

## IP5312 寄存器文档

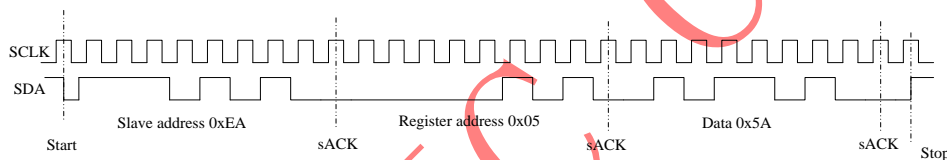
### 1、I2C 协议

The I2C speed support 400Kbps. Support 8 bit address width and 8bit data width. Transmit and receive MSB first. The default slave address is 0xEA.

I2C acts as slave and is controlled by the master. The SCK line of the I2C interface is driven by the master. The SDA line could be pulled up to VCC by a 2.2Kohm resistor and pulled

down by either the master or the slave. A typical WRITE sequence for writing 8bits data to a register is shown in below figure. A start bit is given by the master, followed by the slave address, register address and 8-bit data. After each 8-bit address or data transfer, the IP5312 gives an ACK bit. The master stops writing by sending a stop bit. All 8 bits data must be written before the register is updated.

Example: Write 8bit data 0x5A to register 0x05, and the slave address is 0xEA



Note: Sack generated by Slave, Mack generated by Master, and Mnack is a NACK generated by Master

Figure1 I2C WRITE

A typical READ sequence is shown in below figure. First the master has to write the slave address, followed by the register address. Then a restart bit and the slave address specify that a READ is generated. The master then clocks out 8 bits at a time to read data.

Example: Read 8bit data 0x5A from register 0x05, and the slave address is 0xEA

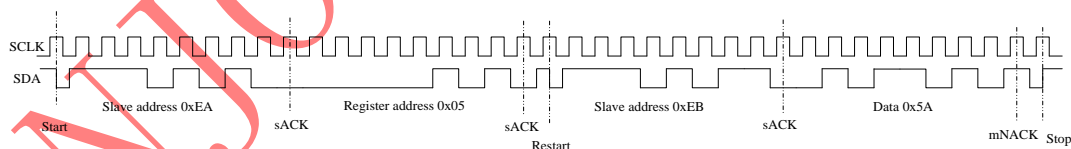
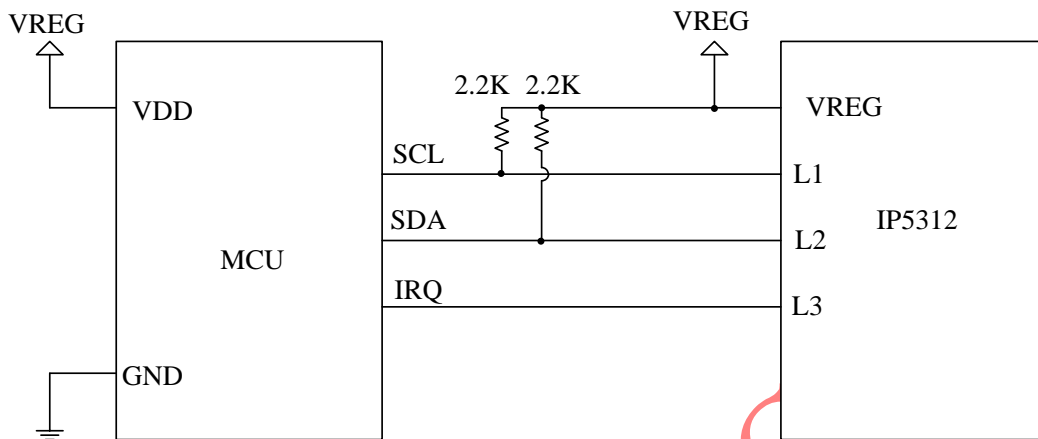


Figure 2 I2C Read

## 2、I2C 应用注意事项



MCU需要从IC VREG取电

IC工作时，IRQ为高电平  
IC休眠时，IRO为高阻状态

I2C 连接示意图

1、IP5312 标准品默认支持 I2C，不需要单独定制 I2C 版本；

2、IP5312 从休眠状态转入工作状态（按键、负载接入、5V 充电接入）时，IP5312 内部首先会检测 L1、L2 脚的是否被上拉到 3.1V（VREG），如果 L1 L2 同时被上拉到 3.1V 则进入 I2C 模式，L3 输出一个 3.1V 的高电平；如果没有检测到 L1 L2 同时上拉则进入 LED 灯显模式，每次从休眠进入工作状态都会进行检测；

