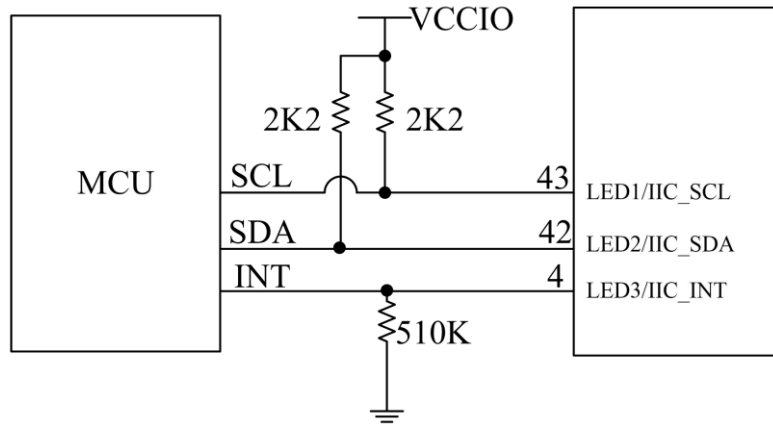


1 典型应用说明

1.1 I2C 连接方式

IP2368 可作为从设备, MCU 可通过 I2C 接口来读取或设置 IP2368 的电压、电流、功率等信息, IP2368 I2C 连接方式如下:



1.2 I2C 注意事项

- IP2368 的 I2C 设备地址: 写为 0xEA, 读为 0xEB. 如需设为其他地址, 可以通过定制实现;
- IP2368 的 I2C 通讯电压是 3.3V, 如 MCU 端是 5V 电压, 则需要加电平转换芯片, 转到 3.3V;
- IP2368 INT 应用说明: IP2368 休眠时检测到 INT 为高就会唤醒, 唤醒之后, IP2368 主动拉高 INT, 100ms 之后, MCU 可进行 I2C 通信, 进行寄存器的读写操作; IP2368 在进入休眠之前, 会切换为输入高阻来检测 INT 状态, 如果为高电平, 则认为 MCU 不允许 IP2368 进入休眠, 如果为低电平, 则 IP2368 进入休眠; MCU 在检测到 INT 为低后, 16ms 内要停止访问 IC;
- IP2368 的 I2C 最高支持 250k 通信频率, 考虑时钟偏差, 建议 MCU 的 I2C 通信时钟用 100k-200k;
- 如果要修改 IP2368 某个寄存器的值, 需要先将对应寄存器的值读出来, 然后对需要修改的 Bit 进行与或运算之后, 把计算得到的值写入该寄存器中, 其他未开放的寄存器不能随意修改. 寄存器的默认值以读到的值为准, 不同 IC 默认值可能不同;
- IP2368 I2C 通信是实时数据, 在接收到请求之后, 需要进中断进行数据准备, 准备时间较长, 所以 MCU 在 I2C 通讯时需要在发送地址后判断是否收到 ACK 和增加 50us 延时 (参考 I2C 应用示例); 建议单字节读取, 100k 的 I2C 通信频率, 每个字节之间增加 1ms 延时;
- 在 I2C 读取数据末尾, 读取完最后一个字节之后, 一定要给出 NACK 信号, 否则 IP2368 会认为还在继续读取数据, 下一个时钟会继续输出下一个数据, 导致无法收到 STOP 信号, 最后读取错误;
- 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;
- 本文档只针对 IP2368_I2C_COUT/IP2368_I2C_NACT 的型号, 其他型号无效;

1.3 I2C 应用示例

在 IP2368 INT 引脚持续为高 100ms 之后, MCU 可以进行 I2C 通信, 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 IP2368 内部信息 (电量、充放电状态、按键状态); 最后进行特殊需求的操作 (如特殊指示灯、充放电管理、快充请求管理); MCU 检测到 INT 为低后, 16ms 内需要停止访问 I2C.

