

## IP2366 寄存器说明文档

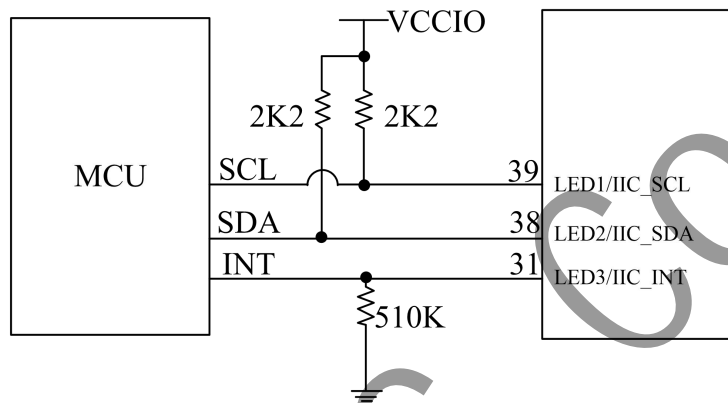
## 版本/修订历史

版本	日期	修订内容	拟制/修订人
V1.00	2023-3-24	初版释放	IT555
V1.10	2023-4-18	修改待机 EN 唤醒； 增加选择充电 PDO 档位	IT555
V1.11	2023-4-24	删除 I2C 读取电量 说明（不支持读取 电池电量）	IT555
V1.12	2023-6-13	增加待机状态充 电唤醒说明	IT555
V1.13	2023-6-26	低电电压档位最 低支持到 2.5V	IT555
V1.14	2023-10-27	涓流电流设置最 低为 100mA	IT555
V1.15	2024-2-28	低电电压默认值 修改	IT555

## 1 典型应用说明

### 1.1 I2C 连接方式

IP2366 可作为从设备，MCU 可通过 I2C 接口来读取或设置 IP2366 的电压、电流、功率等信息，IP2366 I2C 连接方式如下：



### 1.2 I2C 注意事项

- IP2366 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；
- IP2366 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；
- IP2366 INT 应用说明：IP2366 休眠时能通过充电和 EN 唤醒，唤醒之后，IP2366 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；IP2366 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 IP2366 进入休眠，如果为低电平，则 IP2366 进入休眠；MCU 在检测到 INT 为低后，16ms 内要停止访问 IC；
- IP2366 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 IP2366 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；
- IP2366 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 IP2366 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 IP2366\_I2C 的型号，其他型号无效；

### 1.3 I2C 应用示例

在 IP2366 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 IP2366 内部信息（充放电状态）；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