3、由于 IP5312 由休眠进入工作状态时会进行 I2C 检测，所以 MCU 在休眠的时候需要将 SDA 和 SCK 配置为输入或者高阻状态，直到检测到 INT 为高时才开始读写 I2C 数据，否则会导致 IC 在由休眠进入工作状态时检测到 L1 或者 L2 没有被上拉而无法进入 I2C 状态

4、由于 IP5312 由休眠进入工作状态时会进行 I2C 检测以及 IP5312 内部的数字电平都是 3.1V 所以 MCU 供电必须有 VREG 供电，如果 MCU 用外部的 LDO 供电，当 BAT 没电或者小于 2V 时 VIN 接入 5V 给 IP5312 供电，VREG 有电系统会进行 I2C 检测，但是 MCU 没有电，SDA 和 SCK 的状态不定，可能导致 L1 和 L2 没检测到上拉无法进入 I2C 模式；

5、如果要修改 IP5312 某个寄存器的时候需要先将相应寄存器的值读出来对需要修改的 BIT 位进行与或运算后再把计算的值写进这个寄存器，确保只修改需要修改的 bit 其他 bit 的值不能随意改动

## 6、I2C 通讯波形介绍

I2c master 写的时候，先传 8bit 数据，第 9 个 bit 读 slave 返回的 ack，ack 为低代表写入成功，为高代表写入不成功。

I2c master 读的时候，最后一个 byte 传输是 slave 返回数据，master 返回 nack（高电平），代表读结束；如果 master 返回的是 ack（低电平），则说明读没有结束，master 会继续读。

所以第九个 bit 的 ack 信号要看 master 端是读操作还是写操作：

因为 IP5312 只能做 slave：

如果往 IP5312 寄存器写入数据，IP5312 返回 ack 为低电平；

如果从 IP5312 读取数据，IP5312 返回 nack 高电平），（master 必须发 NACK，否则会有异常）代表读结束

## 3、寄存器功能描述

标示为“Reserved”的寄存器位有特殊控制作用，不可改变原有的值，否则会出现无法预期的结果。对寄存器的操作必须按照“读-->修改-->写”来进行，只修改要用到的 bit，不能修改其他未用 bit 的值。

## SYS\_CTL0

寄存器地址= 0x01

Bit(s)	Name	Description	R/W	Reset
7	EN_CHG2BST	充电器拔出后, 是否自动开启 BOOST 0: 不开启 1: 开启	R/W	1
6		Reserved		
5	EN_IRQ	IRQ 使能 (I2C 时 LED3 为 IRQ 功能) 0: disable 1: Enable	R/W	1
4		Reserved		
3	EN_LIGHT	LIGHT 使能 0: disable 1: Enable	R/W	1
2	EN_BST	Boost 使能 0: disable 1: Enable	R/W	1
1	EN_CHG	充电使能 0: disable 1: Enable	R/W	1
0		Reserved		

## SYS\_CTL1

寄存器地址= 0x03

Bit(s)	Name	Description	R/W	Reset
7: 6		Reserved		
5	EN_ISYSLOW	Boost 输出轻载关机使能 0: disable 1: Enable	R/W	1
4: 0		Reserved		

## ADC\_DIG\_CTL6

寄存器地址= 0xC9

Bit(s)	Name	Description	R/W	Reset
7:6		Reserved		
5:0	SET_ISYSLOW	Boost 输出轻载关机电流阈值 $\approx \text{data} * 4.3\text{mA}$ 这里设置的是总的输出电流。不建议关机电流设的太	R/W	01 0000

		小，否则一旦 ADC 被干扰可能存在无法自动关机的情况。		
--	--	------------------------------	--	--

## NTC\_CTL2

寄存器地址= 0x04

Bit(s)	Name	Description	R/W	Reset
7:5		Reserved		
4	EN_NTC_MID	NTC 中间档是否要使 ISET 减半 0: 不操作 1: 减半	R/W	1
3	EN_NTC_CHG	NTC 高低温关 Charger 使能	R/W	1
2	EN_NTC_BST	NTC 高温关 BOOST 使能	R/W	1
1	EN_NTC_SC	NTC 短接到地时，是否关闭 NTC 功能 1: 关闭 0: 不关闭	R/W	1
0	EN_NTC	NTC 使能	R/W	1