例如:

往 0x05 寄存器写入数据 0x5A

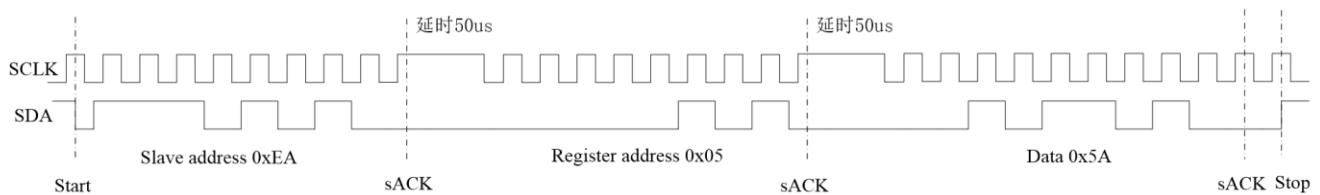


图 1 I2C Write 0x05

从 0x05 寄存器读回数据

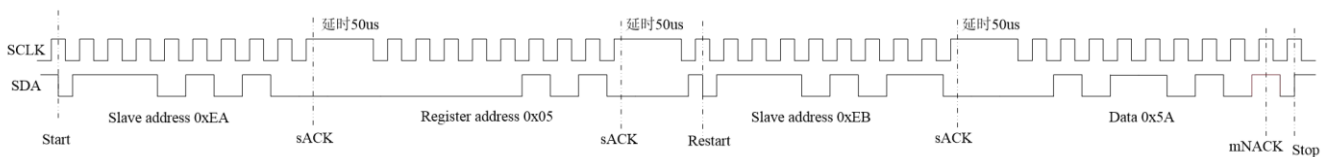


图 2 I2C Read 0x05

实际从 0x31 寄存器读回数据

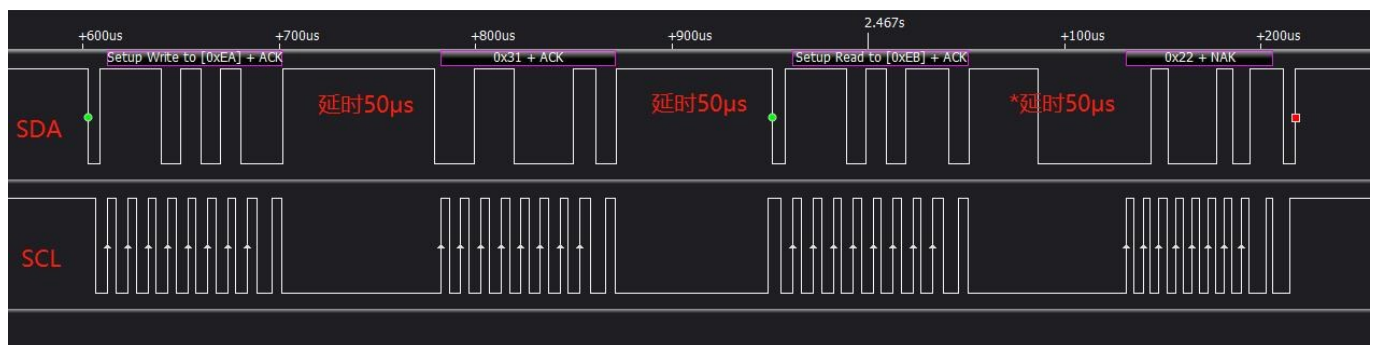


图 3 I2C Read 0x31

2 寄存器列表:

2.1 可读/写操作寄存器

[0x00] SYS_CTL0 (charge 使能寄存器)

Bit(s)	Name	Description	R/W	rst
7	En_LOADOTP	开机唤醒重新复位寄存器值使能 0: 不重新复位寄存器值 1: 重新复位寄存器值 该 bit 不建议修改为 0, 如果需要修改, 软件需要定期 复位寄存器默认值, 如 VINok VBUOk 信号触发后	R/W	1
6	En_RESETEMCU	MCU 重新复位寄存器 写 1: 重新复位寄存器为默认值, 复位后该 bit 自动恢复为 0	R/W	0
5	En_INT_low	有异常的时候 INT 拉低 2MS, 提示 MCU 有异常发生 1: Enable 0: disable	R/W	0
4	En_Vbus_SinkDPdM	C 口输入 DM DP 快充使能 1: Enable 0: disable	R/W	1
3	En_Vbus_SinkPd	C 口输入 Pd 快充使能 1: Enable 0: disable	R/W	1
2	En_Vbus_SinkSCP	C 口输入 SCP 快充使能 1: Enable 0: disable	R/W	1
1	En_Vbus_Sinkctrl	C 口 MOS 输入使能 1: Enable, 打开 C 口 MOS 0: disable, 关闭 C 口 MOS	R/W	1
0	En_Charger	Charger 充电使能 (关闭后不可充电) 1: Enable 0: disable	R/W	1