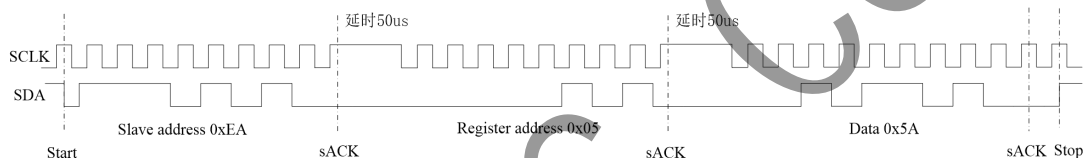


图 1 I2C Write 0x05

从 0x05 寄存器读回数据

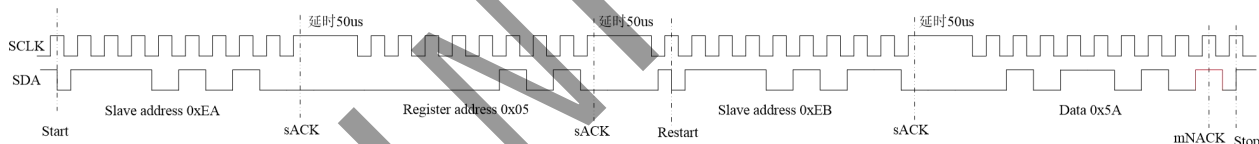


图 2 I2C Read 0x05

实际从 0x31 寄存器读回数据

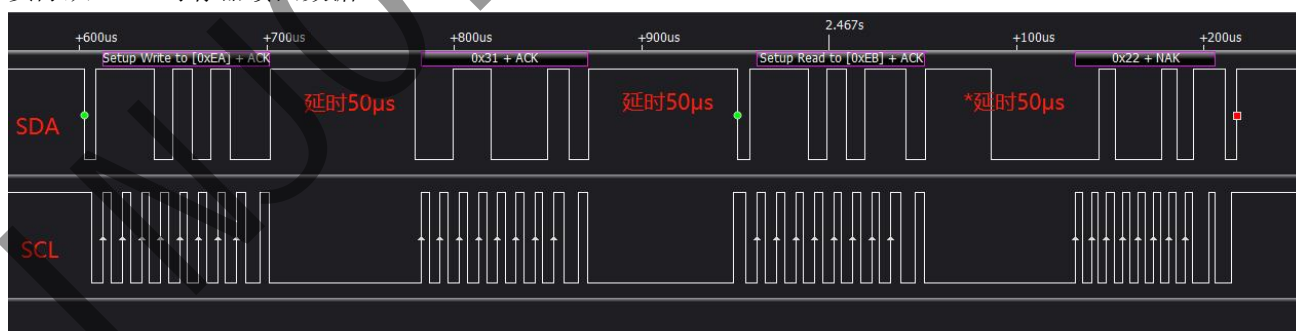


图 3 I2C Read 0x31

## 2 寄存器列表:

### 2.1 可读/写操作寄存器

(部分可通过外围配置的参数，寄存器的默认值会根据外围配置而改变)

#### (0x00) SYS\_CTL0 (charge 使能寄存器)

I2C 地址 0XEA 寄存器地址= 0x00

Bit(s)	Name	Description	R/W	RESET
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_RESETMCU	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，复位后应等待 2S 再进行读写寄存器	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	reserved		R/W	0
0	En_Charger	Charger 充电使能（关闭后不可充电） 1: Enable 0: disable	R/W	1

#### (0x02) SYS\_CTL2 (Vset 充满电压设定)

I2C 地址 0XEA 寄存器地址= 0x02

Bit(s)	Name	Description	R/W	RESET
7:0	Vset	单节电池充满电压 $Vset = N * 10mV + 2500mV$ (最高 4.4V)	RW	

**(0x03) SYS\_CTL3 (Iset 充电功率或电流设置)**

I2C 地址 0XEA 寄存器地址= 0x03

Bit(s)	Name	Description	R/W	RESET
7:0	Iset	电池端电流最大限制（不可配置小于停充电流） $Iset = N * 100mA$ （最大为 9.7A）	R/W	01100001

**(0x06) SYS\_CTL6 (涓流充电电流设置)**

I2C 地址 0XEA 寄存器地址= 0x06

Bit(s)	Name	Description	R/W	RESET
7:0	Itk	涓流充电电流设置 $Itk = N * 50mA$ （最低 100mA）	R/W	00000100

**(0x08) SYS\_CTL8 (停充电流和再充电阈值设置)**

I2C 地址 0XEA 寄存器地址= 0x08

Bit(s)	Name	Description	R/W	RESET
7:4	Istop	停充充电电流设置 $Istop = N * 50mA$	R/W	0010
3:2	Vrch	再充电阈值 00: 充电后没有再充电功能 01: $V_{TRGT} - N * 0.05$ 10: $V_{TRGT} - N * 0.1$ 11: $V_{TRGT} - N * 0.2$ $V_{TRGT}$ ---- 充电电压 N ---- 电池串环节数	R/W	10
1:0	Reserved			

**(0x09) SYS\_CTL9 (待机使能和低电电压设置)**

I2C 地址 0XEA 寄存器地址= 0x09

Bit(s)	Name	Description	R/W	RESET
7	En_Standby	待机使能 1: 使能 0: 不使能	R/W	1
6	Standby	写 1 进待机，单次有效（bit7 必须为使能才能写 1） 1: 非充电情况下立刻进入待机， 0: 正常流程	R/W	0

