

IP5332 寄存器说明文档

1 I2C interface

芯片同时只能支持一种I2C连接方式，按照对应的方式连接就会关闭Function功能，自动进入I2C模式。The I2C speed support 400Kbps.Support 8 bit 寄存器地址width and 8bit data width. Transmit and receive MSB first. The default slave 寄存器地址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 3.3Kohm resister 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 寄存器地址and 8-bit data. After each 8-bit 寄存器地址or data transfer, the IP5332 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 寄存器地址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 寄存器地址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 寄存器地址is 0xEA

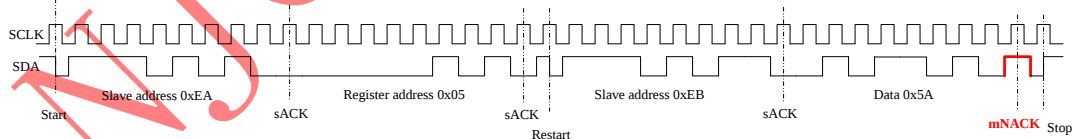
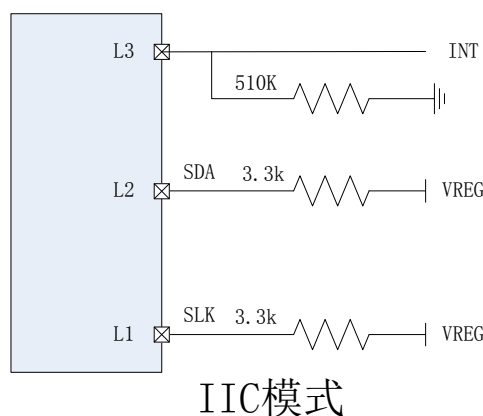


Figure 2 I2C Read

特别说明：在I2C读取数据的末尾，在最后一个BYTE读取完时，一定要给NACK信号，否则IP5332会以为MCU还需要继续读取数据，下一个SCLK就会输出下一个数据，无法正常收到STOP信号，可能会导致I2C总线拉死。

2、I2C Application Notes



- 1、IP5332 标准品默认支持 I2C，不需要单独定制 I2C 版本；
- 2、IP5332 I2C 最高频率支持 400K，考虑到 MCU 时钟偏差，在应用 I2C 时 MCU 通讯的时钟建议用 200K 左右；
- 3、IP5332 从休眠状态转入工作状态（按键、负载接入、5V 充电接入）时，IP5332 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；
- 4、由于 IP5332 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 为高后持续 500ms 以上才开始读写 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
- 5、由于 IP5332 由休眠进入工作状态时会进行 I2C 检测以及 IP5332 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 IP5332 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；
- 6、如果要修改 IP5332 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 其他未开放的 bit 的值不能随意改动，寄存器的默认值以读到的值为准，不同批次的 IC 默认值可能会存在差异。
- 7、MCU 操作流程：INT 为高后唤醒 MCU 后等 500ms 就可以读写 I2C 寄存器，可先初始化寄存器（需要修改特殊功能时才修改寄存器，如果不需要修改可以不写寄存器）然后读取 IC 内部信息（电量、充放电状态、按键状态）进行特性需求的（如特殊指示灯、充放电管理、快充请求管理）操作，INT 为低后就代表 IP5332 已经进入了待机 MCU 也可以考虑进入低功耗模式。

3 典型应用说明

3.1、充放电状态判断

充电状态标志位： 0XD1 bit4=1;
 放电状态标志位： 0XD1 bit4=0;
 充满状态标志位： 0XD7 bit=1;
 未充满状态标志位： 0XD7 bit=0;
 判断当前 B 口(VIN)输入电压有效： 0XD2 bit4=1 ;
 判断当前 C 口(VBUS) 输入电压有效： 0XD2 bit5=1;

3.2、LED 灯电量数据

操作步骤:

将 INT 为高后就将 IP5332 0X0A bit7 写 1, 使能寄存器控制 LED 模式, 然后再设置 0X0A bit6-5 为几灯模式(INT 为高后 1S 内需要写完这两个寄存器);

再读取 0XDB bit4-0, 来判断当前是几个电量指示灯。

3.3、IP5332 内部电量信息

IP5332 内部有电芯电压的 OCV 电压 (通过电芯端电流和设定的内阻补偿后计算出来的电压, 充电只会增加放电只会减少), MCU 可读取 BAT 的 OCV 电压后针对不同的电芯设定不同的转灯电压点, 精准的调整电量的均匀度。

BATOCV 信息寄存器:

寄存器地址= 0x7A

Bit(s)	Name	Description	R/W
7:0	BATOCV[7:0]	BATOCV 数据的低 8bit BAT 电压经过电芯内阻和电芯电流进行补偿后的电压	R

寄存器地址= 0x7B

Bit(s)	Name	Description	R/W
7:0	BATOCV [15:8]	BATOCV 数据的高 8bit OCV=BATOCV*0.26855mv+2.6V BAT 电压经过电芯内阻和电芯电流进行补偿后的电压	R

3.4 MCU 读取 IP5332 内部的电流电压信息来做百分比显示

IP5332 内部有 BAT 端电流寄存器 (此电流是通过输出功率和效率初步计算出来的的电流值, 所以电流不同的时候是有一定误差), MCU 读取 BAT 端电流后可以通过时间和电流积分来调整电量的百分比显示。例如电芯容量为 10000mAH, 平均分为 100 等份后 1%的容量就为 100 mAH, 放电时 BAT 的电流每隔 1S 就累加一次, 当累加的值达到 100 mAH 相应的电量就减 1%, 充电则相反。

BAT 电流 (通过输出的功率和效率计算出来的电流信息) 寄存器:

寄存器地址= 0x66

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

寄存器地址= 0x67

Bit(s)	Name	Description	R/W
7:0	BATIADC[15:8]	BATIADC 数据的高 8bit $IBAT = BATVADC * 1.27883mA$ LSB=1.27883mA 对应的是 10mOhm 采样电阻 (补码格式, 充电为正, 放电为负) 例如: 00000000_00000001 表示 1 11111111_11111111 表示 -1 11111111_11111110 表示 -2	R

3.5 MCU 通过寄存器调整电芯充电电压

操作步骤

1、先将 0X2C bit4 写 0, 将电芯充电电压切换为内部寄存器设定

Bit(s)	Name	Description	R/W	RESET
4	EN_VSET_R	VSET 设置模式 1: 通过外部 VSET 引脚设置 0: 通过寄存器设置 0X22 bit3-2	RW	1

2、再通过调节 0X22 bit3-2 的值来设定不同的充电电压

Bit(s)	Name	Description	R/W	RESET
3:2	VCHG_SET	充电恒压电压设置 11: 4.5V 10: 4.4V 01: 4.35V 00: 4.2V	RW	00