[0x01] SYS_CTL1 (串关节数设置、电池类型、电流设置模式)

Bit(s)	Name	Description	R/W	rst
7:4	Reserved			
3	En_BATmode_set	设置电池类型使能 (电池类型由寄存器 0x01[2]) 1: Enable, 允许设置电池类型 0: disable, 不允许设置电池类型	R/W	0
2	Set_BATmode	电池类型设置 0: 磷酸铁锂电池, 单节电池涓流转恒流电压 2.5V, 充满电压 3.6V 左右 1: 普通锂电池, 单节电池涓流转恒流电压 3.0V, 充满电压 4.2V 左右	R/W	1
1	En_Isetmode_set	选择电流设置模式使能 1: Enable, 允许选择电流设置模式 0: disable, 不允许选择电流设置模式	R/W	0
0	Set_Isetmode	选择电流设置模式 (电流和功率寄存器 0x03 [6:0]) 0: Iset 设置的是电池端电流 1: Iset 设置的是输入端功率	R/W	1

[0x02] SYS_CTL2 (Vset 充满电压设定)

Bit(s)	Name	Description	R/W	rst
7	En_Vset_set	设置充满电压使能 1: Enable, 允许设置 0: disable, 不允许设置 充满电压 置充满电压	R/W	0
6:0	Vset	充满电压设置 磷酸铁锂电池模式时 (0x01[2]=0), 单节电池充满电压 $Vset=N*10+3500mV$ (最高 3.7V) 普通锂电池模式时 (0x01[2]=1), 单节电池充满电压 $Vset=N*10+4000mV$ (最高 4.4V)	R/W	0x0A

[0x03] SYS_CTL3 (Iset 充电功率或电流设置)

Bit(s)	Name	Description	R/W	rst
7	En_Iset_set	设置充电功率或电流使能 1: Enable, 允许设置充电功率或电流 0: disable, 不允许设置充电功率或电流	R/W	0
6:0	Iset	电池端电流或功率设置 当设置为电池端电流时 (0x01[0]=0), 电池端电流 $Iset=N*100mA$ (最大为 5A) 当设置为充电输入功率模式时 (0x01[0]=1), 设置的 充电功率 $Pmax=N*1W$ (充电最大为 100W)	R/W	0x3C

[0x04] SYS_CTL4 (电池容量设置)

Bit(s)	Name	Description	R/W	rst
7	En_FCAP_set	设置电池容量功能使能 1: Enable, 允许设置电池容量 0: disable, 不允许设置电池容量	R/W	0
6:0	Fcap	电池容量 $FCAP= N*200mAh$	R/W	0x28

[0x06] SYS_CTL6 (当前电量)

Bit(s)	Name	Description	R/W	rst
7:0	Cap_Now	当前电量 (可读写) Cap Now=N	R/W	x

[0x07] SYS_CTL7 (涓流充电电流、阈值和充电超时设置)

Bit(s)	Name	Description	R/W	rst
7:4	ltk	涓流充电电流设置 (最大的涓流充电电流 400ma) $ltk=N*50mA$	R/W	0x04
3:2	Vtk	单节电池涓流转恒流充电电压阈值 当设置为磷酸铁锂模式时 (0x01[2]=0) 00:2.3V 01:2.4V 10:2.5V 11:2.6V 当设置为普通锂电池模式时 (0x01[2]=1) 00:2.8V 01:2.9V	R/W	10

		10:3.0V 11:3.1V		
1:0	Charge_OT	充电超时设置 00: disable, 没有充电超时功能 01:24h 10:36h 11:48h	R/W	0x02

[0x08] SYS_CTL8 (停充电流和再充电阈值设置)

Bit(s)	Name	Description	R/W	rst
7:4	Istop	停充充电电流设置 Istop=N*50mA	R/W	0x02
3:2	Vrch	再充电阈值 00: 充饱后没有再充电功能 01: VTRGT – N*0.05 10: VTRGT – N*0.1 11: VTRGT – N*0.2 VTRGT – 充饱电压 N – 电池串环节数	R/W	0x02
1:0	Reserved			

[0x09] SYS_CTL9 (待机使能和低电电压设置)

