

AMG8802

9-17 串集成开关电源以及电流采集智能电池采集前端

1 产品特性

- 保护功能
 - 过压/欠压保护功能
 - 充电高温/低温保护功能
 - 放电高温/低温保护功能
 - 内部过温保护功能
 - 放电过流 1/2, 短路保护功能
 - 充电过流保护功能
- 电池电压平衡功能
- 电池连接断线检查和保护功能
- 负载检测, 充电器检测功能
- 电子锁控制功能
- EDSG 控制充放电管功能
- I2C 接口和 ALARM 输出, 与外部 MCU 通讯
- 通用电压数模转换器 ADC1
 - 17 路电池电压采集通道
 - 1 路电流采集通道
 - 可选 BAT, VDDA, VCC, VBO, PACK 电压
 - 1 路 PACK 电压采集通道 (可选)
 - 2 路 AUX 电压采集通道 (可选)
 - 3 路外部温度采集通道 (可选)
- 高精度电流采集和库仑计专用 ADC2
- 集成一路高压 BUCK
- MOSFET 驱动: 低边 NMOS 驱动
- 独立工作或配合 MCU 工作
- 工作模式: 正常工作模式, 休眠模式
- 封装: QFP-48

2 产品描述

AMG8802 是一款专为 9 到 17 串锂电池包设计的具有多合一功能的高可靠性、高性能的 SOC 级锂电池包主控芯片。

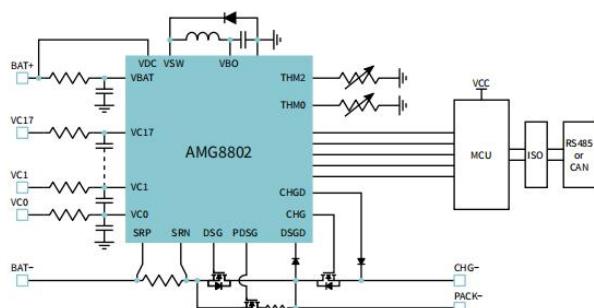
基于创新的 MagicBMS 系统架构, 为锂电池包 BMS 主板提供高可靠性、高精度、高耐压、低功耗、低 PCB 占板面积的集成化解决方案。

内部集成了高精度信号采集转换、全功能保护硬件状态机、高耐压低功耗电源转换, 无需外部 MCU 和电源转换芯片配合, 就可独立监测和保护锂电池包, 也可配合外部 MCU 对锂电池包监测, 同时实现软件保护及电量计算, 从而增加系统设计的冗余, 进一步增强保护可靠性。

AMG8802 的全功能保护硬件状态机及高精度数据采集单元, 显著降低了对外部 MCU 的性能要求, 减少了系统 BOM 成本。

AMG8802 可应用在电动自行车, 储能装置, 以及园林工具等应用场合。

料号	封装	尺寸 (典型值)
AMG8802-QPAR	QFP-48	7mm × 7mm



目录

1 产品特性	1
2 产品描述	1
3 版本	3
4 引脚图	4
5 极限电气性能	7
6 ESD性能	7
7 热性能	8
8 正常工作范围	8
9 电气参数	9
10 产品概述	15
11 功能介绍	20
12 推荐应用	48
14 封装	65
15 卷带信息	66
16 订货信息	67



3 版本

日期	版本	变更描述
2022/9/18	V1.0	简版发布

4 引脚图

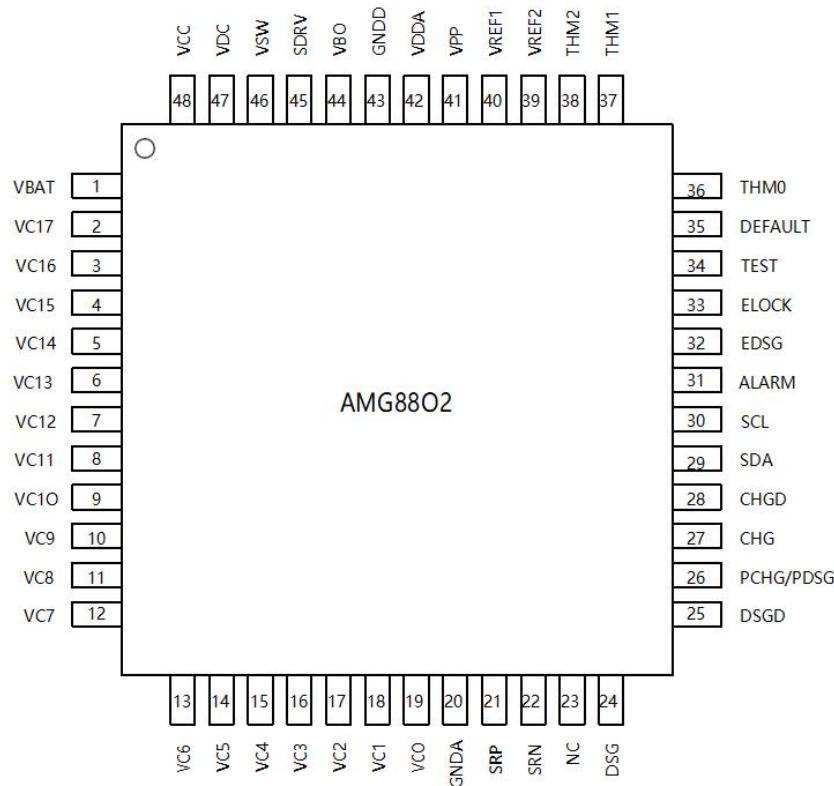


图 1: AMG8802 引脚图

引脚	名称	类型	引脚描述
1	VBAT	I	电池包电压检测端
2	VC17	I	Cell117 电池正连接端
3	VC16	I	Cell116 电池正连接端和 Cell117 电池负连接端
4	VC15	I	Cell115 电池正连接端和 Cell116 电池负连接端
5	VC14	I	Cell114 电池正连接端和 Cell115 电池负连接端
6	VC13	I	Cell113 电池正连接端和 Cell114 电池负连接端
7	VC12	I	Cell112 电池正连接端和 Cell113 电池负连接端
8	VC11	I	Cell111 电池正连接端和 Cell112 电池负连接端
9	VC10	I	Cell110 电池正连接端和 Cell111 电池负连接端
10	VC9	I	Cell119 电池正连接端和 Cell110 电池负连接端
11	VC8	I	Cell118 电池正连接端和 Cell119 电池负连接端
12	VC7	I	Cell117 电池正连接端和 Cell118 电池负连接端
13	VC6	I	Cell116 电池正连接端和 Cell117 电池负连接端

Continued on next page

引脚	名称	类型	引脚描述
14	VC5	I	Cell15 电池正连接端和 Cell16 电池负连接端
15	VC4	I	Cell14 电池正连接端和 Cell15 电池负连接端
16	VC3	I	Cell13 电池正连接端和 Cell14 电池负连接端
17	VC2	I	Cell12 电池正连接端和 Cell13 电池负连接端
18	VC1	I	Cell11 电池正连接端和 Cell12 电池负连接端
19	VC0	I	Cell11 电池负连接端
20	GNDA	G	模拟地
21	SRP	I	电流采集正端
22	SRN	I	电流采集负端
23	NC	X	NC
24	DSG	O	放电 MOS 驱动端
25	DSGD	I	负载状态检测端
26	PCHG/PDSG	O	预充/放电 MOS 驱动端
27	CHG	O	充电 MOS 驱动端
28	CHGD	I	充电器连接检测端
29	SDA	I/O	I2C 通讯接口数据线
30	SCL	I/O	I2C 通讯接口时钟线
31	ALARM	O	中断输出引脚
32	EDSG	I	放电 MOS 控制端 *
33	ELOCK	I	电子锁输入端 *
34	TEST	I	内部测试用, 请悬空或下拉
35	DEFAULT	I	DSG 缺省态配置端 *
36	THMO	O	外部温度检测端子 0
37	THM1	O	外部温度检测端子 1
38	THM2	O	外部温度检测端子 2
39	VREF2	O	电压基准输出端 2 (0.262V)
40	VREF1	O	电压基准输出端 1 (2.6215V)
41	VPP	I	OTP 烧录电源输入端
42	VDDA	O	LDO 输出端
43	GNDD	G	数字地
44	VBO	P	BUCK 输出端
45	SDRV	I	电源增强 MOS 驱动端, 建议外接 100kΩ 电阻到地

Continued on next page

引脚	名称	类型	引脚描述
46	VSW	O	BUCK 外部电感接入端
47	VDC	P	BUCK 电源输入端
48	VCC	P	芯片电源

* 相关引脚功能描述参考正文章节

5 极限电气性能

参数		最小值	最大值	单位
Input Voltage	VDC, VBAT, VCC to GNDA	-0.3	100	V
	VCO to GNDA	-0.3	6.0	V
	VC1~VC17 to GNDA	-0.3	100	V
	THMO~THM2 to GNDA	-0.3	6.0	V
	SRP, SRN to GNDA	-0.3	5.5	V
	DSGD to GNDA	-0.3	100	V
	CHGD to GNDA	VCC-100	100	V
	SCL, SDA to GNDA	-0.3	6.0	V
	EDSG, ELOCK, DEFAULT to GNDA	-0.3	6.0	V
Output Voltage	VSW to GNDA	-0.3	100	V
	VBO to GNDA	-0.3	6.0	V
	VREF1, VREF2 to GNDA	-0.3	5.5	V
	DSG to GNDA	-0.3	20	V
	PCHG/PDSG	VCC-100	VCC+0.3	V
	CHG	VCC-100	VCC+0.3	V
	ALARM to GNDA	-0.3	6.0	V
Ambient free-air temperature, T_A		-40	85	°C
Junction temperature, T_J		-40	105	°C
Storage temperature, T_{STG}			125	°C

6 ESD 性能

符号	名称	标准	能力	单位
HBM	Human Body Model ESD	ANSI/ESDEC/JEDEC JS-001-2011	2000	V
CDM	Charged Device Model ESD	JEDEC-EIA/JESD22-C101E	1000	V

- (1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
- (2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

7 热性能

封装	θ_{J^0}	θ_{J^0}	单位
QFP-48	66.0	19.6	°C /W

(1) 供设计参考使用。

8 正常工作范围

以下参数基于常温 25°C 环境

参数		测试条件	最小值	典型值	最大值	单位
VCC	芯片供电电压		6		100	V
VC _n	电池电压范围		1.5		4.5	V
T _A	环境温度		-40		85	°C

9 电气参数

如无特殊说明, 典型值的测试条件均为 $T_A=25^\circ\text{C}$, $V_{CC} = 60\text{V}$ 。最大值, 最小值参数的测试条件为 $T_A=-40^\circ\text{C}$ to 85°C , $V_{CC} = 60\text{V}$ 。

工作参数

参数	描述	测试条件	最小值	典型值	最大值	单位
VCC	工作电压		6		100	V
V_{VDDA}	芯片 VDDA 输出电压	$C_{VDDA} = 1 \mu\text{F}$, No Load		5.47	5.60	V
		$C_{VDDA} = 1 \mu\text{F}$, $I_{Load} = 5\text{mA}$	4.5			V
I_{CC_NOR}	正常工作模式下功耗	VCC=60V, $T_A=25^\circ\text{C}$, 功能全开, 250ms 扫描周期, ADC2 转换使能		80	150	μA
		VCC=60V, $T_A=25^\circ\text{C}$, 功能全开, 250ms 扫描周期, ADC2 停止转换		45	120	μA
		VCC=60V, $T_A=-40^\circ\text{C} \sim 85^\circ\text{C}$, 功能全开, 250ms 扫描周期			200	μA
I_{CC_SLP}	休眠模式下功耗	VCC=60V, $T_A=25^\circ\text{C}$, 休眠模式 2		4.0		μA
		VCC=60V, $T_A=-40^\circ\text{C} \sim 85^\circ\text{C}$, 休眠模式 2		6.0		μA
I_{LKG}	VC 引脚漏电	休眠模式 2, 无电压采样	-100		100	nA

Buck 特性

参数	描述	测试条件	最小值	典型值	最大值	单位
V_{DC}	Buck 电源输入电压		12		100	V
R_{on}	Buck 导通电阻	$I_{BO} < 330\text{mA}$		30		ohm
V_{BO}	Buck 输出电压 ($V_{BO} = 3.6\text{V}$), buck_sel = 0	No load, $V_{DC} > 24\text{V}$, HSEL=0	3.5	3.6	3.7	V
		$I_L = 30\text{mA}$, $V_{DC} = 48\text{V}$, HSEL=0	3.5	3.6	3.7	V
	Buck 输出电压 ($V_{BO} = 5.4\text{V}$), buck_sel = 1	No load, $V_{DC} > 24\text{V}$, HSEL=0	5.2	5.4	5.6	V
		$I_L = 30\text{mA}$, $V_{DC} = 48\text{V}$, HSEL=0	5.2	5.4	5.6	V
I_{BO}	带负载能力	$L = 22 \mu\text{H}$, $V_{DC} > 48\text{V}$, HSEL=1			160	mA

电压采样精度

参数	描述	测试条件	最小值	典型值	最大值	单位
V _{ACC_IC}	单体电压测量精度 (芯片) ⁽¹⁾	2.5V ≤ VC _n - VC _{n-1} ≤ 4.0V, T _A =25°C	-1.5		1.5	mV
		2.5V ≤ VC _n - VC _{n-1} ≤ 4.0V, T _A =-10°C ~ 60°C	-2.0		2.0	mV
		2.0V ≤ VC _n - VC _{n-1} ≤ 4.2V, T _A =25°C	-2.0		2.0	mV
		2.0V ≤ VC _n - VC _{n-1} ≤ 4.2V, T _A =-10°C ~ 60°C	-4.0		4.0	mV
		2.0V ≤ VC _n - VC _{n-1} ≤ 4.2V, T _A =-40°C ~ 85°C	-7.0		7.0	mV
V _{ACC_BRD}	单体电压测量精度 (板级) ⁽²⁾	2.0V ≤ VC _n - VC _{n-1} ≤ 4.2V, T _A =25°C	-2.0		2.0	mV
		2.0V ≤ VC _n - VC _{n-1} ≤ 4.2V, T _A =-10°C ~ 60°C	-5.0		5.0	mV
		2.0V ≤ VC _n - VC _{n-1} ≤ 4.2V, T _A =-40°C ~ 85°C	-10.0		10.0	mV
V _{RES_IC}	单体电压采样分辨率			±0.16		mV
V _{RES_BAT}	总压测量分辨率			±3.2		mV
V _{ACC_BAT}	总压测量精度	V _{BAT} = 69.6V, T _A = 25°C	-0.6		0.6	V
		V _{BAT} = 69.6V, T _A = 0°C ~ 60°C	-1.0		1.0	V
		V _{BAT} = 69.6V, T _A = -40°C ~ 85°C	-1.2		1.2	V

(1) 基于单体芯片测试数据

(2) 芯片经过回流焊温度后, 板级精度会出现偏移。建议通过板级的多点校准, 提升板级精度。上表中参数范围为板级校准后精度。

电流采样精度

参数	描述	测试条件	最小值	典型值	最大值	单位
V _{CURR_ADC2}	ADC2 电流采样输入范围	V _{SRP} - V _{SRN}	-200		200	mV
V _{ZDF_ADC2}	ADC2 采样零点漂移	T _A =25°C	-40.0	±10	40.0	μV
V _{ACC_ADC2}	ADC2 采样精度 (板级)	V _{SRP} - V _{SRN} = 5.0mV, T _A = 25°C		±1.2		%
		V _{SRP} - V _{SRN} = 5.0mV, T _A = -40°C ~ 85°C		±1.4		%
		V _{SRP} - V _{SRN} = 7.5mV, T _A = 25°C		±0.8		%
		V _{SRP} - V _{SRN} = 7.5mV, T _A = -40°C ~ 85°C		±1.0		%
		V _{SRP} - V _{SRN} = 15mV, T _A = 25°C		±0.55		%
		V _{SRP} - V _{SRN} = 15mV, T _A = -40°C ~ 85°C		±0.65		%
V _{RES_ADC2}	ADC2 采样分辨率			±0.5		μV

Continued on next page

参数	描述	测试条件	最小值	典型值	最大值	单位
V_{CURR_ADC1}	ADC1 电流采样输入范围	PGA 关闭时 $ V_{SRP} - V_{SRN} $	-200		200	mV
		PGA 开时 (默认) $ V_{SRP} - V_{SRN} $	-200		200	mV
V_{ZDF_ADC1}	ADC1 采样零点漂移	$T_A = 25^\circ C$	-0.95	± 1	1.05	mV
V_{ACC_ADC1}	ADC1 采样精度	$ V_{SRP} - V_{SRN} = 15\text{mV}$, $T_A = -40^\circ C \sim 85^\circ C$		± 5		%
V_{RES_ADC1}	ADC1 采样分辨率			± 2.5		μV
V_{ZDF_COMP1}	放电电流比较器零点漂移		-2.5		2.5	mV
V_{ZDF_COMP2}	过流 2 保护比较器零点漂移		-5		5	mV
V_{ZDF_COMP3}	短路保护比较器零点漂移		-20		20	mV

温度采样

参数	描述	测试条件	最小值	典型值	最大值	单位
V_{THM}	温度测量输入端电压		0.0		2.6215	V
I_{SRC_LT}	输出电流当测量低温时			12.0		μA
I_{SRC_HT}	输出电流当测量高温时			100.0		μA
V_{ACC_THM}	温度采样精度	$T_A = 25^\circ C$		± 2.0		mV
R_{12K}	内部 12K 参考电阻	$T_A = 25^\circ C$		12.0		$k\Omega$
		$T_A = -40^\circ C \sim 85^\circ C$		± 8.0		%

REF 参数

参数	描述	测试条件	最小值	典型值	最大值	单位
V_{REF1}	VREF1 电压	$T_A = 25^\circ C$	2.620	2.621	2.623	V
V_{REF2}	VREF2 电压	$T_A = 25^\circ C$		0.262		V
V_{VREF1}	VREF1 温漂	$T_A = -40^\circ C \sim 85^\circ C$		± 15.0		ppm/°C
V_{VREF2}	VREF2 温漂	$T_A = -40^\circ C \sim 85^\circ C$		± 15.0		ppm/°C

驱动参数

参数	描述	测试条件	最小值	典型值	最大值	单位
I_{CHG}	CHG on 输出电流	驱动电流配置可选范围	10.0		50.0	μA
		OPTION[11:10] = 0x00, $I_{CHG} = 10 \mu A$	9.0	10.0	11.0	μA
V_{DSG}	DSG on 输出电压	$T_A = 25^\circ C$	8.0	12.0	15.0	V
V_{DSG}	DSG off 电压	$T_A = 25^\circ C$			100.0	mV
I_{DSG}	DSG off 电流	$T_A = 25^\circ C$		7.5		mA
T_{off}	EDSG 的 PWM 控制 DSG 的开关延时			200		ns

充电器/负载检测参数

参数	描述	测试条件	最小值	典型值	最大值	单位
V_{CHGIN}	充电器插入识别阈值			-1.0	-0.6	V
V_{LDIN}	负载插入识别阈值		1.1			V