3、如果客户需要微调电芯充电电压时, 可调整恒压加压电压, IP5332 默认加压 14mV 充电

Bit(s)	Name	Description	R/W	RESET
1:0	R_CV	4.5V/4.4V/4.35V/4.2V 电池恒压快充: 11: 恒压增加 42mv 10: 恒压增加 28mv 01: 恒压增加 14mv, 默认加 14mv 00: 不增加, 如果不加压时可能有些 IC 不能正常停充, 需要 MCU 自己判断 是否有充饱, 充饱条件可根据电芯电压和电芯电流同时来设定如 4.2V 不加压, 电芯电压大于 4.175V & IBAT 小于 200mA, 就认为是 充饱状态, 不需要一直等 IP5332 充满标志位置起。	RW	01

3.6 MCU 通过寄存器关闭和打开所有口的输入输出快充协议

1、通过寄存器关闭和打开输入快充协议 (不包括 PD 协议):

关闭输入快充协议: 将 0XA1 bit4-bit2 写 000;

打开输入快充协议: 将 0XA1 bit4-bit2 写 111;

2、通过寄存器关闭和打开输出快充协议 (不包括 PD 协议)

关闭输出快充协议: 将 0XA2 bit6-bit0 写 000000;

打开输出快充协议: 将 0XA2 bit6-bit0 写 111111;

3、关闭输入输出 PD 快充协议

关闭输入输出 PD 快充: 0X1C bit1 写 0;

打开输入输出 PD 快充: 0X1C bit1 写 1;

注: IP5332 输入输出 PD 快充协议只有总使能, 输入输出没有单独的使能寄存器。

3.7 MCU 通过寄存器关闭和打开不同口的输入输出 DM DP 的快充协议

寄存器地址= 0x3E

Bit(s)	Name	Description	R/W	RESET
3	EN_QC_VBUS	VBUS 通路快充使能 0: disable 1: enable	RW	1
2	EN_QC_VIN	VIN 通路快充使能 0: disable 1: enable	RW	1
0	EN_QC_VOUT1	VOUT1 通路快充使能 0: disable 1: enable	RW	1

3.8 MCU 读取 NTC PIN 的电压来判断不同的温度

1、设置 NTC 复用为 ADC: 0X51 bit6-4 写为 010

寄存器地址= 0x51

Bit(s)	Name	Description	R/W	RESET
6:4	NTC_SEL	000: NTC 功能 010: ADC 功能	RW	000

2、开启 GPIO_ADC 的 ADC 功能: 0X80 bit4 写 1

寄存器地址= 0x80

Bit(s)	Name	Description	R/W	RESET
4	EN_GPIO_ADC	GPIO_ADC 使能 0: disable 1: enable	RW	0

3、关闭 NTC 功能: 0X04 bit0 写 0

寄存器地址= 0x04

Bit(s)	Name	Description	R/W	RESET
0	EN_NTC	NTC 保护使能 1: enable 0: disable	R/W	1

4、读取 NTC PIN 的电压: 读 0X78 0X79 的值来计算 NTC PIN 的电压想做相应的处理。

4 可读/写操作寄存器

*Reserved 的寄存器不可随意写入数据, 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。

*本文档的寄存器默认值仅代表某一种规格, 绝大多数规格的寄存器默认值与本文档并不对应, 所以在

读写操作时需特别注意按位操作。

4.1 SYS_CTL1(boost 和 charger 使能寄存器)

寄存器地址= 0x01

Bit(s)	Name	Description	R/W	RESET
7:3		Reserved		XX
2	EN_BOOST	Boost 升压使能 1: enable 0: disable	R/W	1
1	EN_Charger	Charger 充电使能 1: enable 0: disable	R/W	1
0		Reserved		XX

4.2 SYS_CTL2 (按键控制寄存器)

寄存器地址= 0x03

Bit(s)	Name	Description	R/W	RESET
7	EN_loadotp	开机寄存器复位使能 1: enable 0: disable (disable 此功能后, IP5332 不掉电的情况下每次开机寄存器不会复位为默认值。一般情况不建议操作该 bit 寄存器, 防止 MCU 误写寄存器后开机不能自动恢复为默认值)	R/W	1
6:4		Reserved		XX
3	Set_onoff_time	按键长按的时间设置 0: 2s 1: 3s	R/W	0
2	En_onoffrst_r	按键超长按 10S 复位使能 1: enable 0: disable	R/W	1
1:0	Set_onoff_dn	按键关机方式选择 00/10: disable 01: 短按两次 11: 长按	R/W	01

4.3 SYS_CTL3 (芯片内部温度和 NTC 温度控制寄存器)

寄存器地址= 0x04

Bit(s)	Name	Description	R/W	RESET
7	EN_BSTTMDN	芯片高温关 BOOST 使能 1: enable 0: disable	R/W	1
6	EN_CHGTMDN	芯片高温关 charger 使能 1: enable 0: disable	R/W	1
5	EN_NTCL_BST	NTC 低温关 BOOST 使能 1: enable 0: disable	R/W	1
4	EN_NTC_MID	NTC 中温充电电流减半使能 1: enable 0: disable	R/W	1
3	EN_NTC_CHG	NTC 高低温关 Charger 使能 1: enable 0: disable	R/W	1
2	EN_NTCH_BST	NTC 高温关 BOOST 使能	R/W	1

		1: enable 0: disable		
1	EN_NTC_SC	NTC 接地时(NTC<0.2V), 关闭 NTC 功能使能 1: enable 0: disable	R/W	0
0	EN_NTC	NTC 保护使能 1: enable 0: disable	R/W	1

4.4 IC_TEMP(IC 内部过温寄存器)

寄存器地址= 0x42

Bit(s)	Name	Description	R/W	RESET
7:3		Reserved		XX
2	ENTSBST	芯片内部温度保护点全部增加15C 1: enable 0: disable	R/W	0
1	HT	芯片内部温度高温报警和恢复温度 1 :140C, 80C 0 :130C, 80C	R/W	1
0	TSEN	芯片内部温度检测使能 1: enable 0: disable	R/W	1

4.5 CHG_NTC_TEMP(充电 NTC 阈值寄存器)

寄存器地址= 0x43

Bit(s)	Name	Description	R/W	RESET
7	EN_INTC	使能NTC放出20UA电流 1: enable 0: disable	R/W	1
6:5	CHG_NTC_HT	充电NTC高温阈值 11: 0.56V (理论温度45度) 10: 0.49V (理论温度50度) 01: 0.43V (理论温度55度) 00: 0.38V (理论温度60度)	R/W	01
4:3	CHG_NTC_MT	充电NTC中温阈值 11: 0.6V (理论温度41度) 10: 0.58V (理论温度43度) 01: 0.56V (理论温度45度) 00: 0.54V (理论温度47度)	R/W	01
2:1	CHG_NTC_LT	充电NTC低温阈值 11: 1.52V (理论温度-20度) 10: 1.49V (理论温度-15度) 01: 1.44V (理论温度-10度) 00: 1.32V (理论温度0度)	R/W	01

4.6 BST_NTC_TEMP(放电 NTC 阈值寄存器)

寄存器地址= 0x54

Bit(s)	Name	Description	R/W	RESET
7:6	BST_NTC_HT	放电NTC高温阈值 11: 0.56V (理论温度45度) 10: 0.49V (理论温度50度) 01: 0.43V (理论温度55度) 00: 0.38V (理论温度60度)	R/W	01
5:4	BST_NTC_LT	放电NTC低温阈值 11: 1.52V (理论温度-20度)	R/W	11