## NTC\_CTL3

寄存器地址= 0x19

Bit(s)	Name	Description	R/W	Reset
7		Reserved		
6	EN_NTC_BST	NTC 低温关 BOOST 使能 0: disable 1: Enable	R/W	1
5: 0		Reserved	R/W	0

## SYS\_CTL4

寄存器地址= 0x05

Bit(s)	Name	Description	R/W	Reset
7	Force_reset		R/W	0
6	Force_light		R/W	0
5	Force_boost		R/W	0

4			R/W	0
3	En_force_restart	先将 bit3 写 1, 再将 bit7 写 1, 可内部触发芯片复位	R/W	0
2	En_force_light	先将 bit2 写 1, 再将 bit6 写 1, 可内部触发开启 LIGHT; 开启之后, 先将 bit6 写 0, 再将 bit2 写 0, 可关闭 LIGHT	R/W	0
1	En_force_boost	先将 bit1 写 1, 再将 bit5 写 1, 可内部触发开启 BOOST 开启之后, 可以将这两个 bit 都写 0, 方便下一次开启。只能将 0x01 寄存器 bit2 写 0 才能强制关闭 BOOST, 否则只能等 BOOST 自动关闭	R/W	0
0			R/W	0

## ONOFF\_CTL1

寄存器地址= 0x06

Bit(s)	Name	Description	R/W	Reset
7:6	Set_tmp_time_vout	轻载自动关机时间设置 00: 8s 01: 16s 10: 32s 11: 63s	R/W	10
5:2		Reserved		
1:0	Onoff_timeset	Onoff long time set (onoff 长按键时间设定) 00:1s 01:2s 10:3s 11: 4s	R/W	01

## ONOFF\_CTL2

寄存器地址= 0x07

Bit(s)	Name	Description	R/W	Reset
7:2		Reserved		
1	En_Onoff_dn	Onoff 按键关机使能: 0: 关闭关机功能 1: 使能关机功能	R/W	1
0		Reserved		

## ONOFF\_CTL3

寄存器地址= 0x08

Bit(s)	Name	Description	R/W	Reset
7:4		Reserved		
3	Set_Light_sw	设置开关照明灯的方式: 0: Onoff 长按 1: Onoff 双击 (bit3 和 bit2 不建议设成相同的方式, 否则结果可能错乱)	R/W	0
2	Set_Onoff_dn	设置 Onoff 按键关机方式: 0 Onoff 双击 1: Onoff 长按 (bit3 和 bit2 不建议设成相同的方式, 否则结果可能错乱)	R/W	0
1:0		Reserved		

## SYS\_CTL5

寄存器地址= 0x09

Bit(s)	Name	Description	R/W	Reset
7		Reserved		
6	RESET_Config	每次进入工作状态, 是否复位 GPIO 配置 1: 复位 GPIO 配置 0: 不复位 GPIO 配置 如果要保持 GPIO 的配置在关机、开机都不变, 需要将该 bit 写 0;	R/W	1
5:0		Reserved		

## SYS\_CTL6

寄存器地址= 0x0A

Bit(s)	Name	Description	R/W	Reset
7:5		Reserved		
4	CHG_OUT	1: 支持同充同放 0: 不支持同充同放 当开启该功能时, 若方案上还外加了输入快充协议 IC, 为了防止进入同充同放功能时, 协议 IC 同时又向适配器请求高压 (这样有可能会烧坏普通	R/W	0

		手机), 那么在开启该功能时, 必须将输入快充协议 IC 关闭。 方法 1: 0x5A 配置为 0x08, 0x5B 配置为 0xC1, 并且将 RSET 引脚连接到输入快充协议 IC, 作为协议 IC 的使能。充电时为 0, 放电时为 1。 方法 2: 0x5A 配置为 0x02, 0x5B 配置为 0xC1, 并且将 L4 引脚连接到输入快充协议 IC, 作为协议 IC 的使能。单充电时为 0, 放电和边充边放时为 1。 方法 3: 单片机用 IO 控制直接关闭输入快充协议 IC。		
--	--	--	--	--