均衡参数

参数	描述	测试条件	最小值	典型值	最大值	单位
I_{BAL}	内部均衡电流	$R_{FL} = 100 \Omega$, $VC_n - VC_{n-1} = 4.0V$			10	mA

断线诊断参数

参数	描述	测试条件	最小值	典型值	最大值	单位
V_{OW_TH}	断线诊断触发电压	$T_A = 25^\circ C$	0.75	0.8	0.85	V
V_{OW_RLS}	断线诊断释放电压	$T_A = 25^\circ C$	0.95	1	1.05	V
t_{OW}	断线诊断周期	$T_A = 25^\circ C$	60	64	68	s

IO 参数

参数	描述	测试条件	最小值	典型值	最大值	单位
V_{IH}	输入高阈值电压	DEFAULT, TEST, ELOCK, EDSG, SDA, SCL 输入高	$0.7 \times V_{BO}$	V_{BO}		V
V_{IL}	输入低阈值电压	DEFAULT, TEST, ELOCK, EDSG, SDA, SCL 输入低			$0.2 \times V_{BO}$	V
V_{OH}	输出高阈值电压	ALARM, SDA 输出高	$0.7 \times V_{BO}$	V_{BO}		V
V_{OL}	输出低阈值电压	ALARM, SDA 输出低			$0.2 \times V_{BO}$	V

I²C 参数

参数	描述	测试条件	最小值	典型值	最大值	单位
f_{SCL}	时钟频率	SCL duty =50%			100	kHz
$t_{HD:STA}$	Start 信号 Hold 时间		4.0			μs
t_{LOW}	SCL 时钟低信号周期		4.7			μs
t_{HIGH}	SCL 时高信号周期		4.0			μs
$t_{SU:STA}$	重新开始信号 Repeat Start 信号建立时间		4.7			μs
$t_{HD:DAT}$	SDA 输入数据 Hold 时间		0			ns
$t_{SU:DAT}$	SDA 输入数据建立时间		250			ns
t_r	SCL 时钟上升时间	10% to 90%			1000	ns
t_f	SCL 时钟下降时间	90% to 10%			300	ns
$t_{SU:STOP}$	STOP 信号建立时间		4.0			μs
t_{BUF}	总线空闲时间		4.7			μs
t_{RST}	I ² C 总线复位时间		1.9		2.1	s
R_{PU}	上拉电阻		1.5			kΩ

OTP 参数

参数	描述	测试条件	最小值	典型值	最大值	单位
V_{VPP}	OTP 烧写电压			7.2	7.4	V
	OTP 读取电压			5.0	5.6	V

10 产品概述

AMG8802 内部集成了两个独立的 ADC，允许对最多 17 串电池电压进行采样，以及对温度输入，和电流进行采样。AMG8802 支持独立工作模式，可以在没有 MCU 控制的前提下，通过集成的低边充放电管的驱动输出，自主进行电池包保护和释放的管理。

AMG8802 支持在 MCU 控制下，获取更高精度的电流计量值，以及对预充、预放以及 MOSFET 的控制。

AMG8802 集成了一个低功耗的高压 Buck 转换器，可以直接从高压电池包降压为 5.4/3.6V，为 MCU，RS485，CAN 等芯片供电。

10.1 内部框图

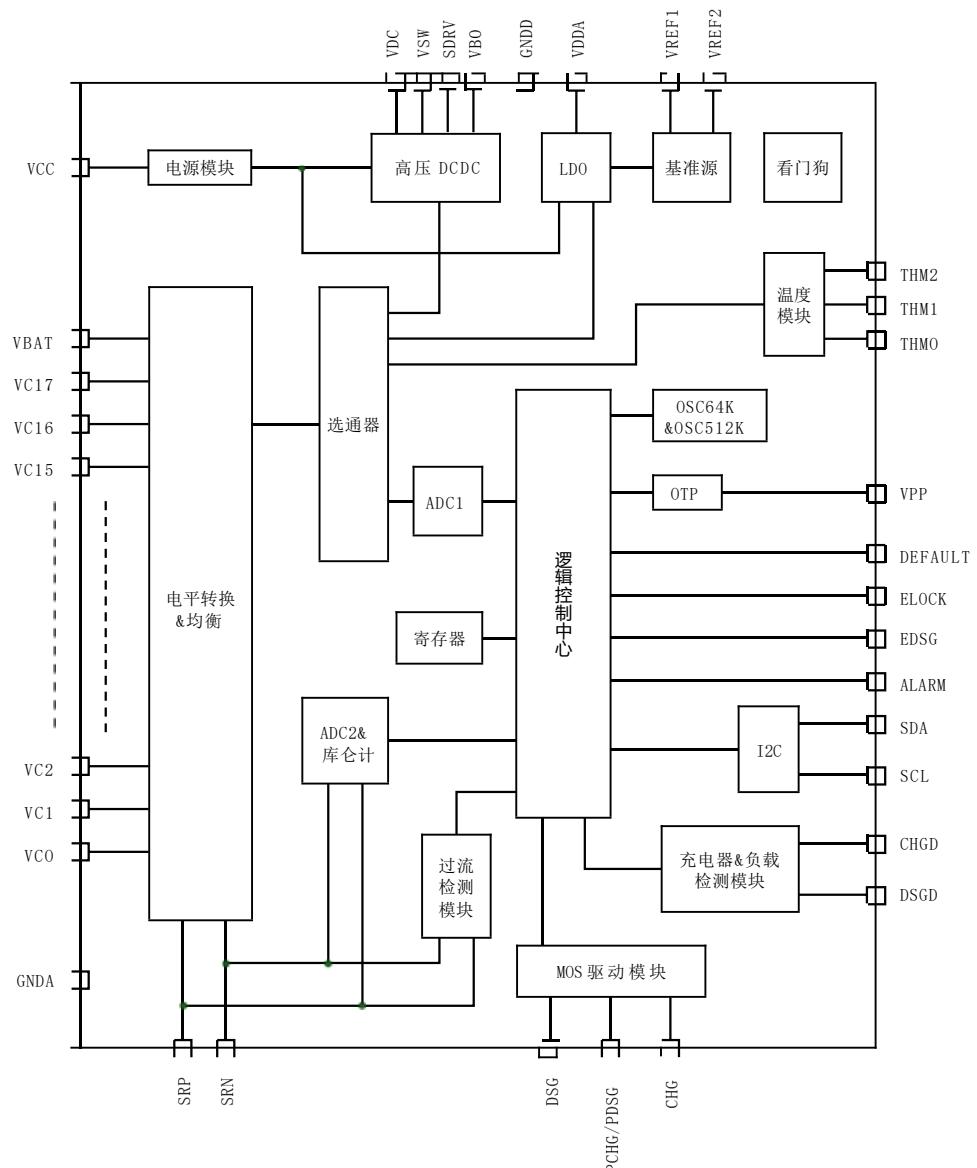


图 2: AMG8802 内部框图

10.2 电源部分

10.2.1 内部 LDO

芯片内部集成了一路典型值 5.4V 输出的 LDO，提供给芯片内部使用。LDO 输入为 VCC 引脚，输出为 VDDA 引脚。同时，也可以给外部提供有限的供电（小于 5mA）。

VCC 连接电池总正电压，最低工作电压不低于 6V。建议 VCC 引脚前串联连接防反二极管，以及 RC 滤波电路。在 VCC 引脚连接 TVS 保护管，增加对 VCC 引脚的输入防护能力。建议 VCC 引脚的 RC 滤波参数选择 51Ω 电阻和 $2.2\mu F/100V$ 电容。

输出 VDDA 引脚建议连接 $2.2\mu F$ 和 $0.1\mu F$ 滤波电容。

10.2.2 高压 Buck

芯片内部集成了一路高压 buck 控制器，输出的典型值为 3.6V。可以通过配置寄存器 [CFGLOCK\[5\]](#) ([0x1F](#), [buck_sel](#))，调整输出为 5.4V。

通过 [0xXXA9](#) 写入寄存器 [BUCKDOWN \(0xBA\)](#)，即可关断 Buck。关断 Buck 后，芯片将进入最低功耗 模式，通过充电器插入，可以唤醒芯片。芯片唤醒后，Buck 重新使能。

Buck 最大的输出电流约为 160mA (输入电压 $VDC > 45V$ ，外接电感 $L = 22\mu H$ ，输出电压 3.6V)，若输入电压下降，则输出电流能力会下降。空载时，输入端的功耗约为 $13\mu A$ 。

Buck 的输入管脚为 VDC，VDC 连接电池总正电压，最低工作电压不低于 18V。建议 VDC 引脚前串联连接防反二极管。在 VDC 引脚连接 TVS 保护管，增加对 VDC 引脚的输入防护能力。适当增加 VDC 管脚的电容可以提供 Buck 的性能。

VSW 为内部集成的上管输出引脚，此管脚外接肖特基二极管和电感。肖特基推荐使用耐压比较高的二极管（至少高于 VDC 的 1.2 倍），电感推荐使用 $22\mu H$ 。Buck 的电压反馈端为管脚 VBO，VBO 同时也是输出电压端，此管脚应加 $10\mu F$ 和 $0.1\mu F$ 滤波电容。其典型的应用图 3 所示：

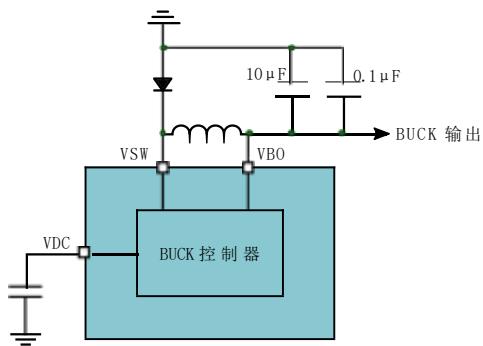


图 3: 典型高压 Buck 应用图

10.2.3 编程电压 VPP

芯片内部集成 OTP，用来存储参数部分，OTP 对应的寄存器空间为 [0x10~0x2F] 在芯片上电后，芯片在启动以后，OTP 的内容会被映射到寄存器地址空间 [0x40~0x5F]。

VPP 是内部 OTP 的供电电源，在对 OTP 进行读操作时，需要提供 VDDA 电压到此管脚；对 OTP 进行编程操作时，需要在 VPP 管脚加 7.2V 的编程电压。

10.3 参考电压

芯片内部集成了两路参考电压，分别为 VREF1 和 VREF2，其电压典型值分别为 2.6215V 和 0.2622V。

参考源 VREF1 主要用于作为 ADC1 的参考电压，其通过 VREF1 管脚输出，此管脚需加上 $2.2 \mu F$ 和 $0.1 \mu F$ 的滤波电容。VREF1 上电默认使能输出。

参考源 VREF2 主要用于作为 ADC2 的参考电压，其通过 VREF2 管脚输出，此管脚需加上 $2.2 \mu F$ 和 $0.1 \mu F$ 的滤波电容。只有当 ADC2 使能时，VREF2 才会开启。请阅读 ADC2 相关章节了解如何开启 ADC2。

10.4 ADC

芯片内部集成了 2 路 ADC，分别为 ADC1 和 ADC2。

10.4.1 ADC1

ADC1 的电压参考源为 VREF1 (2.6215V)，时钟源为 512KHz。

ADC1 采集信号的输入可选如下表所列：

序号	信号输入源
1	电池电压采样 $VC_n - VC_{n-1}$, $n = 1$ to 17
2	温度采样 THM0 ~ THM2
3	电流采样 $ SRP - SRN $
4	内部参考电阻上的电压采样
5	VCC 电压采样
6	VBAT 电压采样
7	VDDA 电压采样
8	VBO 电压采样
9	DSGD 管脚电压采样
10	THM1 管脚电压采样
11	THM2 管脚电压采样

ADC1 可以按照配置自动进行周期采样，也可以软件触发采样。当其在自动采样时，如果有 MCU 触发采样的请求，ADC1 会在下一次自动采样前，执行触发采样命令，进行采样。建议 MCU 及时读取采样转换的结果，防止被下一次自动转换结果覆盖。

电压，温度保护依靠 ADC1 的采样结果进行判断。

自动采样

在正常模式下，ADC1 自动采样指的是芯片根据配置，以固定的采样周期进行采样。ADC1 的自动采样周期可以通过寄存器 `CBCFG[15:14]` (0x1D, `chk_period[1:0]`) 设置，可选 4 种采样周期，分别为：125ms、250ms、500ms、1s。

设置寄存器 `OTP_MD[9]` (0xF0, `auto_chk_stop`) 为 1 时，禁止自动采样。

通过寄存器 `OPTION[7:6, 4]` (0x1E, `adc1_crst_lsb[1:0]`, `adc1_vltg_lsb`)，ADC1 采样电压通道可以配置为 14bit 或 16bit，ADC1 采样电流通道可配置为 14bit、16bit 和 18bit。

ADC1 采样完成后，会更新相应的寄存器，用户可以通过 I2C 通讯读取相关采样值。

在休眠模式 1 下，ADC1 会以 2 秒或者 4 秒的周期被唤醒自动采样一次，对休眠模式 1 状态下的安全保护事件监测。

在休眠模式 2 下，ADC1 自动采样被关闭。

软件触发采样

芯片除了自动采样外，还可以配置寄存器进行手动触发采样，手动触发采样不会影响自动采样，芯片把触发采样的请求插入到某一时刻自动采样时序结束后。

通过寄存器 [ADC1REQ \(0x70\)](#)，设置采样位数，采样通道，以及采样的相关配置。当配置此寄存器开始采样命令后，触发采样将插入时序。完成此次采样后，更新结果到相关的寄存器，通过 I2C 通讯读取。

10.4.2 ADC2

ADC2 的参考源为 VREF2，时钟源为 64K。ADC2 只用来做电流采样，电流采样结果将用于库仑计的累加源。

ADC2 上电默认关闭，可以通过设置寄存器 [ADC2EN\[0\] \(0xD0, sw_adc2_en\)](#) 进行使能。ADC2 只在正常模式下使能，在休眠模式下被禁止。

通过寄存器 [ADC2EN\[1\] \(0xD0, adc2_1sb\)](#)，ADC2 的精度可以选择 18bit 或者 20bit。选择 18bit 时，采样周期为 62.5ms，选择 20bit 时，采样周期为 125ms。

11 功能介绍

11.1 工作模式

芯片可以支持 2 种低功耗模式，可通过设置寄存器 CFGLOCK 的 sleep_option[1:0] (0x1F [bit7:bit6]) 来设置低功耗模式：

‘00’：不支持休眠；

‘01’：支持休眠模式 2；

‘10’或‘11’：支持休眠模式 1。

下表例举了系统在正常模式、休眠模式 1、休眠模式 2 的功能模块工作情况：

VCC 平均功耗	正常模式	休眠模式 1	休眠模式 2	深度休眠
	120 μ A	60 μ A	4.7 μ A	2.8 μ A
ADC1	保持	每 2 秒/4 秒检测一次	关闭	关闭
ADC2	保持	关闭	关闭	关闭
DSG	保持	保持	关闭	关闭
CHG	保持	保持	关闭	关闭
PCHG/PDSG	保持	保持	关闭	关闭
充电器唤醒	无	无	支持	支持
负载唤醒	无	无	支持	关闭
充放电电流唤醒	无	支持	关闭	关闭
I2C	保持	保持	保持	关闭

11.1.1 休眠模式 1

休眠模式 1 指的是电池处于不充不放的静置状态时，系统前端降低采集周期的的一种正常状态，可通过充放电电流或者通讯控制进行退出。设置寄存器 CFGLOCK[7:6] (0x1F, sleep_option[1:0]) 为‘10’或者‘11’。设置好这个休眠模式后，在以下条件都满足时，会进入休眠模式 1：

- 系统当前的状态为“不充不放”；
- 当前系统不处于过温或者低温保护状态；
- 没有均衡。

当满足以上条件后，经过 15 秒进入休眠模式 1。休眠模式 1 不强制关充放电管，会保持原来的状态。

sleep_option[1:0]设置为‘10’时，进入休眠模式 1 后，内部 ADC1 会每 2s 做一次检测；
sleep_option[1:0] 设置为‘11’时，进入休眠模式 1 后，内部 ADC1 会每 4s 做一次检测。

使用 I2C 通讯设置寄存器 SLPWKUP (0xBB) 位，写入 0xEE，可让硬件快速进入休眠模式，而不管硬件的休眠条件是否已经满足；在休眠模式下，写入 0xCC 可以快速唤醒，包括唤醒由本器件硬件进入的休眠。

满足以下任意条件时，芯片会被唤醒，进入正常工作模式：

- 如果寄存器 SWCFG[12] (0xB5, inchg_wkup_en) 为 1, 即允许充电电流唤醒, ADC1 每 2 秒或者 4 秒检测后, 如果系统状态是“充电状态”, 系统会被唤醒。
- 如果寄存器 SWCFG[13] (0xB5, indsg_wkup_en) 为 1, 即允许放电电流唤醒, ADC1 每 2 秒或者 4 秒的检测, 如果检测到系统状态是“放电状态”, 系统会被唤醒。当放电电流阈值大于寄存器 OPTION[3:2] (0x1F, indsg_th[1:0]) 所设置的阈值, 系统立即被唤醒。请参考“放电过流 2 和短路的检测与保护模块”章节了解此阈值的解释。

注: 关于系统的“不充不放状态”或者“充电状态”或者“放电状态”, 请参考“ADC1 的电流测量”章节内容。

11.1.2 休眠模式 2

设置寄存器 CFGLOCK[7:6] (0x1F, sleep_option[1:0]) 为‘01’, 在以下条件都满足时, 会进入休眠模式 2:

- 系统当前的状态位“不充不放”;
- 当前系统不处于过温或者低温保护状态;
- 当前系统不处于内部过热保护状态;
- 当前系统不处于断线保护状态;
- 当前系统不处于充放电过流保护或短路保护状态;
- 没有均衡。

当满足以上条件后, 经过 15 秒进入休眠模式 2。休眠模式 2 强制关充放电管。

在 Buck 没有被软件关闭时, I2C 是可以通讯的。使用 I2C 通讯设置寄存器 SLPWKUP (0xBB) 位, 写入 0xEE, 可让硬件快速进入休眠模式, 而不管硬件的休眠条件是否已经满足; 在休眠模式下, 写入 0xCC 可以快速唤醒, 包括唤醒由本器件硬件进入的休眠。

满足以下任意条件时, 芯片会被唤醒, 进入正常工作模式:

- 如果寄存器 SWCFG[14] (0xB5, chgrin_wkup_en) 为 1, 即允许充电器插入唤醒, 如果此时不处于过压保护状态, 插入充电器后, 系统会被唤醒。
- 如果寄存器 SWCFG[15] (0xB5, ldon_wkup_en) 为 1, 即允许负载插入唤醒, 如果此时不处于过放保护状态, 插入负载后, 系统会被唤醒。

11.1.3 深度休眠

如发生内部过热 OHT 事件时, 也会强制关闭 Buck, 等待内部过热事件释放, OHT 释放之后 16ms 打开 Buck, 大约 4ms 之后恢复现场。

通过写 0xA9 到寄存器 BUCKDOWN (0xBA), 芯片关闭充放电管以及预充(放), 停止 ADC 采集和其他功能, 等待 16ms 后, Buck 关闭, 进入深度休眠。此时, I2C 通讯被禁止。系统只开启充电器检测功能, 一旦检测到充电器插入, 会重新打开 Buck, 然后等待 4ms, 恢复现场。