Bit(s)	Name	Description	R/W	rst
7	En_Standby	待机使能 1: 使能 0: 不使能	R/W	1
6	En_BATlow_Set	电池低电电压设置使能 (电池电压设置寄存器 0x0A) 0: disable 1: Enable	R/W	0
5	En_BAT_Low	关掉电池低电关机功能 0: disable 1: Enable	R/W	0
4:0	Reserved			

[0x0A] SYS_CTL10 (电池低电电压设置)

Bit(s)	Name	Description	R/W	rst
7:5	Set_BATlow	电池低电电压设置 000: 锂电池 2.80V*N/铁锂电池 2.3V*N 001: 锂电池 2.90V*N/铁锂电池 2.4V*N 010: 锂电池 3.00V*N/铁锂电池 2.5V*N 011: 锂电池 3.10V*N/铁锂电池 2.6V*N 100: 锂电池 3.20V*N/铁锂电池 2.7V*N N: 电池串环节数	R/W	0x02
4:0	Reserved			

[0x0B] SYS_CTL11 (输出使能寄存器)

Bit(s)	Name	Description	R/W	rst
7	En_Dc_Dc_Output	放电输出使能 (关闭后不可输出) 1: 使能 0: 不使能	R/W	1
6	En_Vbus_Src_DP_dM	C 口输出 DP/DM 快充使能 1: Enable 0: disable	R/W	1
5	En_Vbus_SrcPd	C 口输出 Pd 快充使能 1: Enable 0: disable	R/W	1
4	En_Vbus_SrcSCP	C 口输出 SCP 快充使能 1: Enable 0: disable	R/W	1
3:0	Reserved			

[0x0C] SYS_CTL12 (输出最大功率选择寄存器)

Bit(s)	Name	Description	R/W	rst
7:5	Vbus_Src_Power	Vbus1 输出功率选择: 000:20W 001:25W 010:30W 011:45W 100:60W 101:100W	R/W	0x05
4:0	Reserved			

100W 需要加 Emark 识别电路.

[0x22] TypeC_CTL8 (TYPE-C 模式控制寄存器)

Bit(s)	Name	Description	R/W	rst
7:6	Vbus_Mode_Set	Vbus CC 模式选择 00: UFP 01: DFP 11: DRP	R/W	0
5:0	Reserved			

[0x23] TypeC_CTL9 (输出 Pdo 电流设置寄存器)

Bit(s)	Name	Description	R/W	rst
7	En_5VPdo_3A/2.4A	5VPdo 电流设置 1:3A 0:2.4A	R/W	1
6	En_Pps2Pdo_Iset	Pps2 Pdo 电流设置使能 1: Enable 0: disable *使能后输出功率、过流以设置的 Pdo 电流为准, 过 流为设置 Pdo 电流 1.1 倍	R/W	0
5	En_Pps1Pdo_Iset	Pps1 Pdo 电流设置使能 1: Enable 0: disable *使能后输出功率、过流以设置的 Pdo 电流为准, 过 流为设置 Pdo 电流 1.1 倍	R/W	0

4	En_20VPdo_Iset	20VPdo 电流设置使能 1: Enable 0: disable *使能后输出功率、过流以设置的 Pdo 电流为准, 过流为设置 Pdo 电流 1.1 倍	R/W	0
3	En_15VPdo_Iset	15VPdo 电流设置使能 1: Enable 0: disable *使能后输出功率、过流以设置的 Pdo 电流为准, 过流为设置 Pdo 电流 1.1 倍	R/W	0
2	En_12VPdo_Iset	12VPdo 电流设置使能 1: Enable 0: disable *使能后输出功率、过流以设置的 Pdo 电流为准, 过流为设置 Pdo 电流 1.1 倍	R/W	0
1	En_9VPdo_Iset	9VPdo 电流设置使能 1: Enable 0: disable *使能后输出功率、过流以设置的 Pdo 电流为准, 过流为设置 Pdo 电流 1.1 倍	R/W	0
0	En_5VPdo_Iset	5VPdo 电流设置使能 1: Enable 0: disable	R/W	0

[0x24] TypeC_CTL10 (5VPdo 电流设置寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	5VPdo_Iset	5VPdo 电流设置 5VPdo=20mA*N (默认 3A,Max=3A)	R/W	0x96

[0x25] TypeC_CTL11 (9VPdo 电流设置寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	9VPdo_Iset	9VPdo 电流设置 9VPdo=20mA*N (默认 3A,Max=3A)	R/W	0x96