## SYS\_CTL7

寄存器地址 = 0x0B

Bit(s)	Name	Description	R/W	Reset
7	VOUTG_CHG	1: 充电状态时, VOUTG 信号由 (0x0B) bit4 寄存器控制 0: 充电状态时, VOUTG 信号由系统自动控制 在放电状态, VOUTG 始终是由系统自动控制的	R/W	0
6: 5		Reserved		
4	VOUTG	VOUTG 寄存器控制外部 PMOS。在充电状态且 bit7=1 时有效 0: 关闭 1: 开启		0
3:0		Reserved		

## SYS\_CTL8

寄存器地址 = 0x0C

Bit(s)	Name	Description	R/W	Reset
7:1		Reserved		
0	VOUT_DT	VOUT 口手机插入自动开机功能使能: 0: disable 1: Enable	R/W	1



## SYS\_CTL9

寄存器地址 = 0x0D

Bit(s)	Name	Description	R/W	Reset
7	EN_STD_IIC	待机时保持 IIC 通信使能	R/W	0
6	EN_STD_CLK	待机时保持 IIC 时钟使能 ——该 bit 使能时，为了降低待机功耗，进入待机后 IIC 时钟会自动切换到低频时钟，此时 IIC 的访问时钟需低于 10k。待机/工作状态可以通过 L3（INT）引脚进行区分。	R/W	0
5:3		Reserved		
2		使能 VOUT2 通路 1: 使能 VOUT2（普通输出口） 0: 关闭 VOUT2（普通输出口） 如果需要双 A 口输出，当普通输出口插入负载后，快充输出口也只输出 5V；当普通输出口负载拔出后(在输出需要加采样电阻通过 MCU 的 ADC 采样电流小于 100mA 即认为负载已经拔出)，MCU 可以将该 bit 写 0，并且关闭 Boost_en（REG_0x01 bit2），再将该 bit 写 1，Boost_en 开启，就可以自动恢复快充输出	R/W	1
1:0		Reserved		

## SYS\_CTL10

寄存器地址 = 0x0E

Bit(s)	Name	Description	R/W	Reset
7:2		Reserved		
1:0		有充电器插入后，延时开启充电时间设定 00:0s 01:1s 10:2s 11:4s	R/W	0

## SYS\_CTL11

寄存器地址 = 0x14

Bit(s)	Name	Description	R/W	Reset
7:6		Reserved		
3		QC3.0 Enable	RW	
2		Reserved		
1		QC2.0 Enable	RW	
0		DCP Enable	RW	

## BAT\_LOW

寄存器地址 = 0x13

Bit(s)	Name	Description	R/W	Reset
7		Reserved		
6	VBUS 过压	1:14V 0: 5.6V		
5:4	BATLOW_SET	电池低电关机电压设置（BAT 引脚电压，有迟滞） 00: 2.70V↔2.76V 01: 2.76V↔2.84V 10: 2.92V↔3.00V 11: 3.00V↔3.11V	R/W	10
3:0				

## Charger\_CTL0

寄存器地址 = 0x22

Bit(s)	Name	Description	R/W	Reset
7:4		Reserved		
3:2		电池类型选择 11: 4.4v 10: 4.35v 01: 4.3v 00: 4.2v	RW	00
1:0		恒压充电电压加压设置 11: 加压 42mV 10: 加压 28mV 01: 加压 14mV		10

	00: 不加压		
--	---------	--	--

4.30V/4.35V 建议加压 14mV；4.2V 建议加压 28mV；如果客户评估恒压充电电压太高，可以考虑 4.3V/4.35V+48mV 的参数进行配置，  
 例如, 4.35V 配置：

电芯选择 4.3V 配置，恒压充电加压选择 48mV，这时候理论停充电压为 4.35V；  
 如果客户测试发现停充电流太大或者电池充不饱，可以在读到停充标志位置 1 后，读取电池电压是否达到预期的电压值，如果比预期电压低就重新开关充电使能（0X01 bit1）后会继续充电 4 分钟左右就会重新将充满标志位置 1，这时候再来判断电池电压是否达到预期电压值，如果没有充饱就继续上述动作；同时开关充电使能的次数需要做限制，避免恒压充电电压始终比阈值设定电压低导致无法停充的情况。

## Charger\_CTL1

寄存器地址 = 0x29

Bit(s)	Name	Description	R/W	Reset
7		Reserved		
6	BATSET_SEL	电池类型内部寄存器设定还是外部 Vset PIN 设定选择 1: 外部 VSET PIN 设置 0: 内部寄存器设置	RW	1
5: 0		Reserved	RW	