11.1.4 上电后初始状态

Default 用于配置上电后, 芯片是否默认处于欠压(过放)保护状态。Default 管脚电平为高, 芯片上电后, 默认进入过放保护状态, 即放电管驱动 DSG 管脚将输出低电平, 关断放电 MOS。需要满足过放保护条件后, 才会恢复正常状态。

上电时 Default 管脚电平为低, 默认上电后进入正常状态。

11.2 电压采集

芯片可以测量 9~17 串电池电压和一路总压，电池电压测量信号通过 VC0~VC17 输入，总压信号通过 VBAT 输入。单体电压和总压采集输入前端的推荐滤波电路如图 4 所示。推荐前端滤波电路使用 $1\text{k}\Omega$ 电阻和 22nF 电容。

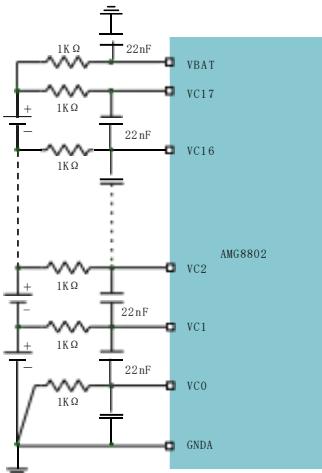


图 4: 电池电压输入滤波电路

当采样串数少于 17 串时，参考下表：

电池串数	悬空的管脚	使用的管脚
9 节电池	VC3 - VC10 与 VC2 短接一起	VC1 - VC2, VC11 - VC17
10 节电池	VC3 - VC9 与 VC2 短接一起	VC1 - VC2, VC10 - VC17
11 节电池	VC3 - VC8 与 VC2 短接一起	VC1 - VC2, VC9 - VC17
12 节电池	VC3 - VC7 与 VC2 短接一起	VC1 - VC2, VC8 - VC17
13 节电池	VC3 - VC6 与 VC2 短接一起	VC1 - VC2, VC7 - VC17
14 节电池	VC3 - VC5 与 VC2 短接一起	VC1 - VC2, VC6 - VC17
15 节电池	VC3 - VC4 与 VC2 短接一起	VC1 - VC2, VC5 - VC17
16 节电池	VC3 与 VC2 短接一起	VC1 - VC2, VC4 - VC17
17 节电池		VC1 - VC17

单体电压采样范围为 $1.5\text{~}5\text{V}$ 。芯片内部使用 ADC1 进行电压采样并转换。

自动采样的模式下，芯片会根据配置的串数进行相应的采样。转换完成后，电芯电压的结果位于寄存器 CELL01~CELL17 (0x91~0xA1) 里面， $\text{LSB} = 0.16\text{mV}$ 。

手动触发采样时，可设置寄存器 [ADC1REQ \(0x70\)](#)，设置转换电芯节数，转换精度等。转换完成后，更新标志位和相应的转换结果寄存器。

总压需要手动触发测量，设置寄存器 [ADC1REQ \(0x70\)](#)，转换的结果位于寄存器 [VBAT \(0xA8\)](#) 里面， $\text{LSB} = 3.2\text{mV}$ 。

11.3 电流采集

电流采样输入引脚为 SRN 和 SRP，信号大小为 $\pm 200\text{mV}$ 。采样电路需要对电流信号进行滤波处理，推荐的电路和内部框如图 5。

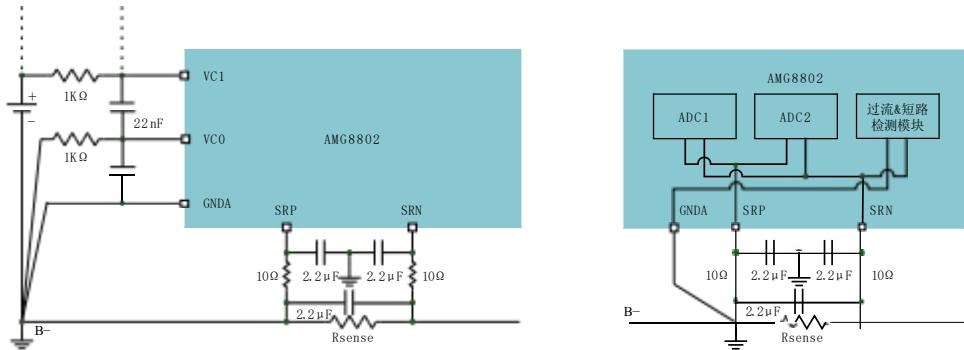


图 5：电流采样推荐电路设计以及内部结构示意图

图 5 为电流采样的推荐电路，电流采样电阻 Rsense 两端，经过滤波网络输入到 SRP 和 SRN 两端，推荐的 RC 滤波网络参数为 10Ω 和 $2.2\mu\text{F}$ 。电流采样管脚的信号会进入三个模块，作为 ADC1 的信号源，作为 ADC2 的信号源，也是内部过流 & 短路检测模块的输入源。

11.3.1 ADC1 电流采集

电流信号可以使用 ADC1 进行采样。ADC1 可以配置为自动电流检测。自动检测电流时，采样精度可以设置为 14bit、16bit 或者 18bit。可通过设置寄存器 `OPTION[7:6]` (`adc1_crrt_1sb[1:0]`) 设置精度。

‘00’ / ‘01’：14 位精度；

‘10’：16 位精度；

‘11’：18 位精度。

测量完成后，电流结果更新在寄存器 `CRRT0` (0xA5) 和 `CRRT1` (0xA6)，两个寄存器共组成一个 18bit 的转换结果，最高位为符号位， $\text{LSB}=2.5\mu\text{V}$ 。

芯片允许使用手动触发 ADC1 的电流采样，设置寄存器 `ADC1REQ` (0x70) 相关的位，转换完成后，会更新相应的标志位和结果寄存器。

可以通过设置寄存器 `CFGLOCK[4:0]` (0x1F, `idle_range[4:0]`) 来设置充放电状态阈值。当采样值绝对值在寄存器设定的范围内，认为电池处于静置状态。当超过该范围，则根据采样的结果符号，判定芯片处于充电状态还是放电状态。

当充放电状态更新时，会产生相应的标志位 `FLAG2[3, 2, 1]` (0xC3, `dischg_flag, chg_flag, idle_flag`)。如果使能 `IE2[3, 2, 1]` (0xC4, `dischg_ie, chg_ie, idle_ie`) 中断输出，在产生这些标志时，`Alarm` 管脚输出中断，直到标志位被清零（写 1 清零）。读取寄存器 `STATUS2[3, 2, 1]` (0xC5, `dischg_status, chg_status, idle_status`)，获取当前充放电状态。

AMG880x 系列在 ADC1 采集电流信号，即 SRP—SRN 电压差时，是通过内部 PGA 将 Shunt 电阻的电压差信号放大到合适的电压范围，进行 ADC1 转换。由于 PGA 在高低温下存在较大的温度系数偏移，导致测量时出现采集信号偏移。

配置 0xE4 寄存器为 0x0104，关闭 PGA 功能，降低 ADC1 在测量电流时的温度偏差。关闭 PGA 之后，

ADC1 的电流转化结果可根据以下函数进行修正。

$$V_{\text{Sense}} = \frac{\text{CRRT_ADC1} - (\text{CRRT_OFST} \times 4)}{K \times \left(1 + \frac{\text{CRRT_GAIN_TRIM}}{4096}\right)}$$

CRRT_ADC1 为 ADC1 的电流采样值, 补码表示, 由两个寄存器组成, 地址分别为 CRRT1 (0xA5, crrt_adc1[17:2]), CRRT0 (0xA6, crrt_adc1[1:0])。CRRT_ADC1 = CRRT1 \ll 2 + CRRT0

CRRT_OFST 为 ADC1 的电流采样的内部 offset 校准参数, 补码表示, 地址为 CRRT_TRIM[15:8] (0x4B, crrt_ofst[7:0])。

CRRT_GAIN_TRIM 为 ADC1 的电流采样的内部 gain 校准参数, 补码表示, 地址为 CRRT_TRIM[7:0] (0x4B, crrt_gain_trim[7:0])。

K 为不同的 LSB 系数, 可根据配置参照下表获得。

ADC1 精度选择	PGA 使能 (默认)	PGA 禁止
18bits	2.48 μ V/LSB	1.98 μ V/LSB
16bits	9.68 μ V/LSB	7.93 μ V/LSB
14bits	39.67 μ V/LSB	31.74 μ V/LSB

例如, 在 ADC1 精度选择为 18bits, 以及 PGA 禁止的条件下, 若 CRRT_OFST 和 CRRT_GAIN_TRIM 均 为 0 时, 系数为 1.98 μ V/LSB。

11.3.2 ADC2 电流采集

ADC2 专用于测量电流使用, 它的参考电压是 VREF (262mV), ADC2 在自动工作模式下不开启的。ADC2 只能通过设置寄存器 [ADC2EN\[0\]](#) (0xD0, `sw_adc2_en`) 使能。

ADC2 可以设置为 18bit 的精度, 或者 20bit 的精度, 设置寄存器 [ADC2EN\[1\]](#) (0xD2, `adc2_1sb`) 来设置 采样精度。

0 表示选择 18bit, $LSB=2 \mu V$, 转换时间为 62.5ms;

1 表示选择 20 位精度, $LSB=0.5 \mu V$, 转换时间为 125ms。

ADC2 的转换结果保存在寄存器 [ADC2D0](#) (0xD8) 和 [ADC2D1](#) (0xD9) 内, 两个寄存器共组成 20bit 的数。 $LSB=0.5 \mu V$ 。

由于外部信号噪声和 ADC 的零点漂移, 会影响库仑计计量, 芯片可以设置一个“零区”的电流阈值 [ADC2ZERO\[7:0\]](#) (0xD2, `adc2_zero_range[7:0]`), 当充放电电流值小于这个电流阈值时, ADC2 转换结果会被清零。建议 MCU 使用 ADC2 来获取电流值, 获取精度更高的电流采样值。

芯片内部库仑计按照 ADC2 的转换周期进行累加的。库仑计共 32 位, 带符号补码表示。库仑计累加结果存放在寄存器 [CCH16](#) (0xDA) 和 [CCL16](#) (0xDB) 中, 此寄存器也可被 MCU 读写操作。当 32bit 库仑计发生溢出时, 会产生相应的标志位, 标志位在寄存器 [FLAG2\[15, 14\]](#) (0xC3, `ccuf_flag, ccof_flag`)。`ccuf_flag` 表示下溢出, `ccof_flag` 表示上溢出, 均为写 1 清零。若设置寄存器 [IE2\[15, 14\]](#) (0xC4, `ccuf_ie, ccof_ie`) 为 1, 当标志位产生时, Alarm 管脚输出中断信号, 直到标志位被清零 (写 1 清零)。

ADC2 只在工作模式下是开启的, 在休眠模式下自动关闭。

11.4 外部温度采集

AMG8802 内部集成 3 路外部温度通道，设置寄存器 `UTDCFG[7:6]` (0x1C, `ts_cfg[1:0]`) 选择测量通道数

量和组合。可配的选项有：

00: 外部只有 TS0;

01: 外部只有 TS0 和 TS1;

10: 外部只有 TS0 和 TS2;

11: 外部有 TS0, TS1 和 TS2。

AMG8802 的温度测量参考示意图如下：

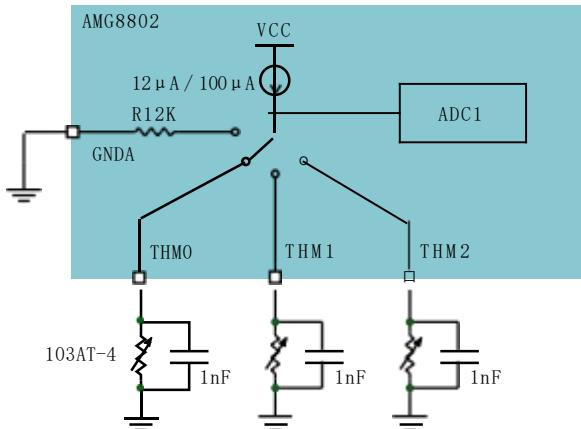


图 6: AMG8802 内部温度测量示意图

图 6 中，测量温度需要在 `TSn` 管脚接 NTC，测量 NTC 电阻时会内部恒流源上拉，电流在各通道形成压降，`ADC1` 对压降进行采样。

芯片内部根据高低温自动切换电流大小，在高温的情况下，芯片输出的测量电流为 $100 \mu A$ ，在低温的情况下，输出的测量电流为 $12 \mu A$ 。

AMG8802 内部集成高精度参考电阻 `R12K`。计算温度时，`R12K` 的电压采样值需要参与计算。

获取温度值的计算方法如下：

- 读取 `TSn` 的转换值，即读取寄存器 `TS0~TS2` (0xA2~0xA4)；
- 读取 `VR12K` 的转换值，即读取寄存器 `VR12K` (0xA7)；
- 算出外部 NTC 此时的电阻值，公式为： $VR = (12K \times TSn) / VR12K$ ；
- 根据算出的 NTC 电阻值 `VR`，通过 103AT-4 的温度公式或查表计算出相应的温度。

11.5 辅助信号采集

ADC1 在测量的辅助信号为：

- VCC 电压
- VDDA 电压
- VBAT 电压
- VD33 电压
- DSGD 电压
- TS1、TS2 作为模拟通道的电压

辅助信号不包含在自动扫描的时序中，需要手动的触发测量辅助信号。方法是直接写寄存器 [ADC1REQ \(0x70\)](#) 设置相关的通道。具体请查询寄存器 0x70 的内容。

11.5.1 VCC 电压采集

指测量 VCC 管脚的电压，使用 [ADC1REQ \(0x70\)](#) 触发测量后，转换结果保存在寄存器 [COM \(0xAF\)](#) 里面，读出寄存器后可以算出 VCC 管脚电压，LSB=3.2mV。

11.5.2 VDDA 电压采集

指的是测量 VDDA 管脚的电压，使用 [ADC1REQ \(0x70\)](#) 触发测量后，转换结果保存在寄存器 [COM \(0xAF\)](#) 里面，读出寄存器后可以算出 VDDA 管脚电压，LSB=0.16mV。

11.5.3 VBAT 电压采集

指的是测量 VBAT 管脚的电压，一般使用它进行电池总压的测量。使用 [ADC1REQ \(0x70\)](#) 触发测量后，转换结果保存在寄存器 [VBAT \(0xA8\)](#) 里面，读出寄存器后可以算出 VBAT 管脚电压，LSB=3.2mV。

需要注意的是，在自动扫描模式下，做一次断线检测也会更新此寄存器，因为断线检测会判断是否每节电芯都要高于 1.5V。

11.5.4 VD33 电压测采集

指的是测量 VBO 管脚的电压，使用 [ADC1REQ \(0x70\)](#) 触发测量后，转换结果保存在寄存器 [COM \(0xAF\)](#) 里面，读出寄存器后可以算出 VBO 管脚电压，LSB=0.16mV。

11.5.5 DSGD 电压采集

DSGD 是负载检测管脚，负载检测一般是在放电 MOS 断开的情况下才进行检测。芯片可以检测此通道的电压值。

使用 [ADC1REQ \(0x70\)](#) 触发测量后，转换结果保存在寄存器 [COM \(0xAF\)](#) 里面，读出寄存器后可以算出 DSGD 管脚电压，LSB=3.2mV。

11.5.6 模拟电压采集

TS1 和 TS2 可以作为通用的模拟输入测量电压。

使用 [ADC1REQ \(0x70\)](#) 触发测量后，转换结果保存在寄存器 [TS1 \(0xA3\)](#) 或 [TS2 \(0xA4\)](#) 里面，读出寄存器后可以算出 TS1 或 TS2 管脚电压，LSB=0.08mV。

11.6 预充和预放驱动

PCHG/PDSG 管脚可以配置为预充功能，也可以配置为预放功能。PCHG/PDSG 管脚驱动外部 MOS 管时，会从管脚输出 $5 \mu A$ 的电流，外部 MOS 的 GS 端需要并联一颗电阻，使得 $5 \mu A$ 的电流流过此电阻，形成 MOS 管的驱动电压。

配置寄存器 `CFGLOCK[13:10]` (0x1F, `pchg_range[5:0]`)，可以选择 PCHG/PDSG 管脚具体是预充功能 还是预防功能。全 0 表示支持预放功能，否则为预充功能。

11.6.1 预充功能

配置寄存器 `CFGLOCK[13:10]` (0x1F, `pchg_range[5:0]`) 不为 0 时，PCHG/PDSG 管脚为预充功能，启 动预充电。

`pchg_range[5:0]` 的步长为 $20.48mV$ ，范围为 $962.56mV \sim 2232.32mV$ 。当电芯电压小于这个电压时，预充电管打开。

当预充 MOS 开启时，充电 MOS 被关闭。

11.6.2 预放功能

配置寄存器 `CFGLOCK[13:10]` (0x1F, `pchg_range[5:0]`) 为 0 时，PCHG/PDSG 管脚为预放功能。

11.7 充电器和负载检测

下图是充电器和负载检测的参考原理图：

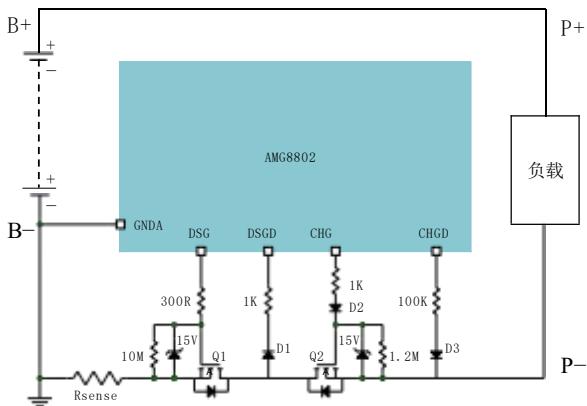


图 7: 充电器/负载检测

图 7 中, 以充放电同口为例。DSG 是放电管驱动管脚, CHG 是充电管驱动管脚, DSGD 是负载检测管脚, CHGD 是充电器检测管脚。

11.7.1 负载检测

通过 DSGD 管脚检测负载, 可参考图 7 的接法。当放电 MOS (Q1) 关断时, 如果使能负载检测, DSGD 内部下拉电阻使能。通过检测 DSGD 电压是否低于 1.6V。假设负载存在, DSGD 被上拉, 表明负载仍然存在。

负载检测在放电管关闭的情况下才会检测。

配置寄存器 `SWCFG[7]` (0xB5, `sw_load_chk`) :

- 1: 表示当放电管关闭后, MCU 需要检测负载在还是不在;
- 0: 无此要求。默认值为 0。

配置寄存器 `OPTION[5]` (0x1E,

`ldchk_md`): 1: 表示需要的时候每隔 1s

检测 64ms;

