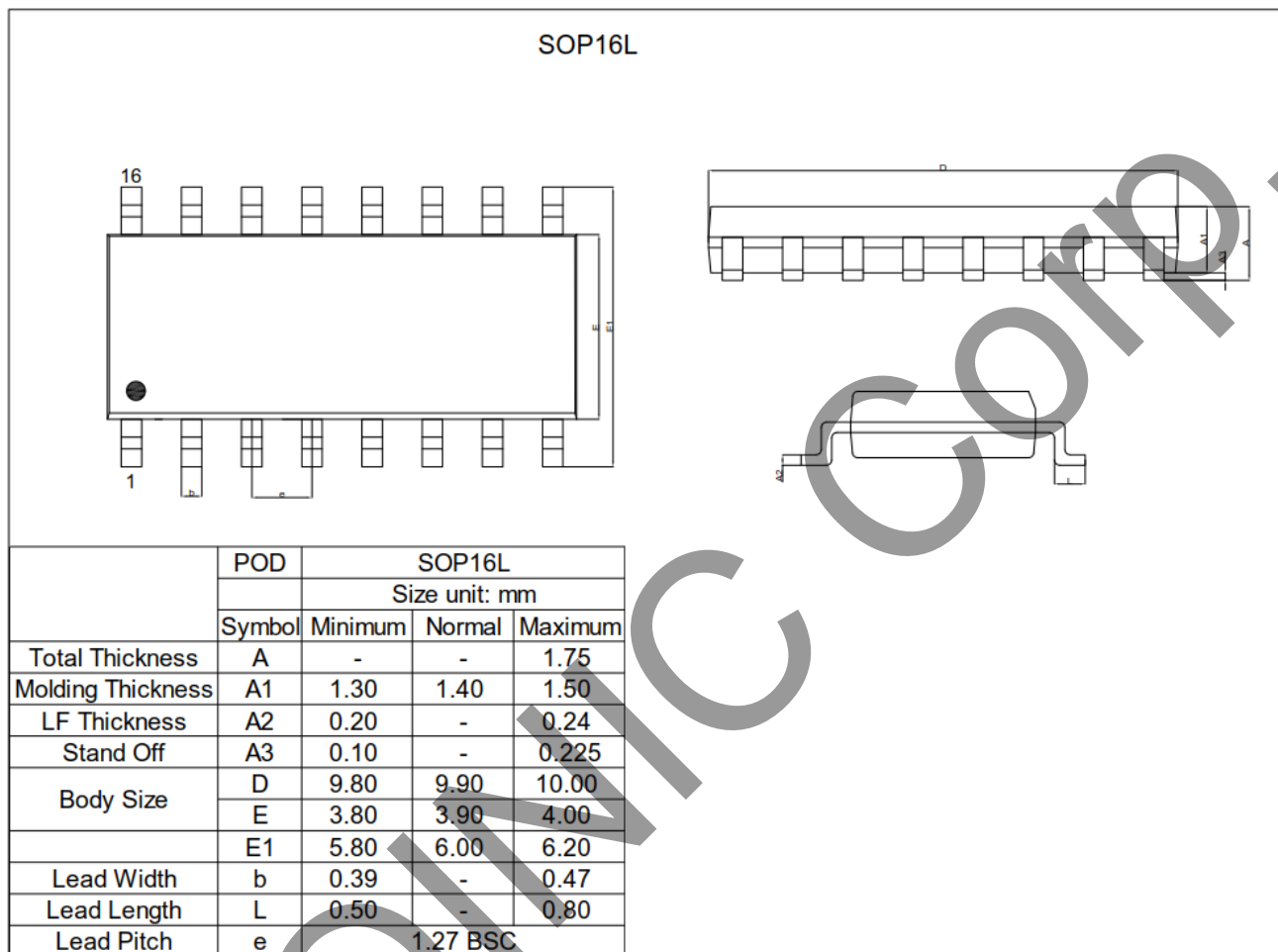


图 5 IP5904 封装信息

18.IC 丝印说明



说明：


- 1、 ——英集芯标志
- 2、IP5904 ——产品型号
- 3、XXXXXXXX ——生产批号
- 4、○ ——Pin1脚位置

图 6 IP5904 IC 丝印说明

19. 责任及版权声明

英集芯科技有限公司有权对所提供的产品和服务进行更正、修改、增强、改进或其它更改，客户在下订单前应获取最新的相关信息，并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的销售条款与条件。

英集芯科技有限公司对应用帮助或客户产品设计不承担任何义务。客户应对其使用英集芯的产品和应用自行负责。为尽量减小与客户产品和应用相关的风险，客户应提供充分的设计与操作安全验证。

客户认可并同意，尽管任何应用相关信息或支持仍可能由英集芯提供，但他们将独力负责满足与其产品及其应用中使用英集芯产品相关的所有法律、法规和安全相关要求。客户声明并同意，他们具备制定与实施安全措施所需的全部专业技术和知识，可预见故障的危险后果、监测故障及其后果、降低有可能造成人身伤害的故障的发生机率并采取适当的补救措施。客户将全额赔偿因在此类关键应用中使用任何英集芯产品而对英集芯及其代理造成的任何损失。

对于英集芯的产品手册或数据表，仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况下才允许进行复制。英集芯对此类篡改过的文件不承担任何责任或义务。复制第三方的信息可能需要服从额外的限制条件。

英集芯会不定期更新本文档内容，产品实际参数可能因型号或者其他事项不同有所差异，本文档不作为任何明示或暗示的担保或授权。

在转售英集芯产品时，如果对该产品参数的陈述与英集芯标明的参数相比存在差异或虚假成分，则会失去相关英集芯产品的所有明示或暗示授权，且这是不正当的、欺诈性商业行为。英集芯对任何此类虚假陈述均不承担任何责任或义务。