		10: 1.49V (理论温度-15度) 01: 1.44V (理论温度-10度) 00: 1.32V (理论温度0度)		
--	--	--	--	--

4.7 SYS_CTL4 (照明灯控制寄存器)

寄存器地址= 0x05

Bit(s)	Name	Description	R/W	RESET
7:4		Reserved		XX
3	En_wled_on_r	WLED 手电筒按键开关控制方式 0: 长按 1: 短按两次	R/W	0
2:1		Reserved		XX
0	En_wled_r	WLED 手电筒 (跟 KEY 复用的) 1: enable 0: disable	R/W	1

4.8 SYS_CTL5 (轻载关机时间设置)

寄存器地址= 0x07

Bit(s)	Name	Description	R/W	RESET
7:6	Set_illow_bst	输出轻载关机进入休眠时间: 00: 8s 01: 16s 10: 32s 11: 63s 如果需要更长的时间来轻载关机, 可以将轻载关机使能 (0x84 bit6 写 0) 关闭一段时间再打开	R/W	10
5:0		Reserved		XX

4.9 SYS_CTL6 (电量灯配置寄存器)

寄存器地址= 0x0A

Bit(s)	Name	Description	R/W	RESET
7	Set_dled_r	LED 模式寄存器设置使能 1: enable 0: disable I2C 模式下, 可以通过 0xDB 寄存器查看电量计算结果, 电量计算结果寄存器见 0xDB 寄存器	R/W	0
6:5	Dled_mode_r	寄存器设置几灯模式计算电量 00: 1 灯 01: 2 灯 10: 3 灯 11: 4 灯	R/W	00
4:0		Reserved		XX

4.10 LED_STATUS (电量信息寄存器)

寄存器地址= 0xDB

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

7:5		Reserved		XX
4:0	LED	电量指示级别 11111: 4 颗灯亮 01111: 3 颗灯亮 00111: 2 颗灯亮 00011: 1 颗灯亮 00001: 放电时低电闪灯 (0CV 阈值低于 3.3V 时) 00000: 关机 默认状态为 1 颗电量灯, 注意一定要用寄存器先写为 4 灯或者其他数目的电量灯	R	XX

4.11 SYS_CTL7 (输出口自动检测负载寄存器)

寄存器地址= 0x0B

Bit(s)	Name	Description	R/W	RESET
7		Reserved		XX
6:5	Set_pod_time	拔掉充电转放电状态时 输出口关闭间隔时间设置 00:1s 01:2s 10:3s 11:4s	R/W	00
4-3		Reserved		XX
Bit2 和 bit0	VOUT1_DET	00: 关闭 VOUT1 口负载接入自动开机检测功能 恢复 VOUT1 口负载接入自动开机检测功能 Bit2 和 bit0 恢复默认值	R/W	10

4.12 SYS_CTL8 (按键控制输出口寄存器)

寄存器地址= 0x0C

Bit(s)	Name	Description	R/W	RESET
7	KEY_TC	按键短按开启 VBUS 输出口模式选择 1: 直接打开 C 口 0: 检测 C 口是否有负载, 如果有负载就开启 C 口	R/W	0
6		Reserved		XX
5	KEY_VOUT1	按键短按开启 VOUT1 输出口模式选择 1: 直接打开 VOUT1 输出口 0: 检测 VOUT1 输出口是否有负载, 如果有负载就开启 VOUT1 输出口	R/W	1
4: 0		Reserved		11111

4.13 SYS_CTL9 (边充边放功能寄存器)

寄存器地址= 0x0D

Bit(s)	Name	Description	R/W	RESET
7:6		Reserved		XX
5:4		VIN 和 VBUS 充电优先级选择 00: 看接入先后顺序, 先接入者优先充电 01: 看输入的电压, 高电压优先 1X: 同等电压时, VBUS 优先	R/W	01
3		Reserved		XX
2		使能同充同放时自动设定给电芯充电电流最大 500mA 左右 1: enable 0: disable	R/W	0
1	En_maxvinlp_r	使能同充同放时自动提高 5V 充电欠压环到 4.72V 优先给负载充电 1: enable	R/W	0

		0: disable		
0	En_same_r	同充同放使能 1: enable 0: disable	R/W	0

4.14 SYS_CTL10

寄存器地址= 0x0E

Bit(s)	Name	Description	R/W	RESET
7		Reserved		XX
6	En_chg2bst_r	Charge 拔出是否自动开启 Boost 1: 开启 0: 不开启 (如果检测到输出口有负载就会开启升压打开输出口)	R/W	1
5: 3		Reserved	R/W	XX
2	En_swclk1_r	切换 I2C 模式 1 待机时钟使能 (L1/L2 的 I2C) 1: enable 0: disable 使能后可在待机时 I2C 可以以低于 10k 的速度进行访问	R/W	0
1:0		Reserved		XX

4.15 SYS_CTL11(常开小电流模式设置)

寄存器地址= 0x0F

Bit(s)	Name	Description	R/W	RESET
7:3		Reserved		XX
2:1	Lowcur_time	常开时间设置 11: 8 小时 10: 6 小时 01: 4 小时 00: 2 小时	R/W	00
0	En_lowcur	输出常开 N 小时使能 (包括多口转单口功能也关闭) 1: enable 0: disable 默认按键双击进入常开 N 小时模式, 按键短按退出常开 N 小时模式, 所以在使能此功能后需要将按键双击功能关闭	R/W	0

4.16 VBAT_LOW (低电关机阈值设置)

寄存器地址= 0x10

Bit(s)	Name	Description	R/W	RESET
7:6		Reserved		11
5:4	BATLOWSET	BAT 实际电压低电机电压压设定 11 3.00 $\leftrightarrow$ 3.10 10 2.90 $\leftrightarrow$ 3.00 01 2.81 $\leftrightarrow$ 2.89 00 2.73 $\leftrightarrow$ 2.81	R/W	10
3:0		Reserved		XX

4.17 VINOV (VIN 输入过压设置)

寄存器地址= 0x11

Bit(s)	Name	Description	R/W	RESET
7:2		Reserved		XX

1:0	VINOVSET	VIN 充电过压设置 11:16V 10:14V 01:6.0V 00:5.6V	R/W	10
-----	----------	--	-----	----

4.18 VBUSOV (VBUS 输入过压设置)

寄存器地址= 0x12

Bit(s)	Name	Description	R/W	RESET
7:2		Reserved		XX
1:0	VBUSOVSET	VBUS 充电过压设置 11:16V 10:14V 01:6.0V 00:5.6V	R/W	10

4.19 BOOST_LINC(输出线补设置寄存器)

寄存器地址= 0x13

Bit(s)	Name	Description	R/W	RESET
7:2		Reserved		XX
1	Rlineext	模拟线补使能 1: enable 0: disable	R/W	1
0	RLINC	线补选择: 1: 250mV@2A, 对应 V _{SY} S 端的输出电压 0: 125mV@2A, 对应 V _{SY} S 端的输出电压	R/W	0