0: 表示当需要的时候一直检测负载。

判断负载是否存在, 是放电过流 1、放电过流 2、短路保护、欠压 (过放)、放电过温、放电低温保护后释放的条件之一。

- 如果寄存器 `OCDCFG[9]` (0x17, `ocsc_rls`) 为 1, 发生放电过流 1、放电过流 2、短路保护后, 如果检测到负载被移除 50ms, 这些保护状态将被释放;
- 如果寄存器 `OPTION[12]` (0x1E, `uv_rls`) 为 1, 发生欠压 (过放) 保护后, 如果检测到负载被移除 50ms, 同时电芯电压满足释放条件, 欠压 (过放) 保护将被释放;
- 如果寄存器 `OPTION[9]` (0x1E, `otdtd_rls`) 为 1, 发生放电过温或低温保护后, 如果检测到负载被移除 50ms, 同时温度满足释放条件, 放电过温或低温保护将被释放。

DSGD 还可以作为 ADC1 的输入信号, 对负载的电压进行测量, 详细内容请阅读“辅助信号采集”章节。

11.7.2 充电器检测

充电器检测是通过 CHGD 管脚实现，参考图 7。当充电 MOS 关断时，如果使能充电器检测，芯片检测 CHGD 电压是否低于 $-0.6V$ 。当此管脚低于 $-0.6V$ 时，表明充电器存在。

假设充电器存在，充电器的电压一般大于电池组电压，充电器正极与 P+ 连接，负极与 P- 连接，由于充电 MOS 是断开的，且充电器电压大于电池电压，此时就会使得 P- 低于 B-，CHGD 管脚被拉到小于 $-0.6V$ 以下，此时芯片判别充电器是存在的。反之就是不存在。

充电器检测在充电管关闭的情况下，进行检测。

配置寄存器 [SWCFG\[6\]](#) (0xB5, sw_chgr_chk)。

1: 表示充电管关闭后，MCU 使能充电器检测；

0: 无此要求。默认值为 0。

充电器是否存在，可以配置为充电过流保护事件的释放条件。如果寄存器 [OCCCFG\[9\]](#) (0x18, occ_rls) 为 1，发生充电过流保护后，每隔 2s 查看充电器是否已经被移除，若移除 OCC 保护将被释放。

11.8 I2C 通讯接口

采用通用的 I2C 端口，速率可达 100Kbps。芯片 I2C 器件地址为 0x18，即写操作为 0x18，读操作为 0x19。

写寄存器操作格式如下：

起始位	器件地址	寄存器地址	寄存器高位	寄存器低位	CRC8 校验	停止位
Start bit	18H	xxH (1byte)	xxH (1 byte)	xxH (1 byte)	xxH (1 byte)	Stop bit

其中写操作需要计算一个 CRC 值，计算方式如下函数：

说明：从第一个字节（器件地址）开始，到 CRC8 前面一个字节，参与 CRC8 计算：

```
1 uint8_t CRC3(uint8_t* buffer, int off, int len)
2 {
3     uint8_t crc = 0;
4     int i;
5
6     while (len-- > 0)
7     {
8         crc ^= buffer[off++]; // 每次先与需要计算的数据异或，计算完指向下一数据
9         for (i = 8; i > 0; --i) // 下面这段计算过程与计算一个字节 crc 一样
10        {
11            if ((crc & 0x80) != 0)
12                crc = (uint8_t)((crc << 1) ^ 0x07);
13            else
14                crc = (uint8_t)(crc << 1);
15        }
16    }
17
18    return crc;
19 }
```

读寄存器操作格式如下：

起始位	器件写地址	寄存器地址	起始位	器件读地址	寄存器高位	寄存器低位	CRC8 校验	停止位
Start bit	18H	xxH (1byte)	Restart bit	19H	xxH (1 byte)	xxH (1 byte)	xxH (1 byte)	Stop bit

CRC8 校验与写操作算法一致。

11.9 中断输出接口

Alarm 管脚可以配置事件中断输出，除了上电复位的事件外，其他事件均需要使能相应的中断使能，才可以输出相应的信号。

芯片启动时，会产生上电复位事件 (POR)，相应标志位寄存器 [FLAG1\[15\]](#) (0xC0, por_flag) 被置位 1。此位需要被清零后，其他被使能的事件发生后才能发出 Alarm 中断信号。

当某个事件中断被使能且发生时，Alarm 输出低电平，只有清掉中断标志寄存器 [FLAG1](#), [FLAG2](#), [FLAG3](#), [FLAG4](#) (0xC0, 0xC3, 0xC6, 0xC8) 对应的标志位，才能恢复 Alarm 信号电平。标志位写 1 清零。

中断使能寄存器 [IE1](#), [IE2](#), [IE3](#) (0xC1, 0xC4, 0xC7) 可以设置各种事件发生后是否使能输出到 Alarm 管脚。

除了中断外，芯片提供状态寄存器，分别是 [STATUS1](#), [STATUS2](#), [STATUS4](#), [STATUS5](#) (0xC2、0xC5、0xC9、0xCA)，可以通过查询芯片当前的状态。

具体的事件标志位、中断使能、状态位，请参考寄存器列表。

11.10 电子锁功能

Elock 管脚称为电子锁功能管脚，输入管脚，内部有弱下拉电阻。输入低电平，电子锁关闭；输入高电平，电子锁打开。Elock 内部下拉阻值为 $5M\Omega$

Elock 管脚有以下功能：

- Elock 信号输入为低时，放电管驱动 (DSG) 和预放电驱动 (PDSG) 被关闭；
- 如果寄存器 OCDCFG[9] (0x17, ocsc_rls) 为 1，发生放电过流 1、放电过流 2、短路后，电子锁如果关闭了 50ms，重新打开即可释放放电过流和短路事件；
- 如果寄存器 OPTION[12] (0x1E, uv_rls) 为 1，低压 (过放) UV 保护后，除了达到释放阈值外，电子锁如果关闭 50ms，重新打开也可释放低压 (过放) UV 保护状态；
- 如果寄存器 OPTION[9] (0x1E, otdutd_rls) 为 1，放电过温低温保护后，除了温度回到释放温度确认值，电子锁如果关闭了 50ms，重新打开也可释放放电过温低温保护状态。

11.11 EDSG 功能

AMG8802 的 EDSG 管脚是一个数字输入管脚，可以直接控制 DSG 管脚的输出，EDSG 内部下拉阻值为 $5M\Omega$ ，控制逻辑可以参考“放电管驱动”章节。

EDSG 管脚与放电管驱动 DSG 的关系如下，0 和 1 分别代表低电平和高电平。X 为不限。

SWCFG[1:0] (0xB5, edsg_md[1:0])	CFGLOCK[14] (0x1F, edsg_ctrl)	EDSG 管脚	DSG 管脚
'00'	0	0	0
		1	1
	1	1	0
		0	1
'01'	X	0	0
		1	1
'10'	X	1	0
		0	1

EDSG 管脚接受 PWM 的信号输入，通过配置相关寄存器，可以实现在 PWM 输入的情况下，对放电过流或短路产生保护。在使用 PWM 驱动放电管时，延时确认方式是以 PWM 脉冲周期数来确定。PWM 模式保护功能描述如下：

- 配置寄存器 **SWCFG[1:0] (0xB5, edsg_md[1:0])** 为 '11'，表示 EDSG 管脚输入的信号为 PWM 波。此时 EDSG 信号和 DSG 的对应关系为高导通，低关断。
- 设置寄存器 **PWMC[15:11] (0xB6, scd_pwm_dt[4:0])**。在使用 PWM 驱动放电管时，延时确认方式是以 PWM 脉冲周期数来确定，如果连续在此寄存器定义的周期数 (1~32) 内检测到短路，就会确认短路保护，立即关断放电管。释放的方式同正常的释放方式一样。正常的短路保护不受此影响。如果 PWM 信号出现连续高电平或者低电平 8ms，自动终止此次计数，计数器清零等待下一次。默认值为 2。
- 设置寄存器 **PWMC[9:0] (0xB6, ocd2_pwm_dt[9:0])**：在使用 PWM 驱动放电管时，延时确认方式是以 PWM 脉冲周期数来确定，如果连续在此寄存器定义的周期数 (1~1024) 内检测到放电过流 2，就会确认放电过流 2 保护，立即关断放电管。释放的方式同正常的释放方式一样。正常的短路保护不受此影响。如果 PWM 信号出现连续高电平或者低电平 8ms，自动终止此次计数，计数器清零等待下一次。默认值为 15。

如果产生了 PWM 模式下的放电过流 2 保护，**FLAG1[6] (0xC0, ocd2_pwm_flag)** 标志位置位，如果 **IE1[6] (0xC1, ocd2_pwm_ie)** 为 1，发生此保护时，Alarm 管脚输出一个中断信号。可以通过查询 **STATUS1[6] (0xC2, ocd2_pwm_status)** 来确定系统目前是否处于这种保护模式下。

如果产生了 PWM 模式下的短路保护，**FLAG1[7] (0xC0, scd_pwm_flag)** 标志位置位，如果 **IE1[7] (0xC1, scd_pwm_ie)** 为 1，发生此保护时，Alarm 管脚输出一个中断信号。可以通过查询 **STATUS1[7] (0xC2, scd_pwm_status)** 确定系统目前是否处于这种保护模式下。

11.12 保护功能

11.12.1 电压保护

芯片内部集成了与电芯电压相关的保护，包括过压（过充）和欠压（过放）保护

图 8 是过欠压保护的产生和恢复的示意图：

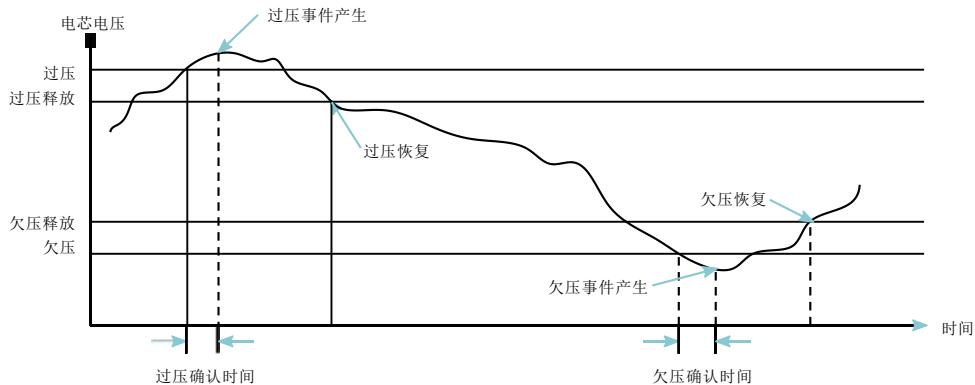


图 8: 过欠压保护示意图

过压 OV (过充) 保护

芯片内部有 3 个过压保护的参数，分别是过压保护值、过压恢复差值、过压检测时间。

过压 OV (过充) 保护产生条件：

- 任一电芯电压 \geq 过压保护值
- 此电压值维持时间 \geq 过压检测时间

过压 OV (过充) 保护释放条件：

- 过压保护值 - 任一电芯电压 \geq 过压恢复差值

设置寄存器 `OVCFG[7:0]` (0x15, `ov_range[7:0]`) 来对过压保护值进行设置，范围为 3276.8mV~4582.4mV，步长为 5.12mV。

设置寄存器 `OVCFG[13:8]` (0x15, `ov_rls_hys`) 来对过压恢复差值进行设置。全 0 表示无 OV 检测和保护功能。范围为 10.24mV~645.12mV，步长为 10.24mV。

设置寄存器 `OVCFG[15:14]` (0x15, `ov_dt[1:0]`) 来设置过压延时确认时。按采样次数计算，00: 2 次，01: 4 次，10: 8 次，11: 12 次。例如设置为 4 次，采样周期为 250ms，过压延时确认时间为 $250\text{ms} \times 4 = 1\text{s}$ 。

过压状态发生后，寄存器 `FLAG1[0]` (0xC0, `ov_flag`) 置 1 (写 1 清 0)；通过查询寄存器 `STATUS1[0]` (0xC2, `ov_status`)，获取当前是否还处在过压状态。

当设置寄存器 `IE1[0]` (0xC1, `ov_ie`) 为 1 时，产生过压保护后，`Alarm` 管脚输出一个中断电平信号（低有效），直到标志位被清零（写 1 清零）。

发生过压（过充）事件后，芯片会进行过压保护，例如关闭充电管，具体保护逻辑请参考过压保护相关章节。

欠压 UV (过放) 保护

芯片内部有 3 个欠压保护的参数，分别是欠压保护值、欠压恢复差值、欠压检测时间。

欠压 UV (过放) 保护产生条件：

- 任一电芯电压 \leq 欠压保护值
- 此电压值维持时间 \geq 欠压检测时间

欠压 UV (过放) 保护释放条件：

- 任一电芯电压 - 欠压保护值 \geq 欠压恢复差值
- 满足上个条件后，如果 [OPTION\[12\] \(0x1F, uv_rls\)](#) 为 0，则欠压状态恢复；若 [uv_rls](#) 位为 1，当检测到负载移除 64ms 或者 Elock 关闭时间超过 50ms 时，欠压状态恢复。

设置寄存器 [UVCFG\[7:0\] \(0x16, uv_range\[7:0\]\)](#) 来对欠压保护值进行设置，范围为 1024mV~3635.2mV，步长为 10.24mV。电芯电压按 14 位计算时 LSB=0.64mV。

设置寄存器 [UVCFG\[13:8\] \(0x16, uv_rls_hys\[5:0\]\)](#) 来对欠压恢复差值进行设置，全 0 表示无 UV 检测和保护功能。范围为 20.48mV~1290.24mV，步长为 20.48mV。

设置寄存器 [UVCFG\[15:14\] \(0x16, uv_dt\[1:0\]\)](#) 来设置欠压延时确认时间，按检测次数计算，00：2 次，01：4 次，10：8 次，11：12 次。这里的次数指的是采样次数，例如设置为 4 次，采样周期为 250ms，欠压延时确认时间就是 $250\text{ms} \times 4 = 1\text{s}$ 。

欠压状态发生后，寄存器 [FLAG1\[1\] \(0xC0, uv_flag\)](#) 会被置 1 (写 1 清 0)；当前是否还处在欠压状态可以查询寄存器 [STATUS1\[1\] \(0xC2, uv_status\)](#)。

如果寄存器 [IE1\[1\] \(0xC1, uv_ie\)](#) 为 1，产生欠压保护时，[Alarm](#) 管脚输出一个中断电平信号 (低有效)，直到标志位被清零 (写 1 清零)。

发生欠压 (过放) 事件后，芯片会进行欠压保护，例如关闭放电管，具体保护逻辑请参考欠压保护相关章节。

11.12.2 电流保护

通过 ADC1 的转换结果，芯片内部进行放电过流保护 1 (OCD1)。放电过流保护 1 有几个需要设置的参数：

- 通过寄存器 [OCDCFG\[8:0\] \(0x17, ocd1_range\[8:0\]\)](#) 设置放电过流保护 1 的阈值。全零表示无 OCD1 检测和保护功能。范围为 0.32mV~163.52mV，步长为 0.32mV。
- 通过寄存器 [OCDCFG\[11:10\] \(0x17, ocd1_dt\[1:0\]\)](#) 设置放电过流 1 的延时确认时间，按检测次数计算。可设置为 2/4/8/12 次。
- 通过寄存器 [OCDCFG\[9\] \(0x17, osc_rls\)](#) 设置放电过流 1 的保护释放条件。0 表示定时释放，定时 32s 后自动释放 OCD1 或者 OCD2 或者 SCD；1 表示负载移除后释放，负载移除后 50ms，或者电子锁关闭 50ms 后释放，否者不释放。

当前放电电流大于过流保护 1 的阈值并持续延时确认时间，放电过流保护 1 产生。当满足第三点释放条件时，放电过流保护 1 被释放。

放电过流保护 1 产生时，在寄存器 [FLAG1\[2\] \(0xC0, ocd1_flag\)](#)。如果设置 [IE1\[2\] \(0xC1, ocd1_ie\)](#) 中断使能，放电过流保护 1 产生时，[Alarm](#) 管脚会输出中断，直到标志位被清零 (写 1 清零)。通过读取寄存器 [STATUS1\[2\] \(0xC2, ocd1_status\)](#) 来获取当前是否处于放电过流 1 保护的状态。

充电过流保护 (OCC)，是通过 ADC1 的转换结果来判断的，充电过流保护有几个需要设置的参数：

- 设置寄存器 `OCCCFG[8:0]` (0x18, `occ_range[8:0]`) 来设置充电过流保护的阈值，全零表示无 OCC 检测和保护功能。范围为 $0.32\text{mV} \sim 163.52\text{mV}$ ，步长为 0.32mV 。
- 设置寄存器 `OCCCFG[11:10]` (0x18, `occ_dt[1:0]`) 来设置充电过流保护的延时确认时间，按检测次数计算。可设置为 2/4/8/12 次。
- 设置寄存器 `OCCCFG[9]` (0x18, `occ_rls`) 来设置充电过流的保护释放条件，0 表示定时释放，定时为 32s，时间到就自动释放 OCC；
1 表示每隔 2s 查看充电器是否已经被移除，若移除 OCC 才被释放，否者不被释放。

当满足当前充电电流大于充电过流保护的阈值并持续延时确认时间，充电过流保护产生。当满足第三点释放条件时，充电过流保护被释放。

充电过流保护产生时，会生产标志位，寄存器 `FLAG1[5]` (0xC0, `occ_flag`) 置位。如果设置寄存器 `IE1[5]` (0xC1, `occ_ie`)，在事件发生时，Alarm 管脚会输出中断，直到标志位被清零 (写 1 清零)。可以通过读取寄存器 `STATUS1[5]` (0xC2, `occ_status`) 来获取当前是否处于充电过流保护的状态。

11.12.3 放电过流 2 和短路保护

芯片内部集成过流比较器，可以用于快速的放电过流保护 2 和短路保护。

放电过流 2 (OCD2) 检测阈值通常大于放电过流 1 (OCD1)，保护延迟时间小于放电过流 1 (OCD1)。放电过流 2 功能不可以被关闭。

OCD2 的参数设置有：

- 放电过流 2 的阈值。通过设置寄存器 `OCDCFG[15:12]` (0x17, `ocd2_th[3:0]`)，设置的范围是 $20\text{mV} \sim 170\text{mV}$ ，步长为 10mV 。
- 放电过流 2 延时确认时间。通过设置寄存器 `OCDCFG[15:12]` (0x18, `ocd2_dt[3:0]`)，可设置的值为 $2\text{ms} \sim 1000\text{ms}$ 。

放电过流 2 的产生条件是，电流信号大于放电过流 2 的阈值，并持续了放电过流 2 的延时确认时间。产生事件后，相应的标志位 `FLAG1[3]` (0xC0, `ocd2_flag`) 置位。如果使能了放电过流 2 的中断输出，即设置了 `IE1[3]` (0xC1, `ocd2_ie`)，Alarm 管脚输出低电平信号，直到标志位被清零 (写 1 清零)。可通过读取寄存器 `STATUS1[3]` (0xC2, `ocd2_status`)，来获取芯片此时是否是放电过流 2 的保护状态。