5	En_BAT_Low	5V 低电关机使能 0: disable 1: Enable (使能后关机电压固定为 5V, 只有软件保护, 且涓流转恒流电压也会关联变化)	R/W	0
4:0	Reserved			

### (0x0A) SYS\_CTL10 (电池低电电压设置)

I2C 地址 0XEA 寄存器地址= 0x0A

Bit(s)	Name	Description	R/W	RESET
7:5	Set_BATlow	电池低电电压设置 (涓流转恒流电压会关联变化, 2.7V 及以下设置后只有软件低电保护) 000: 2.50V*N 001: 2.60V*N 010: 2.70V*N 011: 2.80V*N 100: 2.90V*N 101: 3.00V*N 110: 3.10V*N 111: 3.20V*N N: 电池串联节数	R/W	101
4:0	Reserved			

### (0x0B) SYS\_CTL11 (输出使能寄存器)

I2C 地址 0XEA 寄存器地址= 0x0B

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

I2C 地址 0XEA 寄存器地址= 0x0C

Bit(s)	Name	Description	R/W	RESET
7: 5	Vbus_Src_Power	Vbus1 输出/输入功率选择: 000: 30W 001: 45W 010: 60W 011: 65W 100: 100W 101: 140W	R/W	101
4:0	Reserved			

**(0x0D) SELECT\_PDO (选择充电 PDO 档位)**

需要先读到 0x35 对应档位才能选择,默认选择适配器的最大 PD 档位, 适配器拔出后配置失效, 需要重新识别适配器档位再行配置。

I2C 地址 0XEA 寄存器地址= 0X0D

Bit(s)	Name	Description	R/W	Reset
7:3	Reserved			
2:0	Pdo_select	选择充电 PDO 档位 000: 默认为适配器最大档位 001: 5V 010: 9V 011: 12V 100: 15V 101: 20V	R	

**(0x22) TypeC\_CTL8 (TYPE-C 模式控制寄存器)**

I2C 地址 0XEA 寄存器地址= 0x22

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

**(0x23) TypeC\_CTL9 (输出 Pdo 电流设置寄存器)**

I2C 地址 0XEA 寄存器地址= 0x23

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

I2C 地址 0XEA 寄存器地址= 0x24

Bit(s)	Name	Description	R/W	RESET
7:0	5VPdo_Iset	5VPdo 电流设置 5VPdo=20mA*N	R/W	0x96

**(0x25) TypeC\_CTL11 (9VPdo 电流设置寄存器)**

I2C 地址 0XEA 寄存器地址= 0x25

Bit(s)	Name	Description	R/W	RESET
7:0	9VPdo_Iset	9VPdo 电流设置 9VPdo=20mA*N	R/W	0x96

**(0x26) TypeC\_CTL12 (12VPdo 电流设置寄存器)**

I2C 地址 0XEA 寄存器地址= 0x26

Bit(s)	Name	Description	R/W	RESET
7:0	12VPdo_Iset	12VPdo 电流设置 12VPdo=20mA*N	R/W	0x96

**(0x27) TypeC\_CTL13 (15VPdo 电流设置寄存器)**

I2C 地址 0XEA 寄存器地址= 0x27

Bit(s)	Name	Description	R/W	RESET
7:0	15VPdo_Iset	15VPdo 电流设置 15VPdo=20mA*N	R/W	0x96

**(0x28) TypeC\_CTL14 (20VPdo 电流设置寄存器)**

I2C 地址 0XEA 寄存器地址= 0x28

Bit(s)	Name	Description	R/W	RESET
7:0	20VPdo_Iset	20VPdo 电流设置 20VPdo=20mA*N	R/W	0xFA

**(0x29) TypeC\_CTL23 (Pps1 Pdo 电流设置寄存器)**

I2C 地址 0XEA 寄存器地址= 0x29

Bit(s)	Name	Description	R/W	RESET
7:0	Pps1Pdo_Iset	Pps1 Pdo 电流设置 Pps1 Pdo=50mA*N	R/W	0x3C

**(0x2A) TypeC\_CTL24 (Pps2 Pdo 电流设置寄存器)**

I2C 地址 0XEA 寄存器地址= 0x2A

Bit(s)	Name	Description	R/W	RESET
7:0	Pps2Pdo_Iset	Pps2 Pdo 电流设置 Pps2 Pdo=50mA*N	R/W	0x3C