4.20 TYPE-C_CTRL0 (typec CC 上拉设置寄存器)

寄存器地址= 0x1A

Bit(s)	Name	Description	R/W	RESET
7:4		Reserved		XX
3:2	SRC_Rp	非充电时, USB TypeC 的 SRC 电流设置 00: default 01: 1.5A 10: 3A	RW	10
1:0		Reserved		10

4.21 TYPE-C_CTRL1 (typec CC 模式配置寄存器)

寄存器地址= 0x1B

Bit(s)	Name	Description	R/W	RESET
7:2		Reserved		XX
1:0	CC_mode	CC mode sel 00:UFP 01:DFP	RW	11

		11:DRP		
--	--	--------	--	--

4.22 TYPE-C_CTRL2 (typec PD 协议使能寄存器)

寄存器地址= 0x1C

Bit(s)	Name	Description	R/W	RESET
7:2		Reserved		XX
1	PD function enable	PD 输入输出协议 1: enable 0: disable	RW	1
0		Reserved		1

4.23 TYPE-C_CTRL3

寄存器地址= 0x1E

Bit(s)	Name	Description	R/W	RESET
7		Reserved		XX
6:5	Vmax_PDSink	PD 输入请求最高电压设置 00: 5V 01: 7V 10: 9V 11: 12V	RW	10
4:0		Reserved		XX

4.24 CHG_CTL1 (充电停充电压设置寄存器)

寄存器地址= 0x21

Bit(s)	Name	Description	R/W	RESET
7:4		Reserved		XX
3:2	VCHG_SET	电芯停充电压设置: 4.2V/4.35V/4.4V/4.5V 11: 4.2V/4.35V/4.4V/4.5V 10: 4.185V/4.337V/4.382V/4.476V 01: 4.17V/4.321V/4.366V/4.46V 00: 4.155V/4.306V/4.35V/4.44V	RW	01
1:0	R_CV	Reserved	RW	XX

4.25 CHG_CTL2 (充电充满电压设置寄存器)

寄存器地址= 0x22

Bit(s)	Name	Description	R/W	RESET
7:4		Reserved		XX
3:2	VCHG_SET	电芯充满电压设置: 11:4.5 10:4.4 01:4.35 00:4.2 0X2C bit4 需要先写 0, 设置为寄存器设置充满电压	RW	00
1:0	R_CV	4.5V/4.4V/4.35V/4.2V 电池恒压快充: 11:恒压增加 42mv 10:恒压增加 28mv 01:恒压增加 14mv 00:不增加 此电压可以微调充满电压, 但是停充电压也需要同步设置 且充满电压-停充电压需要大于 40mv, 否则可能有些 IC 不能停充	RW	01

4.26 CHG_CTL3（充电欠压环设置寄存器）

寄存器地址= 0x23

Bit(s)	Name	Description	R/W	RESET
7:6		Reserved		XX
5:3	R_VIL7	充电输入欠压环阈值@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	充电输入欠压环阈值@5V: 111:4.92 110: 4.88 101:4.84 100: 4.8 011: 4.76 010: 4.72 001: 4.64 000: 4.58 边充边放状态下使能 0X0D bit1 后修改该寄存器修改无效，因为边充边放时已经将欠压环固定为 4.72V	RW	000

4.27 CHG_CTL4（充电欠压环设置寄存器）

寄存器地址= 0x24

Bit(s)	Name	Description	R/W	RESET
7:6		Reserved		XX
5:3	R_VIL12	充电输入欠压环阈值@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	充电输入欠压环阈值@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

4.28 CHG_ISET_9V

寄存器地址= 0x26

Bit(s)	Name	Description	R/W	RESET
7:6		Reserved		XX
5:0	Chg_iset_H9V	Charge ISET @ 9V 充电时 输入充电电流: $I_{chg}=ISET*0.05A$ 校准值为 2A 充电电流，如需要调整充电电流时可在校准值的基础上增加或者减小相应的档位	RW	校准值

4.29 CHG_ISET_12V

寄存器地址= 0x27

Bit(s)	Name	Description	R/W	RESET
7:6		Reserved		XX
5:0	Chg_iset_H12V	Charge ISET @ 12V 充电时 输入充电电流: $I_{chg}=ISET*0.05A$ (实际充电电流会比理论值偏小 0~300mA)	RW	校准值

4.30 CHG_ISET_5V_VBUS

寄存器地址= 0x29

Bit(s)	Name	Description	R/W	RESET
7:6		Reserved		XX
5:0	Chg_iset_vbus	Charge ISET @ VBUS 5V 充电时 输入充电电流: $I_{chg}=ISET*0.05A$ 校准值为 2.6A 充电电流, 如需要调整充电电流时可在校准值的基础上增加或者减小相应的档位	RW	校准值

4.31 CHG_ISET_5V_VIN

寄存器地址= 0x2A

Bit(s)	Name	Description	R/W	RESET
7:6				XX
5:0	Chg_iset_vin	Charge ISET @ VIN 5V 充电时 输入充电电流: $I_{chg}=ISET*0.05A$ (实际充电电流会比理论值偏小 0~300mA) 校准值为 2A 充电电流, 如需要调整充电电流时可在校准值的基础上增加或者减小相应的档位	RW	校准值

4.32 CHG_ISET_7V

寄存器地址= 0x2B

Bit(s)	Name	Description	R/W	RESET
7:6	Set_iset_tk	涓流时的 Charge ISET 电池端充电电流: $I_{chg}=ISET*0.05A+0.10A$	RW	XX
5:0	Chg_iset_H5V	Charge ISET @ 7V 充电时 输入充电电流: $I_{chg}=ISET*0.05A$ (实际充电电流会比理论值偏小 0~300mA) 校准值为 2.4A 充电电流, 如需要调整充电电流时可在校准值的基础上增加或者减小相应的档位	RW	校准值

4.33 CHG_TIMER_EN (充电超时设置寄存器)

寄存器地址= 0x2C

Bit(s)	Name	Description	R/W	RESET
7	En_tktime_r	涓流计时使能 0: disable 1: enable	RW	1
6	En_cvtime_r	Charge CV 计时使能 0: disable 1: enable	RW	1

5	En_chgtime_r	Charge 总的计时使能 (CC+CV) 0: disable 1: enable	RW	1
4	En_vset_r	VSET 设置模式 0: 通过寄存器设置(0x22h[3:2]) 1: 通过引脚设置	RW	1
3:1		Reserved		XX
0	En_tk_r	涓流使能 0: disable 1: enable	RW	1

4.34 CHG_TIMER_SET (充电超时设置寄存器)

寄存器地址= 0x2D

Bit(s)	Name	Description	R/W	RESET
7:6	Set_tk_time	涓流超时设置 00: 2h 01: 3h 10: 4h 11: 6h	RW	11
5:4	Set_pcc_time	恒压间断时间 N 设置 (快充满时每隔 N 分钟减充电电流减小到 0 看电芯电压是否充满) 00: 2min 01: 4min 10: 8min 11: 16min	RW	01
3:2	Set_cv_time	Charge CV 超时设置 00: 2h 01: 4h 10: 6h 11: 8h	RW	11
1:0	Set_chg_time	Charge CC+CV 超时设置 00: 8h 01: 12h 10: 16h 11: 24h	RW	11