放电过流 2 的释放条件，可设置寄存器的 `OCDCFG[9]` (0x17, `ocsc_rls`)。0：表示定时释放，定时为 32s，时间到就自动释放；
1：表示要看负载是否被移除 50ms，或者电子锁是否已经关闭 50ms (如果有电子锁输入的话)，否者不会被释放。

短路保护是 BMS 设计中一项非常重要的功能，此时要求系统能非常快速的检测到这种短路情况并关断放电 MOS，实现对短路的保护。

短路需要设置的参数有：

- 放电短路延时确认时间，设置寄存器 `OTCCFG[15:12]` (0x1A, `scd_dt[3:0]`)：可设置的区间为 $30\mu\text{s} \sim 1000\mu\text{s}$ ，步长为 $61\mu\text{s}$ 。
- 放电短路阈值，设置寄存器 `UTDCFG[15:14]` (0x1C, `scd_th[1:0]`)，放电短路的阈值可设置为放电过流保护的 2、3、4、5 倍。

短路的产生条件是，电流信号大于短路的阈值，并持续了短路延时确认时间。产生事件后，相应的标志位 `FLAG1[4]` (0xC0, `scd_flag`) 置位。如果使能了短路的中断输出，即设置了 `IE1[4]` (0xC1, `scd_ie`)，Alarm 管脚输出低电平信号，直到短路状态被释放。可通过读取寄存器 `STATUS1[4]` (0xC2, `scd_status`)，来获取 芯片此时是否是短路的保护状态。

短路的释放条件是，如果设置寄存器的 [OCDCFG\[9\]](#) (0x17, [ocsc_rls](#))。 0: 表示定时释放，定时为 32s，时间到就自动释放；

1: 表示要看负载是否被移除 50ms，或者电子锁是否已经关闭 50ms (如果有电子锁输入的话)，否者不会被释放。

发生放电过流 2 或者短路事件后，放电管、预放电管的驱动会关闭，直到状态恢复后才恢复正常驱动。

该电流比较器的结果也会用于在同口输出的时候，检测到充电或者放电电流之后，强制打开另一个 MOSFET，防止长时间使用 MOSFET 的体二极管，造成损坏。

在充电管关闭时，当有放电电流通过，快速打开充电管，用于对充电 MOS 的保护（使得在同口应用中，放电电流不从充电 MOS 的体二极管过）。这个比较阈值可通过设置寄存器 [OPTION\[3:2\]](#) ([indsg_th\[1:0\]](#))，可设值有：

‘00’ : 2.5mV,

‘01’ : 5mV,

‘10’ : 7.5mV,

‘11’ : 10mV。

假设电流采样电阻为 Rsense，对应的比较电流为 [indsg_th\[1:0\]](#) / Rsense。如果要使能这个功能，还需要设置寄存器 [OPTION\[15\]](#) (0x1E, [dsgon_inchg](#))。

此信号也用于 AMG8802 在休眠模式 1 的状态下，通过放电电流唤醒芯片。一旦达到 [indsg_th\[1:0\]](#) 设置的阈值，即会唤醒芯片恢复到正常工作状态。

11.12.4 推荐 AGND 布局建议

注意

芯片内部的过流比较器是基于 SRN 引脚和 AGND 之间的电压差进行电流的判断的，因此，实际布局的时候，需要考虑大功率线的电流分布和 SRP 以及 AGND 之间的压差是否在合理范围内。

图 9 中所示，A 点位于采样电阻正下方的位置，并且和 SRP 之间基本接近。A 点为推荐的最优化的 AGND 引出位置。B 点和 C 点均远离了大功率电流集中的区域，若从该位置引出连线到 AGND 引脚，则会造成 OCD2/SCD 的保护参数和真实电流之间存在 mV 级别的偏差。

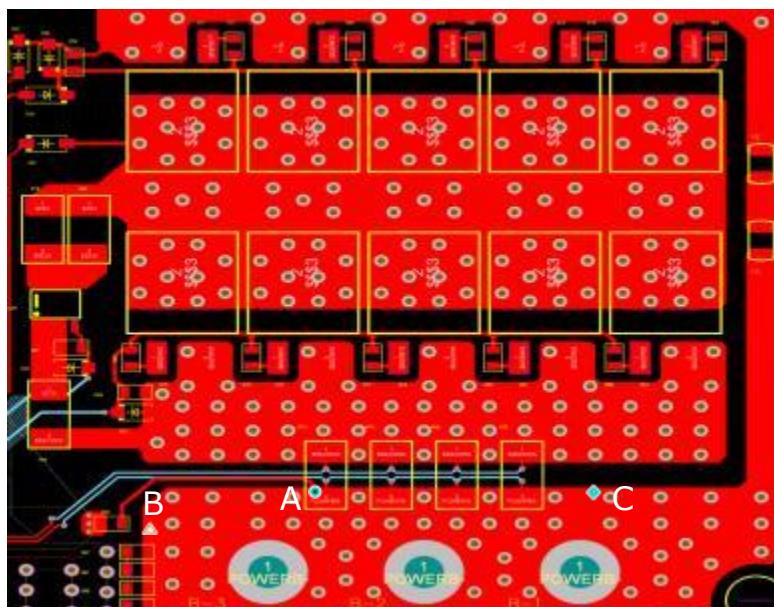


图 9: AGND 布局建议

11.12.5 温度保护

温度相关的保护包括：放电高温保护、放电低温保护、充电高温保护、充电低温保护。

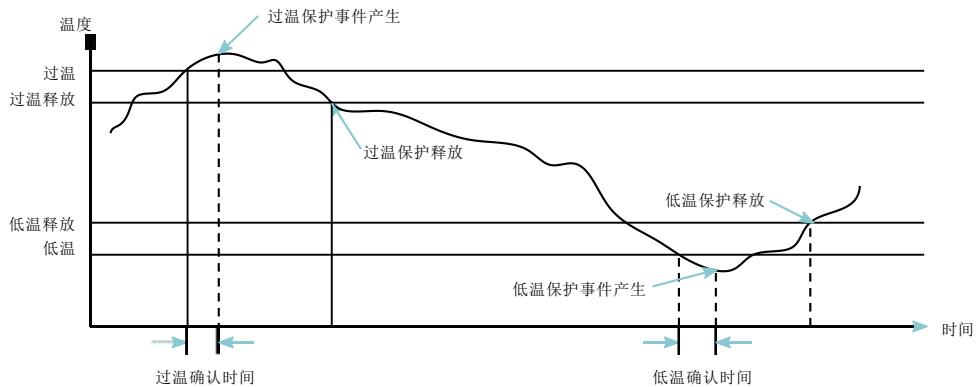


图 10: 温度保护示意图

充电高温保护

充电高温保护需要设置的参数有：

- 充电过温保护起始温度 TP1，设置寄存器 `OTDCFG[13:7]` (0x19, `otc_range[6:0]`)，可设置的范围为 $39^{\circ}\text{C} \sim 67^{\circ}\text{C}$ 。
- 充电过温保护释放滞回温度 TP2，设置寄存器 `OTCCFG[11:6]` (0x1A, `otc_rls_hys[5:0]`)，可设置的范围为 $3^{\circ}\text{C} \sim 10^{\circ}\text{C}$ 。
- 充电过温保护延时确认时间 T，设置寄存器 `OTDCFG[15:14]` (0x19, `ot_dt[1:0]`)。按检测次数计算，00: 2 次，01: 4 次，10: 8 次，11: 12 次。

当 $\text{TP2}=0$ 时，不进行充电过温保护；

当 $\text{TP2}>0$ ，任一温度点高于 TP1，并持续时间 T，即进入充电过温保护；

目前正在充电过温保护时，如果温度降低到 $(\text{TP1}-\text{TP2})$ 时，保护释放。

充电高温保护发生后，会闭充电 MOS，同时标志位寄存器 `FLAG1[9]` (0xC0, `otc_flag`) 置位。如果使能了充电高温保护中断寄存器 `IE1[9]` (0xC1, `otc_ie`) 为 1，发生事件同时，Alarm 管脚会输出中断信号，直到标志位被清零（写 1 清零）。查询当前是否处于充电高温保护状态，可读取寄存器 `STATUS[9]` (0xC2, `otc_status`)。

放电高温保护

放电高温保护需要设置的参数有：

- 放电过温保护起始温度 TP1，设置寄存器 `OTDCFG[6:0]` (0x19, `otd_range[6:0]`)，可设置的范围为 $55^{\circ}\text{C} \sim 85^{\circ}\text{C}$ 。
- 放电过温保护释放滞回温度 TP2，设置寄存器 `OTCCFG[5:0]` (0x1A, `otd_rls_hys[5:0]`)，可设置的范围为 $3^{\circ}\text{C} \sim 10^{\circ}\text{C}$ 。
- 放电过温保护延时确认时间 T，设置寄存器 `OTDCFG[15:14]` (0x19, `ot_dt[1:0]`)，按检测次数计算，00: 2 次，01: 4 次，10: 8 次，11: 12 次。

当 $\text{TP2}=0$ 时，不进行放电过温保护；

当 $\text{TP2}>0$ ，任一温度点高于 TP1，并持续时间 T，即进入放电过温保护；

目前正在放电过温保护时，如果温度降低到 $(\text{TP1}-\text{TP2})$ 时，如果寄存器 `OPTION[9]` (0x1E, `otdutd_rls`) 为 0，则保护释放；如果 `otdutd_rls` 位为 1，需要检测负载已经被移除 50ms 后保护释放。放电高温保护发生后，会关闭放电 MOS，标志位寄存器 `FLAG1[8]` (0xC0, `otd_flag`) 置位。如果使能

了放电高温保护中断，即寄存器 [IE1\[8\]](#) (0xC1, [otd_ie](#)) 为 1，发生事件同时，Alarm 管脚会输出中断信号，直到标志位被清零（写 1 清零）。查询当前是否处于放电高温保护状态，可读取寄存器 [STATUS\[8\]](#) (0xC2, [otd_status](#))。

充电低温保护

充电低温保护需要设置的参数有：

- 充电低温保护起始温度 TP1，设置寄存器 [UTCCFG\[13:7\]](#) (0x1B, [utc_range\[6:0\]](#))，可设置的范围为 $-27^{\circ}\text{C} \sim 3^{\circ}\text{C}$ 。
- 充电低温保护释放滞回温度 TP2，设置寄存器 [UTDCFG\[13:8\]](#) (0x1C, [utc_rls_hys\[5:0\]](#))，可设置的范围为 $3^{\circ}\text{C} \sim 8^{\circ}\text{C}$ 。
- 充电低温保护延时确认时间 T，设置寄存器 [UTCCFG\[15:14\]](#) (0x1B, [ut_dt\[1:0\]](#))，按检测次数计算，00: 2 次，01: 4 次，10: 8 次，11: 12 次。

当 TP2=0 时，不进行充电低温保护；

当 TP2>0，任一温度点低于 TP1，并持续时间 T，即进入充电低温保护；

当前正在充电低温保护时，如果温度上升到 (TP1+TP2) 时，保护释放。

充电低温保护发生后，会关闭充电 MOS，同时标志位也会被置起，即寄存器 [FLAG1\[10\]](#) (0xC0, [utc_flag](#))。如果使能了充电高温保护中断，即寄存器 [IE1\[10\]](#) (0xC1, [utc_ie](#)) 为 1，发生事件同时，Alarm 管脚会输出中断信号，直到标志位被清零（写 1 清零）。查询当前是否处于充电低温保护状态，可读取寄存器 [STATUS\[10\]](#) (0xC2, [utc_status](#))。

放电低温保护

放电低温保护需要设置的参数有：

- 放电低温保护起始温度 TP1，设置寄存器 [UTCCFG\[6:0\]](#) (0x1B, [utd_range\[6:0\]](#))，可设置的范围为 $-30^{\circ}\text{C} \sim 0^{\circ}\text{C}$ 。
- 放电低温保护释放滞回温度 TP2，设置寄存器 [UTDCFG\[5:0\]](#) (0x1C, [utd_rls_hys\[5:0\]](#))，可设置的范围为 $3^{\circ}\text{C} \sim 8^{\circ}\text{C}$ 。
- 放电低温保护延时确认时间 T，设置寄存器 [UTCCFG\[15:14\]](#) (0x1B, [ut_dt\[1:0\]](#))，按检测次数计算，00: 2 次，01: 4 次，10: 8 次，11: 12 次。

当 TP2=0 时，不进行放电低温保护；

当 TP2>0，任一温度点低于 TP1，并持续时间 T，即进入放电低温保护；

当前正在放电低温保护时，如果温度上升到 (TP1+TP2) 时，如果寄存器 [OPTION\[9\]](#) (0x1E, [otdutd_rls](#)) 为 0，则保护释放；如果 [otdutd_rls](#) 位为 1，需要检测负载已经被移除 50ms 后保护释放。

放电低温保护发生后，会关闭放电 MOS，同时标志位也会被置起，即寄存器 [FLAG1\[11\]](#) (0xC0, [utd_flag](#))。如果使能了放电高温保护中断，即寄存器 [IE1\[11\]](#) (0xC1, [utd_ie](#)) 为 1，发生事件同时，Alarm 管脚会输出中断信号，直到标志位被清零（写 1 清零）。查询当前是否处于放电低温保护状态，可读取寄存器 [STATUS\[11\]](#) (0xC2, [utd_status](#))。

过温温度阈值计算

根据 103AT-4 的温度特性，芯片内部设计两路电流源 $12\text{ }\mu\text{A}$ 和 $100\text{ }\mu\text{A}$ 。当测量高温时，使能 $100\text{ }\mu\text{A}$ ，测量低温时，使能 $12\text{ }\mu\text{A}$ 。

当使用 $12\text{ }\mu\text{A}$ 恒流源测量时，比值表示为

$$P12 = \frac{VNTC_ADC}{VR12K_ADC} \times 256 = \frac{RNTC}{R12K} \times 256$$

当使用 $100 \mu A$ 恒流源测量时, 比值表示为

$$P100 = \frac{VR12K_ADC}{VNTC_ADC} \times 256 = \frac{R12K}{RNTC} \times 256$$

对于放电高温保护, 其温度的设置范围设计值为 $60^\circ C \sim 85^\circ C$, 比如把温度保护点设为 $70^\circ C$, 根据 103AT4 温度阻值对应表可查得: $2.228K \Omega @ 70^\circ C$, 根据上面 $P100$ 的公式可得:

$$P100 = \frac{12K}{2.228K} \times 256 = 1378.815$$

四舍五入取整为 1379。根据 OTP 中 $otd_range[6:0]$ 的定义, 在 $otd_range[6:0]$ 中应该写入的值为: $(1379 - 869) / 10 = 51$ 。

下表为 OTD 温度点所对应的设定阈值:

温度点	NTC 阻值	按 $P100$ 公式计算后取整	设定阈值 $otd_range[6:0]$ (取整 $(P100-869) / 10$)	温度点	NTC 阻值	按 $P100$ 公式计算后取整	设定阈值 $otd_range[6:0]$ (取整 $(P100-869) / 10$)
55	3.536	869	0	71	2.163	1420	55
56	3.425	897	3	72	2.1	1463	59
57	3.318	926	6	73	2.039	1507	64
58	3.215	956	9	74	1.98	1552	68
59	3.116	986	12	75	1.924	1597	73
60	3.02	1017	15	76	1.869	1644	78
61	2.927	1050	18	77	1.816	1692	82
62	2.838	1082	21	78	1.765	1741	87
63	2.751	1117	25	79	1.716	1790	92
64	2.668	1151	28	80	1.668	1842	97
65	2.588	1187	32	81	1.622	1894	103
66	2.511	1223	35	82	1.577	1948	108
67	2.436	1261	39	83	1.533	2004	114
68	2.364	1299	43	84	1.492	2059	119
69	2.295	1339	47	85	1.451	2117	125
70	2.228	1379	51				

在进行放电高温 (OTD) 检测时, 如果内部计算出来的 $P100$ 比值大于 1379, 就认为此时温度高于 $70^\circ C$ 。

当发生 OTD 保护后, 释放的阈值通过保护点的迟滞来设置。以 $70^\circ C$ 作为 OTD 的温度保护点, 若温度低于 $65^\circ C$ 时释放, 迟滞的温度为 $-5^\circ C$ 。在 $65^\circ C$ 时, NTC 的阻值为 $2.588K \Omega$ 。从上表可知 $P100 = 1187$,

迟滞设置为

$$H = (P100_{70^\circ C} - P100_{65^\circ C} - 60) / 10 = 13$$

所以应设置参数 `otd_rls_hys[5:0] = 13`。

在芯片内部通过反算会恢复成释放的阈值为 $1379 - (13 \times 10 + 60) = 1189$ ，相比理论上的释放阈值 1187 有误差，该误差在接受范围内。

充电高温 (OTC) 保护及释放计算如同 OTD，区别在于温度保护的范围不一样 ($40^\circ C \sim 65^\circ C$)，`otc_range[6:0]` 和 `otc_rls_hys[5:0]` 的步长为 6。

低温温度阈值计算

对于放电低温保护，其温度的设置范围设计值为 $-30^\circ C \sim 0^\circ C$ 。比如把温度保护点设为 $-20^\circ C$ ，根据 103AT4 温度阻值对应表可查得：67.77K $\Omega @ -20^\circ C$ ，根据上面 p20 的公式可得：

$$P12 = \frac{67.77K}{12K} \times 256 = 1445.76$$

四舍五入取整为 1446。根据 OTP 中 `utd_range[6:0]` 的定义，`utd_range[6:0]` 应为： $(1446 - 582) / 20 = 43$ (取整)。

下表为 UTD 温度点所对应的设定阈值：

温度点	NTC 阻值	按 P12 公式计算后取整	utd_range[6:0] (取整 (P20-582) / 20)	温度点	NTC 阻值	按 P12 公式计算后取整	utd_range[6:0] (取整 (P20-582) / 20)
-35	144.1	3074	125	-17	58.68	1252	34
-34	136.7	2916	117	-16	55.97	1194	31
-33	129.6	2765	109	-15	53.41	1139	28
-32	123.3	2630	102	-14	50.98	1088	25
-31	117.1	2498	96	-13	48.68	1039	23
-30	111.3	2374	90	-12	46.5	992	21
-29	105.7	2255	84	-11	44.43	948	18
-28	100.5	2144	78	-10	42.47	906	16
-27	95.52	2038	73	-9	40.57	865	14
-26	90.84	1938	68	-8	38.77	827	12
-25	86.43	1844	63	-7	37.06	791	10
-24	82.26	1755	59	-6	35.44	756	9
-23	78.33	1671	54	-5	33.9	723	7
-22	74.61	1592	51	-4	32.44	692	6
-21	71.1	1517	47	-3	31.05	662	4

Continued on next page

温度点	NTC 阻值	按 P12 公式计算后取整	utd_range[6:0] (取整 (P20-582) /20)	温度点	NTC 阻值	按 P12 公式计算后取整	utd_range[6:0] (取整 (P20-582) /20)
-20	67.77	1446	43	-2	29.73	634	3
-19	64.57	1377	40	-1	28.48	608	1
-18	61.54	1313	37	0	27.28	582	0

在进行放电低温 (UTD) 检测时, 如果内部计算出来的 p12 值大于 $43 \times 20 + 582 = 1442$ (比理论值 1446 有误差, 对应的温度误差较小, 可以接受), 就认为此时温度低于了 -20°C 。