[0x26] TypeC_CTL12 (12VPdo 电流设置寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	12VPdo_Iset	12VPdo 电流设置 12VPdo=20mA*N (默认 3A,Max=3A)	R/W	0x96

[0x27] TypeC_CTL13 (15VPdo 电流设置寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	15VPdo_Iset	15VPdo 电流设置 15VPdo=20mA*N (默认 3A,Max=3A)	R/W	0x96

[0x28] TypeC_CTL14 (20VPdo 电流设置寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	20VPdo_Iset	20VPdo 电流设置 20VPdo=20mA*N (默认 5A, 需要识别到 emark,Max=5A) 没有识别到 emark 为 3A	R/W	0xFA

[0x29] TypeC_CTL23 (Pps1 Pdo 电流设置寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	Pps1Pdo_lset	Pps1 Pdo 电流设置 Pps1 Pdo=50mA*N (默认 5A, 需要识别到 eMark,Max=5A) 没有识别到 eMark 为 3A	R/W	0x3C

[0x2A] TypeC_CTL24 (Pps2 Pdo 电流设置寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	Pps2Pdo_lset	Pps2 Pdo 电流设置 Pps2 Pdo=50mA*N (默认 5A, 需要识别到 eMark,Max=5A) 没有识别到 eMark 为 3A	R/W	0x3C

[0x2B] TypeC_CTL17 (输出 Pdo 设置寄存器)

Bit(s)	Name	Description	R/W	rst
7	Reserved		R	R
6	En_Src_Pps2Pdo	Pps2 Pdo 使能 1: Enable 0: disable * disable 后没有 Pps2 Pdo	R/W	1
5	En_Src_Pps1Pdo En_Src_Pps1Pdo	Pps1 Pdo 使能 1: Enable 0: disable * disable 后没有 Pps1 Pdo	R/W	1
4	En_Src_20VPdo	20VPdo 使能 1: Enable 0: disable * disable 后没有 20V Pdo	R/W	1
3	En_Src_15VPdo	15VPdo 使能 1: Enable 0: disable * disable 后没有 15V Pdo	R/W	1
2	En_Src_12VPdo	12VPdo 使能 1: Enable 0: disable * disable 后没有 12V Pdo	R/W	1
1	En_Src_9VPdo	9VPdo 使能 1: Enable 0: disable * disable 后没有 9V Pdo	R/W	1
0	Reserved		R	R

2.2 只读状态指示寄存器

[0x30] SOC_CAP_DATA (电芯电量数据寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	Soc_Cap	电芯百分比电量数据 (%) Soc Cap=N	R	X

[0x31] STATE_CTL0(充电状态控制寄存器)

Bit(s)	Name	Description	R/W	rst
7:6	Reserved		R	X
5	CHG_En	充电标志位 1: 充电状态 (VbusOk 就算充电状态) 0: 非充电状态	R	X
4	CHG_End	充满状态标志位 1: 充电已充满 0: 充电未充满	R	X
3	Output_En	放电状态标志位 1: 放电状态且输出口已经打开, 没有任何异常 0: 放电状态输出没有打开或者有放电异常	R	X
2:0	Chg_state	Chg state 000: 待机 001: 涓流 010: 恒流充电 011: 恒压充电 100: 充电等待中 (包括未开启充电等情况) 101: 充满状态 110: 充电超时	R	X

[0x32] STATE_CTL1(充电状态控制寄存器)

Bit(s)	Name	Description	R/W	rst
7:6	Chg_State	Chg state 00:5V 输入充电 01: 高压输入快充充电	R	X
5:0	Reserved		R	X

[0x33] STATE_CTL2(输入 Pd 状态控制寄存器)

Bit(s)	Name	Description	R/W	rst
7	Vbus_Ok	Vbus Ok 1: Vbus 有电 0: Vbus 没电	R	X
6	Vbus_Ov	Vbus Ov 1: Vbus 输入过压 0: Vbus 输入没有过压	R	X
5:3	Reserved			X
2:0	Chg_Vbus	充电电压 111:20V 充电 110:15V 充电 101:12V 充电 100:9V 充电 011:7V 充电 010:5V 充电	R	X

[0x34] TypeC_STATE (系统状态指示寄存器)