4.35 DCDC_FREQ (DC-DC 开关频率设置寄存器)

寄存器地址= 0x31

Bit(s)	Name	Description	R/W	RESET
7:5	BSTFRQ	BST 频率设定 125k+125k*code 开关频率不建议客户调整, 如需调整时可与原厂工程师确认	RW	010
4:2	CHGFRQ	CHG 频率设定 125k+125k*code 开关频率不建议客户调整, 如需调整时可与原厂工程师确认	RW	010
1:0		Reserved		XX

4.36 QC_EN (输入输出 DCP 快充协议使能寄存器)

寄存器地址= 0x3E

Bit(s)	Name	Description	R/W	RESET
7: 4		Reserved		XX
3	EN_QC_VBUS	VBUS 通路输入输出 (不括 PD 和 MKT 协议)	RW	1

		0: disable 1: enable		
2	EN_QC_VIN	VIN 通路输入快充使能 0: disable 1: enable	RW	1
1		Reserved		XX
0	EN_QC_VOUT1	VOUT1 通路输出快充使能（不括 MKT 协议） 0: disable 1: enable	RW	1

4.37 PMOS_REG_CTL0（输入输出口控制寄存器）

寄存器地址= 0x59

Bit(s)	Name	Description	R/W	RESET
7:4		Reserved		XX
3	En_vout1_r	寄存器控制 vout1 通路 0: 状态机自动控制 1: 寄存器控制	R/W	0
2	Vout1_ctrl	寄存器控制 vout1 通路 0: 关闭 1: 开启	R/W	0
1	En_vin_r	寄存器控制 VIN 通路 0: 关闭 1: 开启	R/W	0
0	Vin_ctrl	寄存器控制 VIN 通路 0: 状态机自动控制 1: 寄存器控制	R/W	0

4.38 PMOS_REG_CTL1（输入输出口控制寄存器）

寄存器地址= 0x5A

Bit(s)	Name	Description	R/W	RESET
7:6		Reserved		00
5	Set_vinlp_mode	寄存器设定或者状态设定充电档位 1: 寄存器(0x5Ah[4:3])（只能固定一个档位的电压充电） 0: 状态设定	R/W	0
4:3	Svinloop_r	寄存器设定充电电压档位 00: 5V 01: 7V 10: 9V 11: 12V	R/W	00
2	En_vbusi_r	寄存器控制 VBUS 输入通路 0: 关闭 1: 开启	R/W	0
1	En_vbuso_r	寄存器控制 VBUS 输出通路 0: 关闭 1: 开启	R/W	0
0	Vbus_ctrl	寄存器控制 VBUS 通路控制 0: 状态机自动控制 1: 寄存器控制(0x5A[2: 1])	R/W	0

4.39 FORCE_EN（寄存器关机和复位控制寄存器）

寄存器地址= 0x5B

Bit(s)	Name	Description	R/W	RESET
7	Force_WLED		R/W	0
6	En_force_WLED	先将 bit6 写 1，再将 bit7 写 1，可内部触发开启 WLED； 开启之后，先将 bit7 写 0，再将 bit6 写 0，可关闭 WLED	R/W	0

5	Force_reset		R/W	0
4	Force_boost		R/W	0
3		Reserved		XX
2	En_force_restart	先将 bit2 写 1，再将 bit5 写 1，可内部触发芯片复位	R/W	0
1	En_force_boost	先将 bit1 写 1，再将 bit4 写 1，可内部触发开启 BOOST。 BOOST 开启时，在 bit1 和 bit4 都为 1 的情况下把 bit4 写 0，可触发关 BOOST 并且立即关机。只将 0x01 寄存器 bit2 写 0 只会关闭 BOOST，但不会立即关机，关机还是需要等轻载条件。	R/W	0
0		Reserved		XX

4.40 FLAG0(异常标志位)

寄存器地址= 0x7E

Bit(s)	Name	Description	R/W
7		BOOST 短路异常标志 (需写 1 清 0)	R/W
6		BOOST 欠压异常标志(输出过流标志) (需写 1 清 0)	R/W
5		BOOST 异常打嗝标志 (需写 1 清 0)	R/W
4		BOOST 异常关机标志 (需写 1 清 0)	R/W
3		BOOST 启动失败标志 (需写 1 清 0)	R/W
2		NTC 低温标志 (需写 1 清 0)	R/W
1		NTC 高温标志 (需写 1 清 0)	R/W
0		IC 内部高温标志 (需写 1 清 0)	R/W

4.41 FLAG1 (按键和过压标志位)

寄存器地址= 0x7F

Bit(s)	Name	Description	R/W
7		按键双击标志 (需写 1 清 0)	R/W
6		按键长按标志 (需写 1 清 0)	R/W
5		按键短按标志 (需写 1 清 0)	R/W
4		Reserved	XX
3		Reserved	XX
2		Reserved	XX
1		VBUS 过压标志 (需写 1 清 0)	R/W
0		VIN 过压标志 (需写 1 清 0)	R/W

4.42 BST_POWERLOW (轻载关机功率阈值设置寄存器)

寄存器地址= 0x81

Bit(s)	Name	Description	R/W	RESET
7:5		Reserved		XX
4:0	Set_power_th	Config BOOST powerlow hreshold (轻载关机功率阈值) powerlow LSB 16.88mW 默认值为 300mW，最大值 540mW	RW	10010

4.43 RSET (电池内阻补偿寄存器)

寄存器地址= 0x82

Bit(s)	Name	Description	R/W	RESET
7:4	Set_bat_imp	Register Config impendence 6.25mOhm* set_bat_imp 内部寄存器设定电芯内部补偿	RW	011
3:2	Set_imp_offse	Config impendence offset 00:0 mohm	RW	000

	t	01:12.5 mohm 10:25 mohm 11:50 mohm 设定内阻值= offset+6.25mOhm* set_bat_imp		
1	Sel_ext_imp	电芯内阻 RSET 设置寄存器 1: 外部 RSET 设置寄存器 0: 内部寄存器设置 (0X82 bit7-2)	RW	0
0		Reserved		XX

4.44 BST_ISYSLOW (轻载关机电流寄存器)

寄存器地址= 0x84

Bit(s)	Name	Description	R/W	RESET
7	En_isyslow_r	ISYSLOW BOOST 状态输出总电流轻载关机使能 1:enable 0:disable	RW	0
6	En_powerlow_r	POWERLOW BOOST 状态输出总功率轻载关机使能 1:enable 0:disable 默认为看输出总功率来轻载关机 (阈值设置寄存器在 0x81 bit4:0)	RW	1
5:0	Set_isys_th	Config ISYS_LOW hreshold (轻载输出电流阈值设定) isyslow LSB 2.55766mA	RW	010111

4.45 IPMOSLOW (多口转单口 MOS 关电流阈值设置寄存器)

寄存器地址= 0x86

Bit(s)	Name	Description	R/W	RESET
7:0	Set_ipmos_th	Config IPMOS hreshold 多口输出时主动关闭轻载输出的电流阈值设定 LSB=2.55766mA 对应的是 10mOhm 采样电阻, 实际情况需要根据 PCB 上 VSN 到 VOUT1 走线阻抗+MOS 管导通阻抗跟 10mOhm 的倍数关系进行折算。假如实际阻抗有 20mOhm, 是 10mOhm 的 2 倍, 那么 LSB 也是理论值的 1/2 倍	RW	0x46