当发生 UTD 保护后, 释放的阈值通过保护点的迟滞来设置。比如 -20°C 为 UTD 的温度保护点, 迟滞的温度设为 5°C , 温度高于 -15°C 时释放。

在 -15°C 时, NTC 的阻值为 $53.41\text{K}\Omega$, 从上表可知它的 $\text{P12} = 1139$, 它的迟滞设置为

$$H = (P12_{-20^{\circ}\text{C}} - P12_{-15^{\circ}\text{C}} - 40) / 20 = 13$$

所以应设置参数 $\text{utd_rls_hys}[5:0] = 13$ 。

在芯片内部通过反算会恢复成释放的阈值为 $1446 - (13 \times 20 + 40) = 1146$, 相比理论上的释放阈值 1139 会有一点小误差, 但这个误差是可接受的。所以当比值小于释放阈值 1139 时就释放 UTD。

充电低温 (UTC) 保护及释放如同 UTD, 只是温度保护的范围不一样, $\text{utc_range}[6:0]$ 和 $\text{utc_rls_hys}[5:0]$ 的步长为 12, 而不是 20。

内部过温保护

芯片会检测内部温度, 当内部温度大于 145°C 时, 启动温度保护, 停止所有操作, 关闭所有 MOS 驱动, 关闭 BUCK。并且一直检测温度, 当温度恢复到小于 115°C 时, 恢复芯片所有功能。

11.12.6 断线检测

芯片集成了断线检测功能。断线检测可以在自动运行的模式下，也可以软件触发。

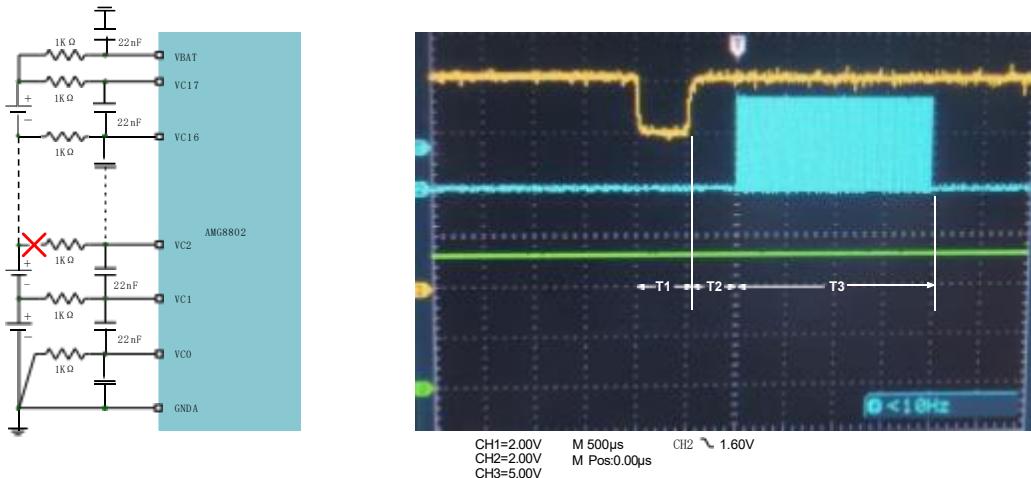


图 11: 断线诊断示意图

假设在图 11 的红色画“×”的地方发生断线。当芯片执行断线检测时，会进行以下动作：

- 使能 VC1 和 VC2 的被动均衡，内部均衡 mos 导通，时间 T1 (见图 11)。
- 关闭 VC1 和 VC2 的被动均衡，延时时间 T2，若断线，VC2-VC1 电压被拉低以后无法恢复；而如果没断线的话，VC2 管脚电压可以快速恢复。
- 经过前 2 步后，开始采样电池电压 (时间段为 T3)。如果电芯电压小于 0.8V 则为断线，如果电芯电压大于 1V 则无断线发生。

触发断线保护后，芯片关闭充电 MOS 驱动，即 CHG 管脚。如果 PCHG/PDSG 管脚被配置为预充功能的话，PCHG/PDSG 管脚驱动也会被关闭。当断线检测到事件恢复之后，CHG 和 PCHG 管脚会重新打开驱动。

芯片允许手动以断线检测的时序触发测量电芯电压，通过设置寄存器 `ADC1REQ[6]` (0x70, `sw_co_sel`) 为 1，然后选择好想测量的通道，触发测量。执行测量命令时，芯片将会以测量断线的方式来进行电压测量，可以参考本节对断线检测原理描述中的波形图。

自动检测

配置寄存器 `OPTION[13]` (0x1E, `co_en`) 为 1，使能自动断线检测；关闭则将此位配置为 0。检测周期为 64s。当断线完成后，会产生相应标志位 `FLAG3[5]` (0xC6, `auto_cochk_done_flag`)，此位置 1 表示有自动断线检测完成，否则表示未完成或没有开始检测。写 1 清零。

软件触发检测

断线检测也可以由软件触发，触发的方式是写 0x1017 到寄存器 `COCHKREQ` (0x72)，即可开始进行断线检测。

断线做完以后会产生相应标志位 `FLAG3[2]` (0xC6, `sw_cochk_done_flag`)，此位置 1 表示断线检测完成，否则表示未完成或没有开始检测。写 1 清零。

检测的结果可以查询寄存器 `FLAG1[13]` (0xC0, `co_flag`)，此位置 1 表示有断线，为 0 则表示正常。在手动触发断线检测时，需要确保平均电压大于 1.5V。否则不会执行断线检测指令。

11.13 被动均衡

芯片集成被动均衡功能，均衡管脚与电压采样管脚复用。被动均衡可以配置相关参数后自主运行，也可以由通讯控制手动设置均衡。当某一节电芯开启被动均衡时，芯片内部有等效的开关导通，形成电流回路。建议外置三极管或 MOS 管，增加均衡电流能力。

当某一节电芯开启被动均衡时，芯片内部有等效的开关导通，形成电流回路。目前芯片内部均衡只支持 6mA 以下的均衡电流。当需要更大的均衡电流时，需要在外置三极管或 MOS 管。

内部均衡和外部均衡的参考示意如图 12：

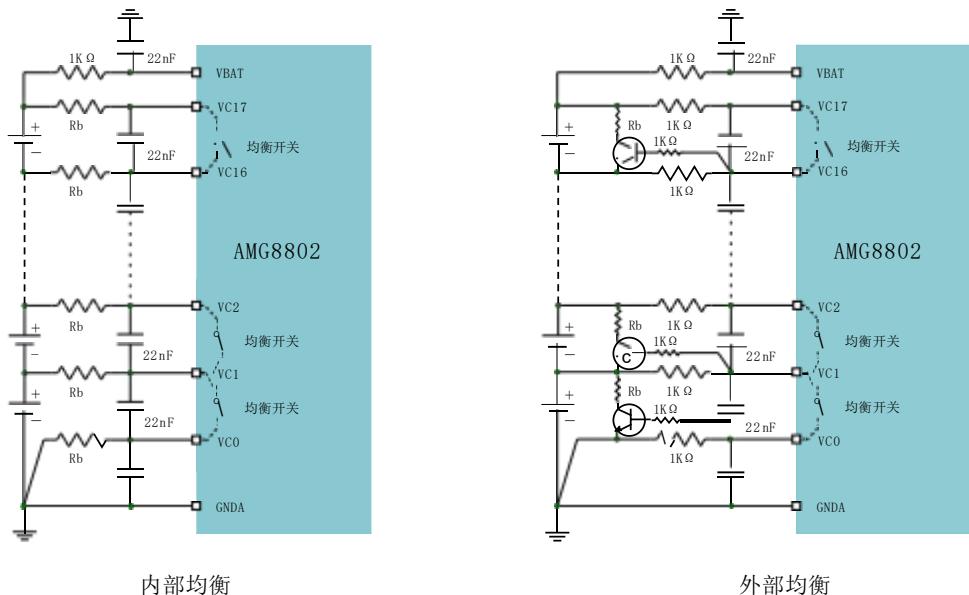


图 12: 电池被动均衡电路示意图

由内部均衡可以看出，某串电芯均衡时，均衡电流会通过两个 R_b 和内部均衡 MOS 的导通阻抗，内部 MOS 导通阻抗约为 300Ω 。均衡电流 $I_b = V_{celln}/(2 \times R_b + R_{INFET})$ 。 R_{INFET} 为内部 MOSFET 导通阻抗。

外部均衡中，某一节电芯均衡时，内部均衡管导通后，驱动外部的三极管（或 MOS 管）导通，均衡电流约为 $I_b = V_{celln}/R_b$ 。

如果有两节相邻的电芯满足了均衡的条件，每一个采样周期完成，均衡的电芯会奇偶进行一次轮换。相邻的电芯不允许同时均衡。

芯片在做电芯电压的采样时，会关闭均衡，采样完成，重新恢复均衡。

11.13.1 自动均衡

自动均衡在配置的电池串数之内进行，例如，芯片被配置为 12 节采样，自动均衡将只判断配置的 12 节电芯的电压。

自动均衡可以被禁止，也可以被使能。下面是具体操作：

禁止被动均衡方式：

· 设置寄存器 CBCFG[6:0] (0x1D, cb_range[6:0]) 为

使能自动被动均衡：

- 设置寄存器 `CBCFG[6:0]` (0x1D, `cb_range[6:0]`) 为均衡起始电压；
- 设置寄存器 `CBCFG[13:12]` (0x1D, `cb_diff[1:0]`) 为均衡电芯压差；

设置参数后，当某一节电芯的电压比最低电压之差大于 `cb_diff[1:0]` 压差，并且此节电芯电压大于均衡起始电压，此节电芯将会被均衡。

为了更清楚的认识自动均衡和采样的时序关系，图 13 是两串相邻电芯在做均衡和采样时的时序切换。在 T1 时间段，第 1 串电池进行均衡；T2 时间段，ADC1 采样，同时均衡处于关闭状态；T3 时间段，第二 串电池进行均衡。

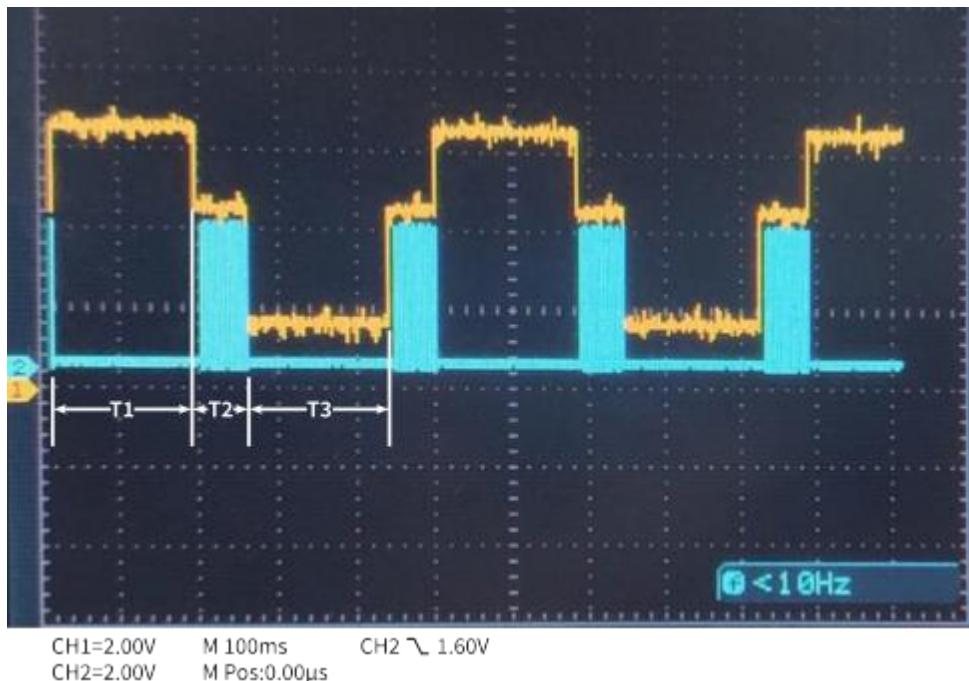


图 13：电池自动均衡时序

11.13.2 手动均衡

手动设置被动均衡的方法如下：

- 设置寄存器 `SWOPTION[0]` (0xB1, `sw_cb_en`) 为 1，由 MCU 来控制均衡。
- 设置寄存器 `SWCB0` 和 `SWCB1` 的 `sw_cb[16:0]`，开启对应电芯的均衡。

需要注意的是，如果相邻两节电芯置位，只有一节能被均衡。芯片会从最低节电芯开始判断，将不符合“相邻电芯不能同时开启均衡”条件的剔除。手动设置均衡，芯片不会自动进行奇偶相邻通道切换。

通过读取寄存器 `CBSEL1` 和 `CBSEL2` 的 `cb_se1[16:0]`，获取当前是电芯均衡状态，读回来的值是实时状态值。需注意，在 ADC1 采样期间，均衡会关闭，如果恰好此时读此寄存器，读回的值是 0，表示均衡被暂时关闭。当采样完成，均衡会被恢复。

芯片内置均衡 30s 超时保护，若外部没有更新 `SWCB` 寄存器，内部看门狗超时，均衡关闭。均衡超时保护后，寄存器 `FLAG2[7]` (0xC3, `swcb_tout_flag`) 置 1，写 1 清零。使能寄存器 `IE2[7]` (0xC4, `swcb_tout_ie`)，`Alarm` 管脚输出中断信号，直到标志位并清零。

通过 MCU 可以诊断均衡通路是否正常。操作如下：

- 手动设置寄存器 `ADC1REQ[12, 6]` (0x70, `sw_cb_adc`, `sw_co_sel`) 置 1, 允许均衡时采样功能。
- 手动对电池进行采样, 芯片将以边做均衡边采样的方式, 更新采样的电池电压值。
- 获取此值以后, 通过和正常情况下电压值的对比, 可以用来当前节电芯的均衡电路是否正常启动。

图 14 是均衡时采样的波形。先开启某一串的电芯的被动均衡, 等待 T_1 时间, 然后开始采样 (T_2 时间段), 采样完成后恢复。

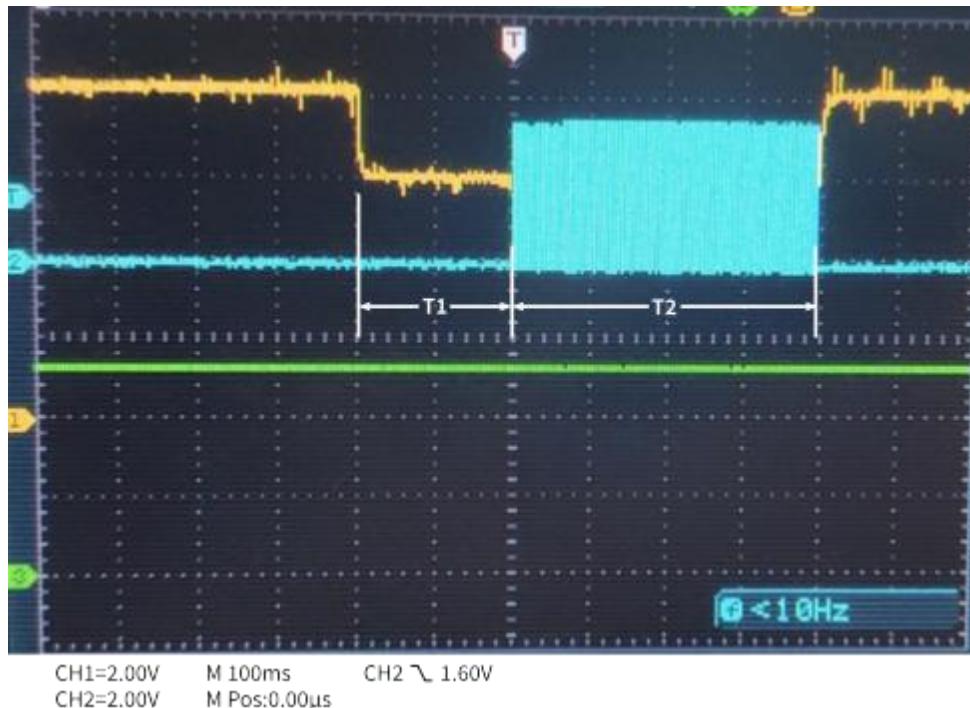


图 14: 电池手动均衡时序

12 推荐应用

12.1 应用电路

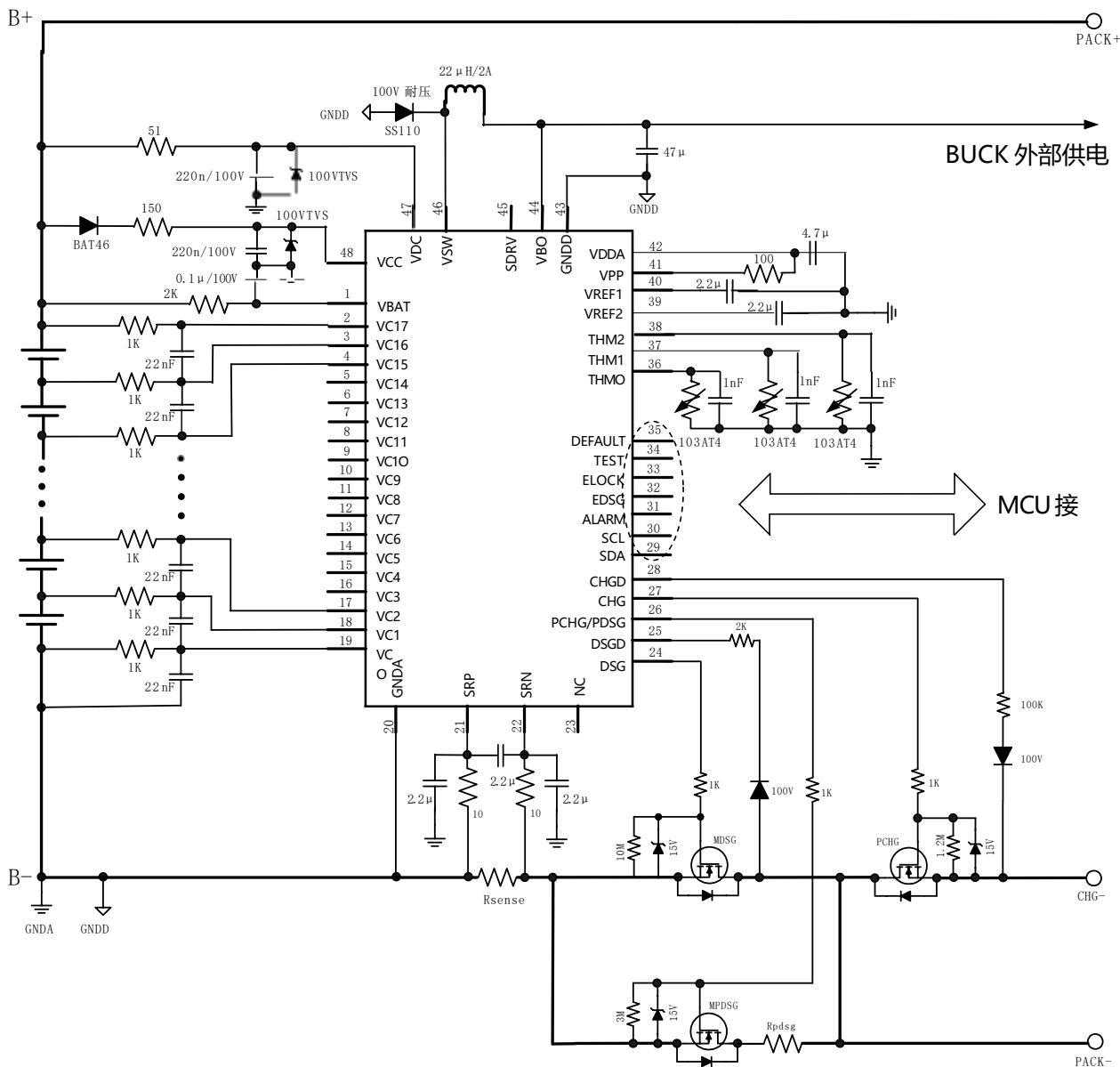


图 15: AMG8802 典型应用电路

12.2 独立工作模式

数字前端芯片（DFE）可以脱离 MCU 独立工作，独立工作时 DFE 自己进行功耗管理，采样和保护等。仅需设置好 OTP 内的寄存器，即可正常工作。

在独立工作模式下，MCU 也可与其通讯，例如做二次保护或实时通讯等。

OTP 用户可以设置的地址范围为 0x15~0x1F，具体见寄存器表的 0x15~0x1F 定义。

OTP 默认是定时从 OTP 映射到 0x55~0x5F 地址。（如果配合 MCU 工作，MCU 可以停止内部定时映射）。

保护功能包括内部过温、过压、欠压、充电过温、充电低温、放电过温、放电低温、二级过流、一级过流、短路、断线，以及对充电 MOS 的保护功能。

（注：下面的寄存器中，按检测次数计算指的是几倍于扫描周期，扫描周期在寄存器 CBCFG (0x1D) 中位 chk_period[1:0] 中配置。）

12.2.1 独立工作模式参数表

过压电压保护

OVCFG[15:14]	0x15	ov_dt[1:0]: 过压延时确认时间，按检测次数计算，00: 2 次，01: 4 次，10: 8 次，11: 12 次。
OVCFG[13:8]	0x15	ov_rls_hys[5:0]: 全零表示无 OV 检测和保护功能。范围为 10.24mV ~645.12mV，步长为 10.24mV。
OVCFG[7:0]	0x15	ov_range[7:0]: 范围为 3276.8mV ~4582.4mV，步长为 5.12mV。电芯电压按 14 位计算时 LSB=0.64mV。