## Charger\_CTL2

寄存器地址 = 0x25

Bit(s)	Name	Description	R/W	Reset
7:6		Reserved		
5:3	R_VIL7	VIN 欠压环阈值@7V: 111:6.7 110: 6.64 101: 6.59 100: 6.53 011: 6.47 010: 6.42 001:6.36 000: 6.25	RW	100
2:0	R_VIL5	VIN 欠压环阈值@5V: 111:4.92 110: 4.88 101: 4.84 100: 4.8	RW	100

		011: 4.76 010: 4.72 001: 4.64 000: 4.58		
--	--	--	--	--

## Charger\_CTL3

寄存器地址 = 0x27

Bit(s)	Name	Description	R/W	Reset
7:6		Reserved		
5:3	R_VIL12	VIN 欠压环阈值@12V: 111: 11.8 110: 11.7 101: 11.6 100: 11.5 011: 11.4 010: 11.3 001: 11.2 000: 11	RW	100
2:0	R_VIL9	VIN 欠压环阈值@9V: 111: 8.8 110: 8.73 101: 8.65 100: 8.58 011: 8.5 010: 8.43 001: 8.35 000: 8.2	RW	100

## CHG\_DIG\_CTL0

寄存器地址 = 0x28

Bit(s)	Name	Description	R/W	Reset
7:5		Reserved		
4:0	CHG_IS_5	输入 5V 时的充电电流设置: $0.2+0.2*data$ (单位 A)	RW	01010

寄存器地址 = 0x51

Bit(s)	Name	Description	R/W	Reset
7:3	CHG_IS_5	输入 5V 时的充电电流设置: $0.2+0.2*data$ (单位 A)		
3:0		Reserved	RW	01010

注意:

在设定 5V 充电电流时，需要同时操作 0X28(bit4:0)和 0X51(bit7:3)，写入的对应 bit 的值要求一样

V1.2 版本更新

## CHG\_DIG\_CTL1

寄存器地址 = 0x2B

Bit(s)	Name	Description	R/W	Reset
7:5		Reserved		
4:0	CHG_IS_5	输入>5V 时的充电电流设置：0.2+0.2*data（单位 A）	RW	10011

## BST\_ISET

寄存器地址 = 0x30

Bit(s)	Name	Description	R/W	Reset
7:6		Reserved		
5:0	BST_5V_ISET	非快充模式下 BOOST 限流设置。约 50mA 一个 step。默认值不固定（默认 3A 左右），需要改变限流时，要先把寄存器值读回去然后再进行加减操作。	RW	x

## BST\_VSET

寄存器地址 = 0xB2

Bit(s)	Name	Description	R/W	Reset
7:5		Reserved		
4:0	BST_5V_VSET	非快充模式下 BOOST 电压设置。0.0667V 一个 step。默认值不固定（默认值 5.1V 左右），需要改变电压时，要先把寄存器值读回去然后再进行加减操作。	RW	x

## MTK\_CTL0

寄存器地址 = 0x45

Bit(s)	Name	Description	R/W	Reset
7:1		Reserved		
0		MTK 输出使能 0: disable 1: Enable	RW	1

## MTK\_CTL1

寄存器地址 = 0x45

Bit(s)	Name	Description	R/W	Reset
7:5		Reserved		
4		MTK 输出使能 0: disable 1: Enable	RW	1
3:0		Reserved		

## ADC\_DIG\_CTL1

寄存器地址 = 0xC4

Bit(s)	Name	Description	R/W	Reset
7:4	set_bat_imp	RSET 寄存器设定 12.5mohm* set_bat_imp	RW	0110
3:2		RSET 电池寄存器地址 00: 0 mohm 01: 12.5 mohm 10: 25 mohm 11: 50mohm 最后设定电阻会加上该寄存器地址值	RW	11
1		RSET 电阻是内部寄存器设定还是外部电阻设定 1: 外部电阻设定 0: 内部寄存器设定	RW	1
0		Reserved		

## IOPIN\_CTL

寄存器地址 = 0x5B

Bit(s)	Name	Description	R/W	Reset
7:6	reserved	reserved		
5:0		00_0001: RSET 作为 Boost_EN 信号输出,当 Boost 使能时, RSET 输出 1, 否则输出 0。(需要该功能时, 需要将 GPIO3 的设定信号使能开启, 即 reg_0X5A 的 bit3=1, 同时 RSET 功能选择要选择 GPIO3 功能, 不能选择 RSET 功能(reg_0X52 的 bit[7:6]=01), RSET 功能要选择通过寄存器控制) Others: reserved	RW	0