4.46 BATOCV_LOW(低电退出主动退出快充设置寄存器)

寄存器地址= 0x88

Bit(s)	Name	Description	R/W	RESET
7:5		Reserved		XX
4	En_dndcp	BATOCV 电芯低电压关闭芯片自带的快充输出功能使能 1:enable 0:disable *外挂快充协议 IC 触发的快充不受此控制	RW	0
3:0	Set_dndcph_r	BATOCV 低电压关快充阈值 (Offset=2.6V, LSB=69mV)	RW	0000

4.47 IPMOSLOW_TIME (多口转单口时间设置寄存器)

寄存器地址= 0x90

Bit(s)	Name	Description	R/W	RESET
7:6		PMOS ilow 时间相对轻载关机时间的关系: 00: 轻载关机时间的一半 01: 与轻载关机时间一样 10: 轻载关机时间的 2 倍 11: 轻载关机时间的 4 倍	R/W	00
5:0		Reserved		XX

4.48 QC_VMAX (设置 QC 协议最大输出电压)

寄存器地址= 0x96

Bit(s)	Name	Description	R/W	RESET
7		Reserved		XX
6		设置 QC SRC 模式支持的 MAX 最大电压 1:12V 0:9V	R/W	1
5:0		Reserved		XX

4.49 BATOCV_LOW_DN

寄存器地址= 0x9F

Bit(s)	Name	Description	R/W	RESET
7:6		Reserved		XX
5:0		BATOCV_LOW (default 2.900V) 2600+data*8.59375mv	R/W	000000

4.50 DCP_DIG_CTL0

寄存器地址= 0xA0

Bit(s)	Name	Description	R/W	RESET
7:4		Reserved		XX
3	En_vbus_dcp_r	VBUS 的 DPC DMC 使能 (DM DP 相关的所有协议包括普通 5V 的苹果、三星、BC1.2 协议) 1:enable 0:disable	RW	1
2	En_vin_dcp_r	VIN 的 DPB DMB 使能 (DM DP 相关的所有协议包括普通 5V 的苹果、三星、BC1.2 协议) 1:enable 0:disable	RW	1
1		Reserved		XX
0	En_yout1_dcp_r	VOUT1 的 DPA1 DMA1 使能 (DM DP 相关的所有协议包括普通 5V 的苹果、三星、BC1.2 协议) 1:enable 0:disable	RW	1

4.51 DCP_DIG_CTL1(输入协议使能寄存器)

寄存器地址= 0xA1

Bit(s)	Name	Description	R/W	RESET
7		Reserved	RW	XX
6		Reserved	RW	XX
5		Reserved	RW	XX
4		SFCP SINK 使能 (输入使能)	RW	0
3		AFC SINK 使能 (输入使能)	RW	1
2		FCP SINK 使能 (输入使能)	RW	1
1		Reserved	RW	XX
0		Reserved	RW	XX

4.52 DCP_DIG_CTL2(输出协议使能寄存器)

寄存器地址= 0xA2

Bit(s)	Name	Description	R/W	RESET
7	En_mtkrx9v_r	MTK PE 1.1 RX 支持的最大电压设置 0: 12V 1: 9V	RW	0
6	En_mtkrx2_r	MTK PE2.0 RX 使能 (输出使能) 1:enable 0:disable	RW	1
5	En_mtkrx1_r	MTK PE1.1 RX 使能 (输出使能) 1:enable 0:disable	RW	1
4	En_sfcpsrc_r	SFCP SRC 使能 (输出使能) 1:enable 0:disable	RW	1
3	En_afcsrc_r	AFC SRC 使能 (输出使能) 1:enable 0:disable	RW	1
2	En_fcpsrc_r	FCP SRC 使能 (输出使能) 1:enable 0:disable	RW	1
1	En_qc3src_r	QC3.0 SRC 使能 (输出使能) 1:enable 0:disable	RW	1
0	En_qc2src_r	QC2.0 SRC 使能 ((输出使能)) 1:enable 0:disable	RW	1

注: QC2.0 和 QC3.0 使能关闭后对应的 AFC 和 FCP 协议也会随着关闭

4.53 BOOST_5V_ISET

寄存器地址= 0xA8

Bit(s)	Name	Description	R/W	RESET
7:6		Reserved		
5:0	Iset_5v_r	BST 5V 时输出总的限流设定, 默认值约为 3.2A, step50mA (该寄存器会影响大于 5V 时的 18W 功率限制) 注意需要修改输出功率时需要在默认值的基础上进行加减, 不能直接写固定值, 写固定值 IC 的个体差异会比较大, 同时寄存器写全 0 输出电流并不一定为 0。 如有降低输出功率时, C 口的 PD 包不能跟随减小, 有存在 PD 包信息大于输出功率的情况。	RW	校准值

4.54 BOOST_VSET (输出电压控制寄存器)

寄存器地址= 0x4C

Bit(s)	Name	Description	R/W	RESET
7:6		Reserved		XX
3:2	Trsel_set	11: 12V 10: 9V 01: 7V 00: 5V 此电压设定后只能应用在单口模式, 多口应用是系统会过压保护	RW	00
1	Trsel_ctl	BOOST 电压控制方式: 1: 寄存器 0X4C bit3-2 控制输出电压 0: IP5332 内部状态机控制输出电压	RW	0

4.55 DCP_DIG_CTL10

寄存器地址= 0xAA

Bit(s)	Name	Description	R/W	RESET
7:6	At_same_mode	同充同放下放电口 DP DM 的状态 11: 支持 Apple、三星、BC1.2 10: 浮空 01: 短接 00: 支持 Apple、三星、BC1.2	RW	01
5:0		Reserved		XX

5 只读状态指示寄存器

5.1 BATVADC_DAT0 (BAT 真实电压寄存器)

寄存器地址= 0x64

Bit(s)	Name	Description	R/W
7:0	BATVADC[7:0]	BATVADC 数据的低 8bit BATPIN 的真实电压	R

5.2 BATVADC_DAT1 (BAT 真实电压寄存器)

寄存器地址= 0x65

Bit(s)	Name	Description	R/W
7:0	BATVADC[15:8]	BATVADC 数据的高 8bit $VBAT = BATVADC * 0.26855mv + 2600mv$ BATPIN 的真实电压	R

5.3 BATIADC_DAT0 (BAT 端电流寄存器)

寄存器地址= 0x66

Bit(s)	Name	Description	R/W
7:0	BATIADC[7:0]	电芯端电流 BATIADC 数据的低 8bit	R

5.4 BATIADC_DAT1 (BAT 端电流寄存器)

寄存器地址= 0x67

Bit(s)	Name	Description	R/W
7:0	BATIADC[15:8]	电芯端电流 BATIADC 数据的高 8bit $IBAT = BATVADC * 1.27883mA$ LSB=1.27883mA 对应的是 10mOhm 采样电阻 (补码格式, 充电为正, 放电为负) 例如: 00000000_00000001 表示 1	R