**(0x2B) TypeC\_CTL17 (输出 Pdo 设置寄存器)**

I2C 地址 0XEA 寄存器地址= 0x2B

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

**(0x2C) TYPEC\_CTL18 (PDO 加 10mA 电流使能, 需要和电流设置寄存器一起配置)**

I2C 地址 0XEA 寄存器地址= 0x2C

Bit(s)	Name	Description	R/W	RESET
7:5		Reserved	R/W	0
4	EN_20VPDO_ADD	20VPDO 加 10mA 电流使能 1: enable 0: disable	R/W	0
3	EN_15VPDO_ADD	15VPDO 加 10mA 电流使能 1: enable 0: disable	R/W	0
2	EN_12VPDO_ADD	12VPDO 加 10mA 电流使能 1: enable 0: disable	R/W	0
1	EN_9VPDO_ADD	9VPDO 加 10mA 电流使能 1: enable 0: disable	R/W	0
0	EN_5VPDO_ADD	5VPDO 加 10mA 电流使能 1: enable 0: disable	R/W	0

**2.2 只读状态指示寄存器****必读部分!!!**

多个寄存器表示同一个状态的, 每次读低 8 位寄存器会更新一次高 8 位和低 8 位的数据, 所以读寄存器必须按照先读低 8 位再读高 8 位的顺序, 才能保证读到的是同一个数据。例: 读 BAT 端电压两个寄存器, 顺序应该是先读 0x50,再读 0x51。

**(0x31) STATE\_CTL0 (充电状态控制寄存器)**

I2C 地址 0XEA 寄存器地址= 0X31

Bit(s)	Name	Description	R/W
7:6	Reserved		R
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
4	CHG_End	充满状态标志位 1: 充电已充满 0: 充电未充满	R
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
2:0	Chg_state	Chg_state	R

		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	
--	--	--	--

### (0x32) STATE\_CTL1 (充电状态控制寄存器)

I2C 地址 0XEA 寄存器地址= 0X32

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

### (0x33) STATE\_CTL2 (输入 Pd 状态控制寄存器)

I2C 地址 0XEA 寄存器地址= 0X33

Bit(s)	Name	Description	R/W
7	Vbus_Ok	Vbus_Ok 1: Vbus 有电 0: Vbus 没电	R
6	Vbus_Ov	Vbus_Ov 1: Vbus 输入过压 0: Vbus 输入没有过压	R
5:3	Reserved		
2:0	Chg_Vbus	充电电压 111: 28V 充电 110: 20V 充电 101: 15V 充电 100: 12V 充电 011: 9V 充电 010: 7V 充电 001: 5V 充电	R

### (0x34) TypeC\_STATE (系统状态指示寄存器)

I2C 地址 0XEA 寄存器地址= 0X34

Bit(s)	Name	Description	R/W
7	Sink_Ok	TypeC Sink 输入连接标志位 1: 有效	R

		0: 无效	
6	Src_Ok	TypeC Src 输出连接标志位 1: 有效 0: 无效	R
5	Src_Pd_Ok	Src_Pd_Ok 输出连接标志位 1: 有效 0: 无效	R
4	Sink_Pd_Ok	Sink_Pd_Ok 输入连接标志位 1: 有效 0: 无效	R
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
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
1:0	Reserved		

### (0x35) RECEIVED\_PDO（接收 PDO 档位）

I2C 地址 0XEA 寄存器地址= 0X35

Bit(s)	Name	Description	R/W
7:5	Reserved		R
4	PDO_20V	设备接收到 PDO20V 1: 有 0: 无	R
3	PDO_15V	设备接收到 PDO15V 1: 有 0: 无	R
2	PDO_12V	设备接收到 PDO12V 1: 有 0: 无	R
1	PDO_9V	设备接收到 PDO9V 1: 有 0: 无	R
0	PDO_5V	设备接收到 PDO5V 1: 有 0: 无	R