## IOPIN\_EN

寄存器地址 = 0x5A

Bit(s)	Name	Description	R/W	Reset
7:0		GPIO[7:0]放出设定信号使能 1: enable 0: disable 当 GPIO 引脚不需要输出 0x5B 寄存器设定的功能时，需要将该寄存器对应的 GPIO bit 写 0。常用的 GPIO 有 GPIO1、GPIO2、GPIO3、GPIO4	RW	0

## MFP\_CTL0

寄存器地址 = 0x52

Bit(s)	Name	Description	R/W	Reset
7	NTC_sel	NTC的功能选择 0: NTC 1: GPIO5	R/W	0
6:5	VSET_sel	VSET 的功能选择 00: VSET 01: GPIO4 10: ADC (输入电压范围0-3v)，数据在GPIADC中间 11: Reserved	R/W	00
4		Reserved		
3	RSET_sel	RSET 功能选择 0: RSET 1: GPIO3	R/W	0
2	LIGHT_sel	LIGHT的功能选择 0: LIGHT 1: GPIO2	R/W	0
1	L4_sel	LED4的功能选择 0: L4 1: GPIO1	R/W	0
0	L3_sel	L3的功能选择 0: L3 1: GPIO0	R/W	0

## GPIO\_IEN0

寄存器地址 = 0x54

Bit(s)	Name	Description	R/W	Reset
7:6	reserved	Reserved		
5:0	GPIO_IEN	GPIO[5:0]input enable 0: disable 1: Enable	RW	0

## GPIO\_OEN0

寄存器地址 = 0x56

Bit(s)	Name	Description	R/W	Reset
7:6	reserved	Reserved		
5:0	GPIO_OEN	GPIO[5:0]output enable 0: disable 1: Enable 在 GPIO 做输出时，需要先将对应的 GPIO_DATA 先配置为对应的值，再开启 GPIO 输出使能	RW	0

## GPIO\_DATA0

寄存器地址 = 0x58

Bit(s)	Name	Description	R/W	Reset
7:6	reserved	Reserved		
5:0	GPIO_DATA	GPIO[5:0] data 0: disable 1: Enable	RW	0

## ADC CTL2

寄存器地址 = 0xC2

Bit(s)	Name	Description	R/W	Reset
7	CHOPEN	CHOPPER 使能	R/W	1
6	VSYS_EN	Boost output voltage ADC 使能	R/W	1
5	ISYS_EN	Boost output current ADC 使能	R/W	1
4	VGPI_EN	GPI 电压 ADC 使能	R/W	0
3	VTS_EN	Temperature Sensor ADC 使能	R/W	0



2	EXTRSET_EN	EXTRSET ADC 使能	R/W	1
1	IBAT_EN	IBAT ADC 使能	R/W	1
0	VBAT_EN	VBAT ADC 使能	R/W	1

## BATVADC\_DAT0

寄存器地址 = 0xD0

Bit(s)	Name	Description	R/W	Reset
7:0	BATVADC[7: 0]	BATVADC 数据的低 8bit	R	X

## BATVADC\_DAT1

寄存器地址 = 0xD1

Bit(s)	Name	Description	R/W	Reset
7:6			-	-
5:0	BATVADC[13: 8]	BATVADC 数据的高 6bit $VBAT = BATVADC * 0.26855mV + 2.6V$ 需要将寄存器 0xD0 和 0xD1 (需将寄存器 0xD1 的 bit[7:6] 丢掉) 的数据组合为 14bit 数据来计算，	R	X

## BATIADC\_DAT0

寄存器地址 = 0xD2

Bit(s)	Name	Description	R/W	Reset
7:0	BATIADC[7: 0]	BATIADC 数据的低 8bit	R	X

## BATIADC\_DAT1

寄存器地址 = 0xD3

Bit(s)	Name	Description	R/W	Reset
7:6			-	-
5:0	BATIADC[13: 8]	BATIADC 数据的高 6bit $IBAT = BATVADC * 2.68554mA$ (补码格式, 充电为正, 放电为负) 例如: 000000_00000001 表示 1, 即充电电流 2.68554mA	R	X