		11111111_11111111 表示-1 11111111_11111110 表示-2 该电流是通过输出功率和效率计算出来的值，不同大小的电流存在一定的误差	
--	--	--	--

5.5 SYSVADC_DAT0 (VSYS 端电压值寄存器)

寄存器地址= 0x68

Bit(s)	Name	Description	R/W
7:0	SYSVADC[7:0]	SYSVADC 数据的低 8bit (VSYS 端电压值)	R

5.6 SYSVADC_DAT1 (VSYS 端电压值寄存器)

寄存器地址= 0x69

Bit(s)	Name	Description	R/W
7:0	SYSVADC [15:8]	SYSVADC 数据的高 8bit (VSYS 端电压值) $SYSV = SYSVADC * 1.61133mV + 15.6V$ (补码格式，有效值通常为负值) 例如: 00000000_00000001 表示 1 11111111_11111111 表示-1 11111111_11111110 表示-2	R

5.7 SYSIADC_DAT0 (10 毫欧采样电阻流过的电流寄存器)

寄存器地址= 0x6A

Bit(s)	Name	Description	R/W
7:0	YSIADC [7:0]	YSIADC 数据的低 8bit (VSYS 端电流)	R

5.8 SYSIADC_DAT1 (10 毫欧采样电阻流过的电流寄存器)

寄存器地址= 0x6B

Bit(s)	Name	Description	R/W
7:0	YSIADC [15:8]	SYSIADC 数据的高 8bit (VSYS 端电流) $YSI = SYSIADC * LSB$ $LSB = 0.6394mA$ 对应的是 10mOhm 采样电阻 (补码格式，放电为负，充电为正) 例如: 00000000_00000001 表示 1 11111111_11111111 表示-1 11111111_11111110 表示-2	R

5.9 VINIADC_DAT0 (VIN 通路 MOS 流过的电流寄存器)

寄存器地址= 0x6C

Bit(s)	Name	Description	R/W
7:0	VINIADC [7:0]	VINIADC 数据的低 8bit (VIN 端电流) 此电流只在多口状态下才起作用	R

5.10 VINIADC_DAT1 (VIN 通路 MOS 流过的电流寄存器)

寄存器地址= 0x6D

Bit(s)	Name	Description	R/W
7:0	VINIADC [15:8]	VINIADC 数据的高 8bit (VIN 端电流) $VINI = VINIADC * LSB$ $LSB = 0.6394mA$ 对应的是 10mOhm 采样电阻，实际情况需要根据 PCB 上 VSN 到 VIN 的走线阻抗+MOS 管导通阻抗跟 10mOhm 的倍	R

		数关系进行折算。假如实际阻抗有 20mOhm，是 10mOhm 的 2 倍，那么 LSB 也是理论值的 1/2 需要在充电状态 VINOK 和 VBUSOK 同时有效且 VIN MOS 开启时该 ADC 才会启动 （补码格式，放电为正，充电为负） 例如：00000000_00000001 表示 1 11111111_11111111 表示-1 11111111_11111110 表示-2 此电流只在多口状态下才起作用	
--	--	--	--

5.11 VBUSIADC_DAT0（VBUS 通路 MOS 流过的电流寄存器）

寄存器地址= 0x6E default 0x00

Bit(s)	Name	Description	R/W
7:0	VBUSIADC[7:0]	VBUSIADC 数据的低 8bit（VBUS 端电流） 此电流只在多口状态下才起作用	R

5.12 VBUSIADC_DAT1（VBUS 通路 MOS 流过的电流寄存器）

寄存器地址= 0x6F

Bit(s)	Name	Description	R/W
7:0	VBUSIADC[15:8]	VBUSIADC 数据的高 8bit（VBUS 端电流） $VBUSI = VBUSIADC * LSB$ $LSB = 0.6394mA$ 对应的是 10mOhm 采样电阻，实际情况需要根据 PCB 上 VSN 到 VBUS 的走线阻抗+MOS 管导通阻抗跟 10mOhm 的倍数关系进行折算。假如实际阻抗有 20mOhm，是 10mOhm 的 2 倍，那么 LSB 也是理论值的 1/2 需要在充电状态 VINOK 和 VBUSOK 同时有效且 VBUS MOS 开启时；或者 VBUS MOS 开启的同时有其他 MOS 也开启时，该 ADC 才会启动； （补码格式，放电为正，充电为负） 例如：00000000_00000001 表示 1 11111111_11111111 表示-1 11111111_11111110 表示-2 此电流只在多口状态下才起作用	R

5.13 VOUT1IADC_DAT0（VOUT1 通路 MOS 流过的电流）

寄存器地址= 0x70

Bit(s)	Name	Description	R/W
7:0	VOUT1IADC[7:0]	VOUT1IADC 数据的低 8bit（Vout1 端电流） 此电流只在多口状态下才起作用	R

5.14 VOUT1IADC_DAT1（VOUT1 通路 MOS 流过的电流寄存器）

寄存器地址= 0x71 default 0x00

Bit(s)	Name	Description	R/W
7:0	VOUT1IADC[15:8]	VOUT1IADC 数据的高 8bit（Vout1 端电流） $VOUT1I = VOUT1IADC * LSB$ $LSB = 0.6394mA$ 对应的是 10mOhm 采样电阻，实际情况需要根据 PCB 上 VSN 到 VOUT1 的走线阻抗+MOS 管导通阻抗跟 10mOhm 的倍数关系进行折算。假如实际阻抗有 20mOhm，是 10mOhm 的 2 倍，那么 LSB 也是理论值的 1/2 需要在 VOUT1 MOS 开启的同时有其他 MOS 也开启时，该 ADC 才会启动； （补码格式，放电为正，充电为负）	R

		例如: 00000000_00000001 表示 1 11111111_11111111 表示-1 11111111_11111110 表示-2 此电流只在多口状态下才起作用	
--	--	--	--

5.15 GPIADC_DAT0 (GPIO ADC 电压值寄存器)

寄存器地址= 0x78

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

5.16 GPIADC_DAT1 (GPIO ADC 电压值寄存器)

寄存器地址= 0x79

Bit(s)	Name	Description	R/W
7:0	GPIADC [15:8]	GPIADC 数据的高 8bit $GPI = GPIADC * 0.26855mv + 1.5V$ (补码格式) 例如: 00000000_00000001 表示 1 11111111_11111111 表示-1 11111111_11111110 表示-2	R

5.17 BATOCV_DAT0 (BATOCV 电压寄存器)

寄存器地址= 0x7A

Bit(s)	Name	Description	R/W
7:0	BATOCV [7:0]	BATOCV 数据的低 8bit BATOCV 电压是 BAT 电压经过电芯内阻和电芯电流进行补偿同时有做数字滤波后的电压	R

5.18 BATOCV_DAT1 (BATOCV 电压寄存器)

寄存器地址= 0x7B

Bit(s)	Name	Description	R/W
7:0	BATOCV [15:8]	BATOCV 数据的高 8bit $OCV = BATOCV * 0.26855mv + 2.6V$ BATOCV 电压是 BAT 电压经过电芯内阻和电芯电流进行补偿同时有做数字滤波后的电压	R