Bit(s)	Name	Description	R/W	rst
7	Sink_Ok	TypeC Sink 输入连接标志位 1: 有效 0: 无效	R	X
6	Src_Ok	TypeC Src 输出连接标志位 1: 有效 0: 无效	R	X
5	Src_Pd_Ok Src_Pd_Ok	Src_Pd_Ok 输出连接标志位 1: 有效 0: 无效	R	X
4	Sink_Pd_Ok	Sink Pd Ok 输入连接标志位 1: 有效 0: 无效	R	X
3	Vbus_Sink_Qc_Ok	输入快充有效标志位 Qc5V 和 Pd5V 不算快充 Ok 1: 有效 0: 无效	R	X
2	Vbus_Src_Qc_Ok	输出快充有效标志位 Qc5V 和 Pd5V 不算快充 Ok 1: 有效 0: 无效	R	X
1:0	Reserved			

[0x35] MOS_STATE (输入 MOS 状态指示寄存器)

Bit(s)	Name	Description	R/W	rst
7	Reserved		R	X
6	Vbus_Mos_State	Vbus 口输入 MOS 状态 0: 关闭状态 1: 开启状态	R	X
5:0	Reserved		R	X

[0x38] STATE_CTL3 (系统过流指示寄存器)

Bit(s)	Name	Description	R/W	rst
7:6	Reserved		R	X
5	Vsys_Oc	Vsys 输出过流标志位, 需写 1 清 0 1: Vsys 输出有触发过流信号 0: Vsys 输出没有触发过流信号 系统在 600ms 内连续检测到两次以上的过流状态就认为过流有效, 并将此标志位置 1, 外部主控读取此标志位即可判断是否有过流异常发生; 从过流状态发生到系统休眠, 时间约 1.5s, 休眠后此标志位会维持为 1, 所以外部主控需要在此时间内读取标志位并作出对应的处理, 然后写 1 把标志位清 0; 如果需要判断过流状态撤销, 则需要把输出口重新打开 (把寄存器 0x22[7]先写 0 再写 1), 然后再读取状态标志位.	R	X
4		Vsys 输出短路标志位, 需写 1 清 0 1: Vsys 输出有触发短路信号 0: Vsys 输出没有触发短路信号	R	X

		系统在 600ms 内连续检测到两次以上的短路状态就认为短路有效, 并将此标志位置 1, 外部主控读取此标志位即可判断是否有短路异常发生; 从短路状态发生到系统休眠, 时间约 1.5s, 休眠后此标志位会维持为 1, 所以外部主控需要在此时间内读取标志位并作出对应的处理, 然后写 1 把标志位清 0; 如果需要判断短路状态撤销, 则需要把输出口重新打开 (把寄存器 0x22[7]先写 0 再写 1), 然后再读取状态标志位.		
3:0	Reserved		R	

[0x50] BATVADC_DAT0 (VBAT 电压寄存器)

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

[0x51] BATVADC_DAT1 (VBAT 电压寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	BATVADC[15:8]	BATVADC 数据的高 8bit VBATPIN 的电压 VBAT=BATVADC (mV)	R	X

[0x52] VsysVADC_DAT0 (Vsys 电压寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	VsysVADC[7:0]	Vsys 电压数据的低 8bit VsysPIN 的电压	R	X

[0x53] VsysVADC_DAT1 (Vsys 电压寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	VsysVADC[15:8]	Vsys 电压数据的高 8bit VsysPIN 的电压 Vsys= VsysVADC (mV)	R	X

[0x54] IVbus_Sink_IADC_DAT0 (输入电流寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	IVbus_ADC[7:0]	充电输入电流数据的低 8bit Vbus 输入的电流	R	X

[0x55] IVbus_Sink_IADC_DAT1 (输入电流寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	IVbusADC[15:8]	充电输入电流数据的高 8bit Vbus 输入的电流 Iin=IVbusADC(mA)	R	X

充电时, 电流存放在 0x54 和 0x55 中. 0x31 寄存器 bit5 是充电标志位.

[0x56] IVbus_Src_IADC_DAT0 (输出电流寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	IVbus_ADC[7:0]	放电输出电流数据的低 8bit Vbus 输出的电流	R	X

[0x57] IVbus_Src_IADC_DAT1 (输出电流寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	IVbusADC[15:8]	放电输出电流数据的高 8bit Vbus 输出的电流 Iout=IVbusADC (mA)	R	X

放电时, 电流存放在 0x56 和 0x57 中. 0x31 寄存器 bit3 是放电标志位.