		111111_11111111 表示-1, 即放电电流 2.68554mA 需要将寄存器 0XD2 和 0XD3 (需将寄存器 0XD3 的 bit[7:6]丢掉) 的数据组合为 14bit 数据来计算		
--	--	--	--	--

## SYSVADC\_DAT0

寄存器地址 = 0xD4

Bit(s)	Name	Description	R/W	Reset
7:0	SYSVADC[7: 0]	SYSVADC 数据的低 8bit	R	X

## SYSVADC\_DAT1

寄存器地址 = 0xD5

Bit(s)	Name	Description	R/W	Reset
7:6			-	-
5:0	SYSVADC [13: 8]	SYSVADC 数据的高 6bit $SYSV = 15600 - (0x4000 - SYSVADC) * 1.61133$ 需要将寄存器 0XD4 和 0XD5 (需将寄存器 0XD5 的 bit[7:6]丢掉) 的数据组合为 14bit 数据来计算	R	X

## SYSIADC\_DAT0

寄存器地址 = 0xD6

Bit(s)	Name	Description	R/W	Reset
7:0	SYSIADC [7: 0]	SYSIADC 数据的低 8bit	R	X

## SYSIADC\_DAT1

寄存器地址 = 0xD7

Bit(s)	Name	Description	R/W	Reset
7:6		Reserved	-	-
5:0	SYSIADC [13: 8]	SYSIADC 数据的高 6bit $SYSI = SYSIADC * 1.0742mA$ （补码格式，放电为正，充电为负） 例如：000000_00000001 表示 1，即放电电流 1.0742mA 111111_11111111 表示-1，即充电电流 1.0742mA 需要将寄存器 0XD6 和 0XD7（需将寄存器 0XD7 的 bit[7:6]丢掉）的数据组合为 14bit 数据来计算	R	X

## RSETADC\_DAT0

寄存器地址 = 0xD8

Bit(s)	Name	Description	R/W	Reset
7:0	RSETADC [7: 0]	RSETADC 数据的低 8bit	R	X

## RSETADC\_DAT1

寄存器地址 = 0xD9

Bit(s)	Name	Description	R/W	Reset
7:6		Reserved	-	-
5:0	RSETADC [13: 8]	RSETADC 数据的高 6bit $V\_RSET = 1.5V + RSETADC * 0.26855mV$ （补码格式，高于 1.5v 为正，低于 1.5v 为负） 例如：000000_00000001 表示 1，即 1.5V+0.26855mV 111111_11111111 表示-1，即 1.5V-0.26855mV 需要将寄存器 0XD8 和 0XD9（需将寄存器 0XD9 的 bit[7:6]丢掉）的数据组合为 14bit 数据来计算	R	X

## GPIADC\_DAT0

寄存器地址 = 0xDC

Bit(s)	Name	Description	R/W	Reset
7:0	GPIADC [7: 0]	GPIADC 数据的低 8bit	R	X

## GPIADC\_DAT1

寄存器地址 = 0xDD

Bit(s)	Name	Description	R/W	Reset
7:6		Reserved	-	-
5:0	GPIADC [13: 8]	GPIADC 数据的高 6bit $GPI = GPIADC * 0.26855mv + 1.5V$ （补码格式，高于 1.5v 为正，低于 1.5v 为负） 例如：000000_00000001 表示 1，即 $1.5V + 0.26855mv$ 111111_11111111 表示 -1，即 $1.5V - 0.26855mv$ 需要将寄存器 0xDC 和 0xDD（需将寄存器 0xDD 的 bit[7:6] 丢掉）的数据组合为 14bit 数据来计算	R	X

## BATOCV\_DAT0

寄存器地址 = 0xDE

Bit(s)	Name	Description	R/W	Reset
7:0	BATOCV [7: 0]	BATOCV 数据的低 8bit	R	X

## BATOCV\_DAT1

寄存器地址 = 0xDF

Bit(s)	Name	Description	R/W	Reset
7:6		Reserved	-	-
5:0	BATOCV [13: 8]	BATOCV 数据的高 6bit $OCV = BATOCV * 0.26855mv + 2.6V$ BATOCV 电压是根据当前电池电压、电流来估算的	R	X

		BATOCV 电压, 其和当前 BAT 电压不是一一对应的关系。 充电时: $BATOCV = VBAT - IBAT * r$ ; 放电时 $BATOCV = VBAT + IBAT * r$ ; r 为电池等效内阻 需要将寄存器 0XDE 和 0XDF (需将寄存器 0XDF 的 bit[7:6] 丢掉) 的数据组合为 14bit 数据来计算		
--	--	--	--	--