5.19 POWER_DAT0 (输入输出功率寄存器)

寄存器地址= 0x7C

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

5.20 POWER_DAT1 (输入输出功率寄存器)

寄存器地址= 0x7D

Bit(s)	Name	Description	R/W
7:0	POWER [15:8]	POWER 数据的高 8bit $OCV = POWER * 8.44mW$	R

5.21 SYS_STATUS（系统状态指示寄存器）

寄存器地址= 0xD1

Bit(s)	Name	Description	R/W
7:5		Reserved	R
4	CHG_EN	当前状态指示 0: 放电状态 1: 充电状态	R
3		Reserved	R
2:0	SYS_STATE	000: 待机 001: 5V 充电 010: 单口同充同放 011: 多口同充同放 100: 高压快充充电 101: 5V 放电 110: 多口 5V 放电 111: 高压快充放电	R

5.22 KEY_IN（系统状态指示寄存器）

寄存器地址= 0xD2

Bit(s)	Name	Description	R/W
7:6		Reserved	R
5	VBUSOK	VBUS 电压有效标志, TYPEC 充电放电该 bit 都会有效 1: VBUS 有电 0: VBUS 没电	R
4	VINOK	VIN 电压有效标志 1: VIN 有电 0: VIN 没电	R
3:1		Reserved	R
0	Key_in	按键引脚的实时状态, 0 表示按键当前被按下了 1: 按键有输入 0: 按键没有输入	R

5.23 OV_FLAG（系统过压/欠压寄存器）

寄存器地址= 0xD3

Bit(s)	Name	Description	R/W
7:4		Reserved	R
3	BATLOW	BAT 电压低于 0x10 寄存器设定的低电阈值 1: BAT 电压低于 0x10 寄存器设定的低电阈值 0: BAT 电压高于 0x10 寄存器设定的低电阈值	R
2	VSYSOV	VSYS 电压高于 5.6V 1: VSYS 电压高于 5.6V 0: VSYS 电压低于 5.6V	R
1	VBUSOV	VBUS 电压高于 0x12 寄存器设定的过压阈值 1: VBUS 过压 0: VBUS 不过压	R
0	VINOV	VIN 电压高于 0x11 寄存器设定的过压阈值 1: VIN 过压 0: VIN 不过压	R

5.24 VIN_VBUS_STATE（VIN/VBUS 充电电压寄存器）

寄存器地址= 0xD5

Bit(s)	Name	Description	R/W
7:6		Reserved	R

5:3	VBUS_STATE	VBUS 充电时，输入电压所属的范围 000:5V 电压范围 001:7V 电压范围 011:9V 电压范围 111:12V 电压范围	R
2:0	VIN_STATE	VIN 充电时，输入电压所属的范围 000:5V 电压范围 001:7V 电压范围 011:9V 电压范围 111:12V 电压范围	R

5.25 CHG_STATUS（充电状态指示寄存器）

寄存器地址= 0xD7

Bit(s)	Name	Description	R/W
7	Chgop	充电工作状态： 0：可能刚好是在停充检测，也可能是充满了，也可能是异常保护了（Timer Out，或者输入 OV，或者 NTC 异常），也可能是未启动充电，具体状态需参考 sys_state、chg_state 及 NTC。 1：正在充电	R
6	Chg_end	充电充满 Charger end 标志： 0：充电未充满 1：充电已充满	R
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
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
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
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

5.26 LOW_STATUS（系统轻载标志位）

寄存器地址= 0xD9

Bit(s)	Name	Description	R/W
7		Reserved	R
6	POWLOW	当前输出总功率小于设定的小功率关机阈值 1：轻载 0：重载	R
5		Reserved	R
4	BATOCV_LOW	BAT_OCV 电压低于 0x9F 寄存器设定的低电关机阈值 1：小于 BATOCV_LOW 0：大于 BATOCV_LOW	R
3	ISYSLOW	当前输出总电流小于设定的小电流关机阈值 1：轻载 0：重载	R
2:0		Reserved	R

5.27 NTC_FLAG（NTC 状态指示寄存器）

寄存器地址= 0xDA

Bit(s)	Name	Description	R/W
7	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
6:4	NTC_IN	000: 高温 100: 中温 110: 正常温度 111: 低温	R
3:0		Reserved	R

5.28 LOWCUR_FLAG（常开 N 小时标准位）

寄存器地址= 0xDE

Bit(s)	Name	Description	R/W
7	Lowcu_en	进入常开 N 小时模式 1: 系统已经进入常开 N 小时模式 0: 系统未已经进入常开 N 小时模式 该 bit 可以指示系统是否已经进入了常开 N 小时的模式, 便于 MCU 来指示特殊的电量指示灯	R
6:0		Reserved	R

5.29 MOS_ON（路径 MOS 开启状态寄存器）

寄存器地址= 0xE5

Bit(s)	Name	Description	R/W
7	SVINVBUS	0: 当前充电使用的是 VIN 通路 1: 当前充电使用的是 VBUS 通路	R
6	VINOK_in	VIN 电压有效标志 1: 有效 0: 无效	R
5	VBUSOK_in	VBUS 电压有效标志, TYPEC 充电放电该 bit 都会有效 1: 有效 0: 无效	R
4	VIN_pmos_en	VIN MOS 开启 1: 开启 0: 未开启	R
3		Reserved	R
2	VBUS_pmos_en	VBUS MOS 开启 1: 开启 0: 未开启	R
1		Reserved	R
0	VOUT1_mos_en	VOUT1 MOS 开启 1: 开启 0: 未开启	R

5.30 BST_V_FLAG（BOOST 电压范围寄存器）

寄存器地址= 0xFB

Bit(s)	Name	Description	R/W
7:4		Reserved	R
3	BST_V_FLAG[3]	BOOST 输出大于 10V 小于等于 12V	R
2	BST_V_FLAG[2]	BOOST 输出大于 8V 小于等于 10V	R
1	BST_V_FLAG[1]	BOOST 输出大于 6V 小于等于 8V	R

0	BST_V_FLAG[0]	BOOST 快充标志	R
---	---------------	------------	---

5.31 TYPEC_OK (TYPEC 连接状态寄存器)

寄存器地址= 0XB8

Bit(s)	Name	Description	R/W
7:6		Reserved	R
5	CC_SRC_OK	TYPE-C Sink 连接成功 (CC 上拉连接成功) 1: 连接成功 0: 未连接	R
1	CC_SNK_OK	TYPE-C SRC 连接成功 (CC 下拉连接成功) 1: 连接成功 0: 未连接	R
0		Reserved	R

5.32 TYPEC_FLAG (TYPEC 上拉状态寄存器)

寄存器地址= 0xFF

Bit(s)	Name	Description	R/W
7	Snk_at_3p0a	TYPE-C 连接的电源输出能力为 3.0A 1: 有效 0: 无效	R
6	Snk_at_1p5a	TYPE-C 连接的电源输出能力为 1.5A 1: 有效 0: 无效	R
5	Snk_at_usb	TYPE-C 连接的电源输出能力为标准 USB 1: 有效 0: 无效	R

5 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。