### (0x38) STATE\_CTL3（系统过流指示寄存器）

I2C 地址 0XEA 寄存器地址= 0X38

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

7:6	Reserved		R
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	R
4	Vsys_Scdt	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 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	R
3:0	Reserved		R

### (0x50) BATVADC\_DAT0 (VBAT 电压寄存器)

I2C 地址 0XEA 寄存器地址= 0X50

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

### (0x51) BATVADC\_DAT1 (VBAT 电压寄存器)

I2C 地址 0XEA 寄存器地址= 0X51

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

### (0x52) VsysVADC\_DAT0 (Vsys 电压寄存器)

I2C 地址 0XEA 寄存器地址= 0X52

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

### (0x53) VsysVADC\_DAT1 (Vsys 电压寄存器)

寄 I2C 地址 0XEA 寄存器地址= 0X53

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

### (0x69) TIMENODE1（时间戳寄存器第一位）（时间戳符号为 ASCII 码字符）

I2C 地址 0XEA 寄存器地址= 0x69

Bit(s)	Name	Description	R/W
7:0	TimeNode1	第一个 ASCII 符号对应值	R

### (0x6A) TIMENODE2（时间戳寄存器第二位）

I2C 地址 0XEA 寄存器地址= 0x6A

Bit(s)	Name	Description	R/W
7:0	TimeNode2	第二个 ASCII 符号对应值	R

### (0x6B) TIMENODE3（时间戳寄存器第三位）

I2C 地址 0XEA 寄存器地址= 0x6B

Bit(s)	Name	Description	R/W
7:0	TimeNode3	第三个 ASCII 符号对应值	R

### (0x6C) TIMENODE4（时间戳寄存器第四位）

I2C 地址 0XEA 寄存器地址= 0x6C

Bit(s)	Name	Description	R/W
7:0	TimeNode4	第四个 ASCII 符号对应值	R

### (0x6D) TIMENODE5（时间戳寄存器第五位）

I2C 地址 0XEA 寄存器地址= 0x6D

Bit(s)	Name	Description	R/W
7:0	TimeNode5	第五个 ASCII 符号对应值	R

**(0x6E) IBATIADC\_DAT0 (BAT 端电流寄存器)**

I2C 地址 0XEA 寄存器地址= 0x6E

Bit(s)	Name	Description	R/W
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)**

I2C 地址 0XEA 寄存器地址= 0x6F

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

**(0x70) ISYS\_IADC\_DAT0 (Isys 端电流寄存器)**

I2C 地址 0XEA 寄存器地址= 0x70

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

**(0x71) Isys\_IADC\_DAT1 (Isys 端电流寄存器)**

I2C 地址 0XEA 寄存器地址= 0x71

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

**(0x74) Vsys\_POW\_DAT0 (Vsys 端功率寄存器)**

I2C 地址 0XEA 寄存器地址= 0x74

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

**(0x75) Vsys\_POW\_DAT1 (Vsys 端功率寄存器)**

I2C 地址 0XEA 寄存器地址= 0x75

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



7:0	Vsys_POW_ADC[15:8]	Vsys 端功率 ADC 数据的高 8bit Vsys_POW= Vsys_POW_ADC(10mW)	R
-----	--------------------	--	---

### (0x77) INTC\_IADC\_DAT0 (NTC 输出电流寄存器)

I2C 地址 0XEA 寄存器地址= 0X77

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

### (0x78) VGPI00\_NTC\_DAT0 (VGPI00\_NTC\_ADC 电压寄存器)

I2C 地址 0XEA 寄存器地址= 0X78

Bit(s)	Name	Description	R/W
7:0	VGPI00_DAT0 [7:0]	VGPI00_ADC 数据的低 8bit	R

### (0x79) VGPI00\_NTC\_DAT1 (VGPI00\_NTC\_ADC 电压寄存器)

I2C 地址 0XEA 寄存器地址= 0X79

Bit(s)	Name	Description	R/W
7:0	VGPI00_DAT1 [15:8]	VGPI00_ADC 数据的高 8bit VGPI00_DAT= VGPI00_ADC (mV) (0~3.3V)	R

## 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。