## SYS\_STATUS

寄存器地址 = 0x71

Bit(s)	Name	Description	R/W	Reset
7	Boost_en	当前状态指示 0: 非放电状态 1: 放电状态	R	X
6	Charger_en	当前状态指示 0: 非充电状态 1: 充电状态	R	X
5:4		Reserved	R	X
3:2	Key_State	当前状态指示 00: 非充电、非放电 01: 充电 10: 放电 (重载) 11: 放电 (轻载) ——轻载关机功能使能时才会出现 11 状态, 该状态持续设定的轻载关机时间之后会关闭放电, 进入 00 状态。	R	X
1:0			R	X

## CHG\_STATUS1

寄存器地址 = 0x79

Bit(s)	Name	Description	R/W	Reset
7: 6	Chg_en_state	充电工作状态: 00: 非充电状态 01: 在充电状态, 但是关闭了充电电流, 可能是在停充检测, 也可能是充满了, 也可能是异常保护了。 10: reserved 11: 正在充电	R	X
5	Chg_ovtime	恒压恒流总计时 Timer Out 0: 恒压恒流总计时未超时 1: 恒压恒流总计时超时	R	X
4	Cv_ovtime	恒压计时 Timer Out 0: 恒压计时未超时 1: 恒压计时超时	R	X
3	Tk_ovtime	涓流计时 Timer Out 0: 涓流计时未超时 1: 涓流计时超时	R	X
2:0	Chg_state	充电状态 000: IDLE 001: 涓流充电阶段 010: 恒流充电阶段 011: 恒压充电阶段 100: 停充检测 101: 电池充满结束 110: Timer Out	R	X

## CHG\_STATUS2

寄存器地址 = 0x7A

Bit(s)	Name	Description	R/W	Reset
7		Reserved	R	X
6	Chg_end	充电充满 Charger end 标志: 0: 充电未充满 1: 充电已充满	R	X
2:0			R	X

## LED\_STATUS

寄存器地址 = 0x7C

Bit(s)	Name	Description	R/W	Reset
7:6		Reserved	R	X
5:0	LED	电量指示级别 111111: 5 颗灯亮 011111: 4 颗灯亮 001111: 3 颗灯亮 000111: 2 颗灯亮 000011: 1 颗灯亮 000001: 放电时低电闪灯 000000: 关机	R	X

## LED\_SET

寄存器地址 = 0xC6

Bit(s)	Name	Description	R/W	Reset
7	set_i2c	固定 I2C 模式使能 0: 上电自动判断 I2C 模式 1: 固定为 I2C 模式	RW	0
6:5	led_mode	寄存器设定需要按几颗灯计算电量 11:5 10:4 01:3 00:reserved	RW	0
4	set_dled_r	LED 模式寄存器设置使能 I2C 模式下, 可以通过 0x7C 寄存器查看电量计算结果	RW	0
3:0		Reserved		

## ONOFF\_FLAG

寄存器地址 = 0xea

Bit(s)	Name	Description	R/W	Reset
7	Short2_flag	按键 1s 内双击标志, 写 1 清零	R/W	0
6	Long_flag	按键长按超过 2s 标志, 写 1 清零	R/W	0
5	Short_flag	按键短按标志, 写 1 清零	R/W	0
4:0		Reserved	R	X

## BOOST\_Status

寄存器地址= 0xf5

Bit(s)	Name	Description	R/W	Reset
7:2		Reserved	R	X
1:0	Vbst_state	11: 输出 12V 10: 输出 9V 01: 输出 7V 00: 输出 5V	R	X

## NTC\_Status

寄存器地址= 0Xe9

Bit(s)	Name	Description	R/W	Reset
7		Reserved	R	X
6	Ntc_min_chg	NTC 中温, 充电电流减半信号	R	X
5	Ntc_std_chg	NTC 高温/低温关 charger 信号	R	X
4	Ntc_std_bst	NTC 高温关 boost 信号	R	X
3	Ntc_sc	NTC 短接标志 0: 外部短接到地, NTC 无效 1: 外部接 NTC 电阻, NTC 有效	R	X
2:0	Ntc_in	analog 过来的 3bits 信号 000: 高温 100: 中温 110: 正常温度 111: 低温	R	X



## 4、责任及版权申明

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在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。