过放电压保护

UVCFG[15:14]	0x16	uv_dt[1:0]: 过放延时确认时间，按检测次数计算，00: 2 次，01: 4 次，10: 8 次，11: 12 次。
UVCFG[13:8]	0x16	uv_rls_hys[5:0]: 全零表示无 UV 检测和保护功能。范围为 20.48mV ~1290.24mV，步长为 20.48mV。
UVCFG[7:0]	0x16	uv_range[7:0]: 范围为 1024mV ~3635.2mV，步长 10.24mV。电压按 14 位计算，LSB=0.64mV。
OPTION[12]	0x1E	uv_rls: 0 表示只要电压回到 UV 释放电压阈值之上，UV 就被释放；1 表示除了达到释放阈值外，还需要看负载是否被移除 64ms，或者电子锁是否已经关闭 64ms（如果有电子锁输入），否者不会被释放。
OPTION[5]	0x1E	1dchk_md: 0 表示当需要的时候一直检测负载，1 表示需要的时候每隔 1s 检测 64ms，都是在放电管关闭下进行。

充电过流保护

OCCCFG[11:10]	0x18	occ_dt[1:0]: 充电过流延时确认时间，按检测次数计算，00: 2 次，01: 4 次，10: 8 次，11: 12 次。
OCCCFG[9]	0x18	occ_rls: 0 表示定时释放，定时为 32s，时间到就自动释放 OCC；1 表示每隔 2s 查看充电器是否已经被移除，若移除 OCC 才被释放，否者不被释放。
OCCCFG[8:0]	0x18	occ_range[8:0]: 充电过流阈值，由电流通道的 ADC1 值来判别。全零表示无 OCC 检测和保护功能。按电流 16bits 计算， $LSB=(0.32mV/8)/4=10\mu V$ 。范围为 0.32mV ~163.52mV，步长为 0.32mV，相当于 16 位电流值的 32 个 LSB。

放电过流保护

OCDCFG[11:10]	0x17	ocd1_dt[1:0]: 放电过流 1 的延时确认时间, 按检测次数计算, 00: 2 次, 01: 4 次, 10: 8 次, 11: 12 次。
OCDCFG[8:0]	0x17	ocd1_range[8:0]: 定义放电过流 1 的阈值, 由电流通道的 ADC1 值来判别。全零表示无 OCD1 检测和保护功能。按电流 16bits 计算, LSB=(0.32mV/8)/4=10 μV。范围为 0.32mV ~163.52mV, 步长为 0.32mV, 相当于 16 位电流值的 32 个 LSB。
OCDCFG[15:12]	0x17	ocd2_th[3:0]: 定义放电过流 2 的阈值, 20mV~170mV, 步长 10mV
OCCCFG[15:12]	0x18	ocd2_dt[3:0]: 放电过流 2 延时确认时间。0000: 2ms, 0001: 5ms, 0010: 10ms, 0011: 20ms, 0100: 30ms, 0101: 40ms, 0110: 60ms, 0111: 80ms, 1000: 100ms, 1001: 150ms, 1010: 200ms, 1011: 300ms, 1100: 400ms, 1101: 600ms, 1110: 800ms, 1111: 1000ms.
OCDCFG[9]	0x17	ocsc_rls: 0 表示定时释放, 定时为 32s, 时间到就自动释放 OCD1 或者 OCD2 或者 SCD; 1 表示要看负载是否被移除 50ms, 或者电子锁是否已经关闭 50ms (如果有电子锁输入), 否者不会被释放。
OTCCFG[15:12]	0x1A	scd_dt[3:0]: 放电短路延时确认时间。大约 30 μs ~1000 μs, 步长为 61 μs。
UTDCFG[15:14]	0x1C	scd_th[1:0]: 放电短路阈值, 分别为放电过流 2 保护阈值的 00: 2 倍, 01: 3 倍, 10: 4 倍, 11: 5 倍。

充电过温保护参数

OTDCFG[15:14]	0x19	ot_dt[1:0]: 过温延时确认时间包括 OTD 和 OTC, 按检测次数计算, 00: 2 次, 01: 4 次, 10: 8 次, 11: 12 次。
OTDCFG[13:7]	0x19	otc_range[6:0]: 定义触发 OTC 保护的范围。可选范围为 39°C ~67°C。
OTCCFG[11:6]	0x1A	otc_rls_hys[5:0]: 定义 OTC 的释放滞回范围。全零表示无 OTC 检测和保护功能。可选范围为 3°C ~10°C。对应数值范围为 18 ~390, 步长为 6。
OTDCFG[6:0]	0x19	otd_range[6:0]: 定义触发 OTD 保护的范围。可选范围为 55°C ~85°C。
OTCCFG[5:0]	0x1A	otd_rls_hys[5:0]: 定义 OTD 的释放滞回范围。全零表示无 OTD 检测和保护功能。可选范围为 3°C ~10°C。对应数值范围为 60 ~680, 步长为 10。
OPTION[9]	0x1E	otdudt_rls: 0 表示温度回到释放温度确认后就释放; 1 表示除了回到释放温度确认之后, 还需要看负载是否被移除 50ms, 或者电子锁是否已经关闭 50ms (如果有电子锁输入), 否者不会被释放。
OPTION[8]	0x1E	ts_dly_sel: 表示对外部温度通道 ADC 测量时的建立时间。如果是选择的高温测量, 0 和 1 分别延时 250us 和 1ms 才做 ADC; 如果是选择的低温测量, 0 和 1 分别延时 1ms 和 4ms. 以支持外部 1nF 和 3.3nF/4.7nF 的电容。

低温保护参数

UTCCFG[15:14]	0x1B	ut_dt[1:0]: 低温延时确认时间, 按检测次数计算, 00: 2 次, 01: 4 次, 10: 8 次, 11: 12 次。
UTCCFG[13:7]	0x1B	utc_range[6:0]: 定义触发 UTC 保护的范围。范围为 -20°C ~5°C, 在 -20 度时, NTC10K 的阻值为 67.77K, 在 5°C 时, 阻值为 22.05K, 分别计算 (67.77*256)/12=1446(取整), (22.05*256)/12=470(取整), 差值为 976, 所以步长取 12 为了更大一点范围, 取值范围为 512 ~2036, 对应温度范围为 3°C ~-27°C。
UTDCFG[13:8]	0x1C	utc_rls_hys[5:0]: 定义 UTC 的释放滞回范围。全零表示无 UTC 检测和保护功能。范围要求为 3°C ~8°C。数值范围为 24 ~768, 步长为 12。
UTDCFG[7:6]	0x1C	ts_cfg[1:0]: 00: 外部只有 TS0, 01: 外部只有 TS0 和 TS1, 10: 外部只有 TS0 和 TS2, 11: 外部有 TS0, TS1 和 TS2。
UTCCFG[6:0]	0x1B	utd_range[6:0]: 定义触发 UTD 保护的范围。-30°C ~0°C, 计算如上, 步长设为 20, 范围为 582 ~3122, 对应温度范围为 0°C ~-35°C。

Continued on next page

UTDCFG[5:0]	0x1C	utd_rls_hys[5:0]: 定义 UTD 的释放滞回范围。全零表示无 UTD 检测和保护功能。范围要求为 3°C ~8°C。数值范围为 40 ~1280, 步长为 20。
-------------	------	---

断线检测保护参数

OPTION[13]	0x1E	co_en: 1 表示支持电芯连接开路检测, 每隔 64 秒检测一次; 0 不支持。
------------	------	--

充电 MOS 保护参数

OPTION[15]	0x1E	dsgon_inchg: 1 表示在充电状态下要强制打开放电管; 0 表示不需要此功能。
OPTION[3:2]	0x1E	indsg_th[1:0]: 用于放电电流比较器的阈值选择。分别为 00: 2.5mV, 01: 5mV, 10: 7.5mV, 11: 10mV。

均衡参数

CBCFG[13:12]	0X1D	cb_diff[1:0]：电芯均衡条件之一，也就是电芯之间电压之差达到这个值。00~11: 对应 10.24mV~40.96mV, 每步 10.24mV, 相当于 16LSB~64LSB (LSB=0.64mV)。
CBCFG[11:8]	0X1D	cell_count[3:0]: 电芯数量, 支持 3~17 串电芯。0000/0001: 3 串; 0010~1111: 对应 4~17 串。
CBCFG[7]	0X1D	cb_ctrl：选择均衡的时机。0: 只在充电状态下做, 1: 在充电状态或者在不充不放状态下做均衡。
CBCFG[6:0]	0X1D	cb_range[6:0]: 电芯均衡条件之一, 电芯电压必须大于此寄存器规定的启动电压。当此值为 0 时, 无电芯均衡功能, 7' d1 ~7' d127 对应范围为 3287.04mV~4577.28mV, 步长为 10.24mV。

预充电 MOS 参数

CFGLOCK[13:8]	0x1F	pchg_range[5:0]: 此值为全零时, 支持预放功能。当非零时, 且当电芯电压低于此值, 启动预充电。范围为 962.56mV~2232.32mV, 步长为 20.48mV = 0.64mV × 32。
---------------	------	--

休眠模式设置

CFGLOCK[7:6]	0x1F	sleep_option[1:0]: 00: 不支持休眠。01: 支持在不充不放状态下, 且没有 OT/UT 和 OHT(过热) 发生以及没有断线, 没有发生 OCC/OCD1/OCD2/SCD 和没有均衡 (也就是, 15 秒后进入休眠状态, 在休眠状态下, 充放电管都被强制关断, 所有的检测被停止, 只打开检测充电器和负载, 等待充电器或者接入负载唤醒, 当然如果有 OV 插入充电器也不会被唤醒的, 如果有 UV 插入负载也是不会被唤醒的。10/11: 支持在不充不放状态下, 且没有 OT/UT 和 OHT(过热) 发生以及没有均衡, 15 秒后进入休眠状态, 但不强制关充放电管, 保持原来的状态, 内部 ADC 检测变为 2s 或 4s 做一次, 等待被充放电电流来唤醒。
CFGLOCK[4:0]	0x1F	idle_range[4:0]: 当电流值小于这个范围就属于不充不放状态。按 14 位电流的 LSB=40 μV, 步长为 2LSB=80 μV, 范围为 (2N+1), 也就是 40 μV~2520 μV。

12.2.2 OTP 烧写

确定好 OTP 参数后，需要烧写到 OTP 里面去，烧写方法为：

- 通过 I2C 口设置寄存器 OTP_MD(0xF0) 为 0x8000，解锁此寄存器的写操作
- 通过 I2C 口设置寄存器 OTP_MD(0xF0) 为 0x8302，使能对 OTP 的编程
- 在 VPP 上加 7.2V 电压，并持续 100ms
- 通过 I2C 口设置 OTP 寄存器 0x15~0x1F，注意每写一个寄存器延时 3ms
- 恢复 VPP 的电压到 VDDA
- 通过 I2C 口设置寄存器 REMAP(0xB0) 为 0x0001，将 OTP 的内容映射到 0x55~0x5F.
- 从 0x55~0x5F 读出数据与写入的数据对比看看是否一致

12.3 配合 MCU 工作模式

配合 MCU 工作时，AMG8802 的保护功能可以同时运行，用户也可以根据需求让 MCU 通过 I2C 接口去修改 AMG8802 保护参数。

12.3.1 MCU 配置自动采样

MCU 可以通过配置 AMG8802 进行自动采样转换，由 AMG8802 根据配置，定期进行数据采集。MCU 只需要根据采样周期，定期进行数据读取即可。

MCU 可以配置 ADC1 对电池电压，电流，以及温度进行自动采样。

相关配置寄存器如下：

CBCFG[15:14]	0x1D	chk_period[1:0]: 检测周期， 00: 125ms, 01: 250ms, 10: 500ms, 11: 1000ms。
OPTION[4]	0x1E	adc1_vltg_lsb: 0 表示 14 位精度（除电流通道以外的其它通道）， 1 表示 16 位精度。用于自动检测。
OPTION[7:6]	0x1E	adc1_crct_lsb[1:0] : 00/01: 14 位精度； 10: 16 位精度； 11: 18 位精度。
OTP_MD[9]	0xF0	auto_chk_stop: 1: 停止内部的自动检测。默认为 0。
UTDCFG[7:6]	0x1C	ts_cfg[1:0]: 温度采集通道选择。 00: 外部只有 TS0, 01: 外部只有 TS0 和 TS1, 10: 外部只有 TS0 和 TS2, 11: 外部有 TS0, TS1 和 TS2。

将 auto_chk_stop 设置为 0，开启自动检测，设置好检测周期和精度，就可以从 0x91~0xA1 读取 17 节电池电压了。

测量结果寄存器

CELL01~17[15:0]	0x91~0xA1	cell101~17_adc1[15: 0] : 电芯 # 1~# 17 电压 ADC 值，带符号补码表达。 LSB = 0.16mV。当做 ADC 时，如果精度选择 14 位，高位对齐，最后两位补 0； 如果选择 16 位，正好放满。
CRRT0[15:0]	0xA5	crct_adc1[17: 2] : 电流通道 ADC 值，带符号补码表达。 当做 ADC 时，如果精度选择 14 位，高位对齐，最后余位补 0； 如果选择 16 位，正好放满，余位 [1: 0] 补 0； 如果选择 18 位精度，最后两位放在 0xA6。 单看这个寄存器 LSB=10 μ V，如果看所有的 18 位，LSB=2.5 μ V。
CRRT1[1:0]	0xA6	crct_data[1: 0]: 如果选择 18 位精度，最后两位放在 0xA6。 单看这个寄存器 LSB=10 μ V，如果看所有的 18 位，LSB=2.5 μ V。
TS0[15:0]	0xA2	ts0_adc1[15: 0] : 外部温度 TS0 通道电压 ADC 值，带符号补码表达。 LSB = 0.08mV。当做 ADC 时，如果精度选择 14 位，高位对齐，最后两位补 0； 如果选择 16 位，正好放满。

Continued on next page

TS1[15:0]	0xA3	ts1_adc1[15: 0] : 外部温度 TS1 通道电压 ADC 值, 表达方式同上。或者 AUX1 的 ADC 值。
TS2[15:0]	0xA4	ts2_adc1[15: 0] : 外部温度 TS1 通道电压 ADC 值, 表达方式同上。或者 AUX2 的 ADC 值。
VR12K[15:0]	0xA7	vr12k_adc1[15: 0]: R12K 通道电压 ADC 值, 带符号补码表达。 LSB = 0.08mV。当做 ADC 时, 如果精度选择 14 位, 高位对齐, 最后两位补 0; 如果选择 16 位, 正好放满。

12.3.2 MCU 手动采样

手动 ADC1 测量

通过对 ADC1REQ 的控制, 可以手动控制 ADC1 进行对电池电压 VCell, 电池包电压 VBAT, 电流以及温度通道等进行测量。另外, 通过 sw_cb_adc 和 sw_co_sel 的配合, 可以进行对均衡功能进行诊断。判断均衡指令是否正确执行。

详细的可选通道可以参考 ADC1REQ[4:0] adc1_req_ch[4:0]。

ADC1REQ[15]	0x70	sw_adc1_req: 此位被写成 1 时, 且 adc1_req_ch[4: 0] 是在有效的通道范围内, 将启动做 ADC。 精度由 sw_adc1_vltg_lsb 或者 sw_adc1_crct_lsb[1: 0] 来选择。 对于电芯来说, 是否要用做断线检测的方式来测量由 sw_co_sel 来决定。 做完 ADC 之后, 此位自动清 0, 同时产生旗标。在此 ADC 没完成前, 此寄存器不能被写。
ADC1REQ[12]	0x70	sw_cb_adc: 配合 sw_co_sel, 当 sw_co_sel=1 时且请求的 cell 通道 ADC, 那么当此位为 1 时将边做均衡边测电压, 以判断均衡开关是否正常。
ADC1REQ[10]	0x70	sw_adc1_vltg_lsb: 用于除电流通道以外所有通道的 ADC 精度选择, 0: 选择 14 位精度, 1: 选择 16 位精度。
ADC1REQ[9:8]	0x70	sw_adc1_crct_lsb[1: 0]: 用于电流通道的 ADC 精度选择, 00/01: 选择 14 位精度, 10: 选择 16 位精度, 11: 选择 18 位精度。
ADC1REQ[7]	0x70	sw_tref_sel: 此配置对请求做外部温度通道有效。 0: 选择参考电流 12 μ A 流过外部温敏 NTC 电阻; 1: 选择参考电流 100 μ A 流过外部温敏 NTC 电阻。
ADC1REQ[6]	0x70	sw_co_sel: 为 1 时, 用做断线检测的方式来测量电芯, 为 0 时, 用正常方式测量电芯。
ADC1REQ[5]	0x70	crct_sync: 1: 表示需要先做电流通道, 然后做 adc1_req_ch[4: 0] 所请求的通道; 0: 无此要求。