[0x6E] IBATIADC_DAT0 (BAT 端电流寄存器)

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

[0x6F] IBATIADC_DAT1 (BAT 端电流寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	IBATIADC[15:8]	电芯端电流 BATIADC 数据的高 8bit IBAT= IBATIADC(mA)	R	X

[0x70] ISYS_IADC_DAT0 (IVsys 端电流寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	ISYSIADC[7:0]	IVsys 端电流 VsysIADC 数据的低 8bit	R	X

[0x71] IVsys_IADC_DAT1 (IVsys 端电流寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	IVsysIADC[15:8]	IVsys 端电流 VsysIADC 数据的高 8bit IVsys = VsysIADC(mA)	R	X

[0x74] Vsys_POW_DAT0 (Vsys 端功率寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	Vsys_POW_ADC[7:0]	Vsys 端功率 ADC 数据的低 8bit	R	X

[0x75] Vsys_POW_DAT1 (Vsys 端功率寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	Vsys_POW_ADC[15:8]	Vsys 端功率 ADC 数据的中 8bit	R	X

[0x76] Vsys_POW_DAT2 (Vsys 端功率寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	Vsys_POW_ADC[7:0]	Vsys 端功率 ADC 数据的高 8bit Vsys_POW = Vsys_POW_ADC(mW)	R	X

[0x77] INTC_IADC_DAT0 (NTC 输出电流寄存器)

Bit(s)	Name	Description	R/W	rst
7	NTC_IADC_DAT	0: 输出 20uA 1: 输出 80uA	R	X
6:0	Reserved			X

[0x78] VGPI00_NTC_DAT0 (VGPI00_NTC_ADC 电压寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	VGPI00_DAT0[7:0]	VGPI00 ADC 数据的低 8bit	R	X

[0x79] VGPI00_NTC_DAT1 (VGPI00_NTC_ADC 电压寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	VGPI00_DAT1[15:8]	VGPI00 ADC 数据的高 8bit VGPI00 DAT= VGPI00 ADC (mV)(0~3.3V)	R	X

[0x7A] VGPI01_Iset_DAT0 (VGPI01_Iset_ADC 电压寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	VGPI01_DAT0[7:0]	VGPI01 ADC 数据的低 8bit	R	X

[0x7B] VGPI01_Iset_DAT1 (VGPI01_Iset_ADC 电压寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	VGPI01_DAT1[15:8]	VGPI01_ADC 数据的高 8bit VGPI01 DAT= VGPI01 ADC (mV)(0~3.3V)	R	X

[0x7C] VGPI02_Vset_DAT0 (VGPI02_Vset_ADC 电压寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	VGPI02_DAT0[7:0]	VGPI02 ADC 数据的低 8bit	R	X

[0x7D] VGPI02_Vset_DAT1 (VGPI02_Vset_ADC 电压寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	VGPI02_DAT1[15:8]	High 8bit of VGPI02 ADC data VGPI02 DAT= VGPI02 ADC (mV)(0~3.3V)	R	X

[0x7E] VGPI03_FCAP_DAT0 (VGPI03_FCAP_ADC 电压寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	VGPI03_DAT0[7:0]	VGPI03 ADC 数据的低 8bit	R	X

[0x7F] VGPI03_FCAP_DAT1 (VGPI03_FCAP_ADC 电压寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	VGPI03_DAT1[15:8]	VGPI03 ADC 数据的高 8bit VGPI03_DAT= VGPI03_ADC (mV)(0~3.3V)	R	X

[0x80] VGPI04_BATNUM_DAT0 (VGPI04_BATNUM_ADC 电压寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	VGPI04_DAT0[7:0]	VGPI04_ADC 数据的低 8bit	R	X

[0x81] VGPI04_BATNUM_DAT1 (VGPI04_BATNUM_ADC 电压寄存器)

Bit(s)	Name	Description	R/W	rst
7:0	VGPI04_DAT0[15:8]	VGPI04_ADC 数据的高 8bit VGPI03_DAT = VGPI03_ADC (mV) (0~3.3V)	R	X

3 版本/修订历史:

版本	日期	修订内容	拟制/修订人
V1.00	2021-10-25	初版释放	IT360
V1.60	2022-05-16	修改排版和说明	IT360
V1.61	2022-07-13	增加 VSYS 功率寄存器 高 8 位	IT360
V1.62	2022-09-13	修改系统过流指示寄存器说明	IT555
V1.63	2023-11-27	修改 I2C 应用示例, 读 回数据不需要延时	IT555