Continued on next page

ADC1REQ[4:0]	0x70	adc1_req_ch[4: 0]: 通道号定义如下: 0x00: 无效, 即使 adc1_req=1, 也会认为是无效的命令, 所以不会启动做 ADC。 0x01 ~ 0x11: 用于电芯 # 1 ~ # 17。 0x12 ~ 0x14: 用于 TS0, TS1 和 TS2。 0x15: 用于电流 CRRT 通道。 0x16: 用于内部 R12K 通道。 0x17: 用于 VCC 通道。 0x18: 用于 VBAT 通道。 0x19: 用于 VDDA 通道。 0x1A: 用于 VD33 通道。 0x1C: 选择 DSGD 通道。 0x1D: 选择 TS1 作为 AUX1 模拟通道。ADC 数据和 TS1 共享一个寄存器。 0x1E: 选择 TS2 作为 AUX2 模拟通道。ADC 数据和 TS2 共享一个寄存器。 0x1F: 无效命令, 同 0x00。
--------------	------	---

不管是否停止内部的自动检测 auto_chk_stop, 均可通过设置寄存器 ADC1REQ 来进行单次测量。测量后的转换值结果位于寄存器 0x91~0xA1。

由于电池总压、电源电压等不在自动测量时序内, 因此如果需要测量这些值, 需要 MCU 发命令来进行 ADC 转换, 获取相关的值。

芯片可以测量 Pack+ 与 Pack- 之间的电压, 通过测量 DSGD 管脚测量, 测量时选择对应的通道转换即可。

芯片可以测量 TS1、TS2 作为 AUX ADC 时的外部电压值。

手动 ADC1 的采样值, 电池电压, 电流和温度采样结果均保存于 CELL01~17 寄存器, 以及 CRRT0/CRRT1 寄存器, 和自动控制的转换结果位置相同。电池总压转换结果保存于 VBAT 寄存器。其余信号的转换结果 保存于 COM, 因此其余信号的转换结果需要转换完成后在 COM 寄存器即使读取, 防止被其他信号的转换 结果覆盖。

CELL01~17[15:0]	0x91~0xA1	cell101~17_adc1[15: 0] : 电芯 # 1~# 17 电压 ADC 值, 带符号补码表达。 LSB = 0.16mV。当做 ADC 时, 如果精度选择 14 位, 高位对齐, 最后两位补 0; 如果选择 16 位, 正好放满。
CRRT0[15:0]	0xA5	crrt_adc1[17: 2] : 电流通道 ADC 值, 带符号补码表达。 当做 ADC 时, 如果精度选择 14 位, 高位对齐, 最后余位补 0; 如果选择 16 位, 正好放满, 余位 [1: 0] 补 0; 如果选择 18 位精度, 最后两位放在 0xA6。 单看这个寄存器 LSB=10 μV, 如果看所有的 18 位, LSB=2.5 μV。
CRRT1[1:0]	0xA6	crrt_data[1: 0]: 如果选择 18 位精度, 最后两位放在 0xA6。 单看这个寄存器 LSB=10 μV, 如果看所有的 18 位, LSB=2.5 μV。
TS0[15:0]	0xA2	ts0_adc1[15: 0] : 外部温度 TS0 通道电压 ADC 值, 带符号补码表达。 LSB = 0.08mV。当做 ADC 时, 如果精度选择 14 位, 高位对齐, 最后两位补 0; 如果选择 16 位, 正好放满。
TS1[15:0]	0xA3	ts1_adc1[15: 0] : 外部温度 TS1 通道电压 ADC 值, 表达方式同上。或者 AUX1 的 ADC 值。
TS2[15:0]	0xA4	ts2_adc1[15: 0] : 外部温度 TS2 通道电压 ADC 值, 表达方式同上。或者 AUX2 的 ADC 值。
VR12K[15:0]	0xA7	vr12k_adc1[15: 0]: R12K 通道电压 ADC 值, 带符号补码表达。 LSB = 0.08mV。当做 ADC 时, 如果精度选择 14 位, 高位对齐, 最后两位补 0; 如果选择 16 位, 正好放满。

Continued on next page

VBAT[15:0]	0xA8	vbat_adc1[15: 0]: VBAT 通道电压 ADC 值, 带符号补码表达。 LSB = $0.08\text{mV} \times 40$ 。当做 ADC 时, 如果精度选择 14 位, 高位对齐, 最后两位补 0; 如果选择 16 位, 正好放满。
COM[15:0]	0xAF	common_adc1[15: 0]: VCC/VDDA/VD33/DSDG 通道电压 ADC 值 (共享), 带符号补码表达。 对于 VCC 通道, LSB = $0.08\text{mV} \times 40$; 对于 VDDA 通道, LSB = 0.32mV ; 对于 VD33 通道, LSB = 0.16mV ; 对于 DSDG 通道, LSB = $0.08\text{mV} \times 40$ 。 当做 ADC 时, 如果精度选择 14 位, 高位对齐, 最后两位补 0; 如果选择 16 位, 正好放满。

手动 ADC2 电流测量

FLAG3[7]	0xC6	adc2_done_flag: 当每次 ADC2 完成一次转换, 将此旗标置为 1, MCU 写 1 清 0。
ADC2EN[0]	0xD0	sw_adc2_en: ADC2 是专用于库仑计的 ADC, 在标称 32768Hz 下工作。1: 启用, 0: 停用。
ADC2EN[1]	0xD0	adc2_1sb: 0 选择 18 位精度, LSB = 2\mu V , 转换时间为 62.5ms; 1 选择 20 位精度, LSB = 0.5\mu V , 转换时间为 125ms。
ADC2ZERO[7:0]	0xD2	零区控制范围, 以 20 位的 LSB 为基本单位。默认值为 8。
ADC2D0[15:0]	0xD8	ADC2 结果, 支持 18 位和 20 位, 带符号补码表示, 高位对齐, 余位在下一个寄存器。
ADC2D1[3:0]	0xD9	ADC2 结果低 4 位

ADC2 精度可以达到 20bit。其使用的参考源是 VREF2。

选择启动 ADC2 (sw_adc2_en = 1), 选择转换时间 adc2_1sb (精度), 设置 ADC2 的零区范围 adc2_zero_range[7:0] (即当 ADC 值小于某一值时, ADC 的结果输出强制变成 0)

每个转换周期内部更新一次 ADC2 的值 final_adc2_result。ADC2 的值换算成电压后, 根据采样电阻值换算成电流值。

ADC2 转换完成, FLAG3[7] adc2_done_flag 置位。ADC2 的电流转换结果保存于 ADC2D0/ADC2D1 寄存器。

ADC2D0[15:0]	0xD8	ADC2 结果, 支持 18 位和 20 位, 带符号补码表示, 高位对齐, 余位在下一个寄存器。
ADC2D1[3:0]	0xD9	ADC2 结果低 4 位

12.3.3 MCU 控制均衡

芯片默认自行根据配置进行均衡。但是通过使能 sw_cb_en, 芯片允许 MCU 直接控制均衡。

相关寄存器如下:

SWOPTION[0]	0xB1	sw_cb_en: 当设为 1 时, 电芯均衡功能由 MCU 接管, 默认值为 0, 均衡由内部的配置来决定。
SWOPTION[1]	0xB1	sw_rls: 当设为 1 时, 所有影响充放电管的事件除了满足硬件条件外, 还需要 MCU 来清除旗标来完成释放。 默认值为 0。

Continued on next page

SWCB0[15:0]	0xB2	sw_cb[16: 0]: 在当 sw_cb_en=1 时有效。 用来选择不同的电芯来做均衡，不能同时选择相邻的电芯，此寄存器如果不在 30 秒内刷新，会自动清零。 [16] 对应电芯 # 17， [0] 放在下一个寄存器对应电芯 # 1。
SWCB0[0]	0xB3	sw_cb[16: 0]: 在当 sw_cb_en=1 时有效。 用来选择不同的电芯来做均衡，不能同时选择相邻的电芯，此寄存器如果不在 30 秒内刷新，会自动清零。 [16] 对应电芯 # 17， [0] 放在下一个寄存器对应电芯 # 1。
CBSEL1[15:0]	0xCE	最终送到模拟侧的均衡使能信号，高有效。电芯 # 1 的均衡使能在下一个寄存器。
CBSEL1[0]	0xCF	cb_sel[0] 为电芯 # 1 的均衡使能信号

设置 sw_cb_en=1，然后设置 sw_cb 位，1 位表示哪一节将被均衡。可以通过读取 cb_sel 位看实际均衡的实时状态

需要注意：

1. 芯片内部会控制当采样时均衡会被自动关闭，采样完成后自动重新开启
2. 用户如果设置寄存器 SWCB 中有相邻两节同时均衡的，那么芯片内部会从低节开始判断，把那些不符合“相邻两节电池不能同时均衡”条件的不进行均衡。在手动设置均衡时，内部没有时序去自动奇偶切换。

12.3.4 MCU 事件处理

片内集成了所有事件的标志位和状态位，可通过配置 Alarm 使能，在有事件发生时通过中断（Alarm 引脚）输出给 MCU。

标志位（FLAG）是可以通过写 1 进行清 0 的，状态位（STATUS）是内部自动生成和清除的。

标志和状态参考下面寄存器：

FLAG1[15]	0xC0	por_flag: 当上电复位时，将此旗标置为 1，MCU 写 1 清 0。
FLAG1[13]	0xC0	co_flag: 当电芯断线被检出时，将此旗标置为 1，MCU 写 1 清 0。
FLAG1[12]	0xC0	oht_flag: 当内部过热时（大约 150 度），将此旗标置为 1，MCU 写 1 清 0。
FLAG1[11]	0xC0	utd_flag: 当任一外部温度被检测到放电低温且被确认时，将此旗标置为此旗标为 1。 由旗标 ts0_utd_flag, ts1_utd_flag 和 ts2_utd_flag 组合而成。 MCU 写 1 清 0，也就是清掉所有的 ts0_utd_flag, ts1_utd_flag 和 ts2_utd_flag。
FLAG1[10]	0xC0	utc_flag: 当任一外部温度被检测到充电低温且被确认时，此旗标为 1。 由旗标 ts0_utc_flag, ts1_utc_flag 和 ts2_utc_flag 组合而成。 MCU 写 1 清 0，也就是清掉所有的 ts0_utc_flag, ts1_utc_flag 和 ts2_utc_flag。
FLAG1[9]	0xC0	otc_flag: 当任一外部温度被检测到充电过温且被确认时，此旗标为 1。 由旗标 ts0_otc_flag, ts1_otc_flag 和 ts2_otc_flag 组合而成。 MCU 写 1 清 0，也就是清掉所有的 ts0_otc_flag, ts1_otc_flag 和 ts2_otc_flag。
FLAG1[8]	0xC0	otd_flag: 当任一外部温度被检测到放电过温且被确认时，此旗标为 1。 由旗标 ts0_otd_flag, ts1_otd_flag 和 ts2_otd_flag 组合而成。 MCU 写 1 清 0，也就是清掉所有的 ts0_otd_flag, ts1_otd_flag 和 ts2_otd_flag。

Continued on next page

FLAG1[7]	0xC0	scd_pwm_flag: 当在 PWM 驱动下, 放电短路被确认时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG1[6]	0xC0	ocd2_pwm_flag: 当在 PWM 驱动下, 放电过流被确认时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG1[5]	0xC0	occ_flag: 当充电过流被确认时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG1[4]	0xC0	scd_flag: 当放电短路确认时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG1[3]	0xC0	ocd2_flag: 当放电过流 2 被确认时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG1[2]	0xC0	ocd1_flag: 当放电过流 1 被确认时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG1[1]	0xC0	uv_flag: 当电芯过放被确认时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG1[0]	0xC0	ov_flag: 当电芯过压被确认时, 将此旗标置为 1, MCU 写 1 清 0。
IE1[13]	0xC1	co_ie: 断线诊断触发后, 使能 Alarm 中断输出。清除 co_flag 后, 清除 Alarm。
IE1[12]	0xC1	oht_ie: 内部温度过热触发后, 使能 Alarm 中断输出。清除 oht_flag 后, 清除 Alarm。
IE1[11]	0xC1	utd_ie: 放电低温触发后, 使能 Alarm 中断输出。清除 utd_flag 后, 清除 Alarm。
IE1[10]	0xC1	utc_ie: 充电低温触发后, 使能 Alarm 中断输出。清除 utc_flag 后, 清除 Alarm。
IE1[9]	0xC1	otc_ie: 充电高温触发后, 使能 Alarm 中断输出。清除 otc_flag 后, 清除 Alarm。
IE1[8]	0xC1	otd_ie: 放电高温触发后, 使能 Alarm 中断输出。清除 otd_flag 后, 清除 Alarm。
IE1[7]	0xC1	scd_pwm_ie: PWM 放电短路触发后, 使能 Alarm 中断输出。清除 scd_pwm_flag 后, 清除 Alarm。
IE1[6]	0xC1	ocd2_pwm_ie: PWM 放电过流 2 触发后, 使能 Alarm 中断输出。清除 ocd2_pwm_flag 后, 清除 Alarm。
IE1[5]	0xC1	occ_ie: 充电过流触发后, 使能 Alarm 中断输出。清除 occ_flag 后, 清除 Alarm。
IE1[4]	0xC1	scd_ie: 放电短路触发后, 使能 Alarm 中断输出。清除 scd_flag 后, 清除 Alarm。
IE1[3]	0xC1	ocd2_ie: 放电过流 2 触发后, 使能 Alarm 中断输出。清除 ocd2_flag 后, 清除 Alarm。
IE1[2]	0xC1	ocd1_ie: 放电过流 1 触发后, 使能 Alarm 中断输出。清除 ocd1_flag 后, 清除 Alarm。
IE1[1]	0xC1	uv_ie: 过放触发后, 使能 Alarm 中断输出。清除 uv_flag 后, 清除 Alarm。
IE1[0]	0xC1	ov_ie: 过压触发后, 使能 Alarm 中断输出。清除 ov_flag 后, 清除 Alarm。
STATUS1[13]	0xC2	co_status: 芯片断线诊断状态, 反映当前检测状态。
STATUS1[12]	0xC2	oht_status: 芯片内部温度过热状态, 反映当前检测状态。
STATUS1[11]	0xC2	utd_status: 芯片放电低温状态, 反映当前检测状态。
STATUS1[10]	0xC2	utc_status: 芯片充电低温状态, 反映当前检测状态。
STATUS1[9]	0xC2	otc_status: 芯片充电高温状态, 反映当前检测状态。
STATUS1[8]	0xC2	otd_status: 芯片放电高温状态, 反映当前检测状态。
STATUS1[7]	0xC2	scd_pwm_status: 芯片 PWM 放电短路状态, 反映当前检测状态。
STATUS1[6]	0xC2	ocd2_pwm_status: 芯片 PWM 放电过流 2 状态, 反映当前检测状态。
STATUS1[5]	0xC2	occ_status: 芯片充电过流状态, 反映当前检测状态。
STATUS1[4]	0xC2	scd_status: 芯片放电短路状态, 反映当前检测状态。
STATUS1[3]	0xC2	ocd2_status: 芯片放电过流 2 状态, 反映当前检测状态。

Continued on next page

STATUS1[2]	0xC2	ocd1_status: 芯片放电过流 1 状态, 反映当前检测状态。
STATUS1[1]	0xC2	uv_status: 芯片过放状态, 反映当前检测状态。
STATUS1[0]	0xC2	ov_status: 芯片过压状态, 反映当前检测状态。
FLAG2[15]	0xC3	ccuf_flag: 当库仑计计数器发生下溢时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[14]	0xC3	ccof_flag: 当库仑计计数器发生上溢时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[13]	0xC3	indsg_wkup_flag: 当在休眠模式 1 中检测到放电电流时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[12]	0xC3	ldon_wkup_flag: 在休眠模式 2 下, 当被负载接入唤醒时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[11]	0xC3	chgrin_wkup_flag: 在休眠模式 2 下, 当被充电插入唤醒时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[10]	0xC3	inchg_wkup_flag: 在休眠模式 1 中, 如果充电电流被检出进入充电状态时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[9]	0xC3	sleep_flag: 当进入 sleep 时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[8]	0xC3	sw_wkup_flag: 当唤醒是由软件命令唤醒时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[7]	0xC3	swcb_tout_flag: 当软件均衡超时, 将此旗标置为 1, MCU 写 1 清 0。 超时后, 自动停止均衡, 且均衡寄存器被清零, 直到下一次均衡寄存器被 MCU 刷入新的数据。
FLAG2[6]	0xC3	chgroff_flag: 当充电器移除被检出, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[5]	0xC3	chgrin_flag: 当充电器插入被检出, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[4]	0xC3	ldoff_flag: 当负载移除被检出, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[3]	0xC3	ldon_flag: 当负载插入被检出, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[2]	0xC3	dischg_flag: 当放电状态被检出, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[1]	0xC3	chg_flag: 当充电状态被检出, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[0]	0xC3	idle_flag: 当不充不放状态被检出, 将此旗标置为 1, MCU 写 1 清 0。
IE2[15]	0xC4	ccuf_ie: 库仑计下溢出触发后, 使能 Alarm 中断输出。清除 ccuf_flag 后, 清除 Alarm。
IE2[14]	0xC4	ccof_ie: 库仑计上溢出触发后, 使能 Alarm 中断输出。清除 ccof_flag 后, 清除 Alarm。
IE2[13]	0xC4	indsg_wkup_ie: 休眠放电电流唤醒触发后, 使能 Alarm 中断输出。清除 indsg_wkup_flag 后, 清除 Alarm。
IE2[12]	0xC4	ldon_wkup_ie: 休眠负载接入唤醒触发后, 使能 Alarm 中断输出。清除 ldon_wkup_flag 后, 清除 Alarm。
IE2[11]	0xC4	chgrin_wkup_ie: 休眠充电器接入唤醒触发后, 使能 Alarm 中断输出。清除 chgrin_wkup_flag 后, 清除 Alarm。
IE2[10]	0xC4	inchg_wkup_ie: 休眠充电电流唤醒触发后, 使能 Alarm 中断输出。清除 inchg_wkup_flag 后, 清除 Alarm。
IE2[9]	0xC4	sleep_ie: 休眠触发后, 使能 Alarm 中断输出。清除 sleep_flag 后, 清除 Alarm。
IE2[8]	0xC4	sw_wkup_ie: 软件唤醒触发后, 使能 Alarm 中断输出。清除 sw_wkup_flag 后, 清除 Alarm。
IE2[7]	0xC4	swcb_tout_ie: 均衡超时触发后, 使能 Alarm 中断输出。清除 swcb_tout_flag 后, 清除 Alarm。
IE2[6]	0xC4	chgroff_ie: 充电器移除触发后, 使能 Alarm 中断输出。清除 chgroff_flag 后, 清除 Alarm。

Continued on next page

IE2[5]	0xC4	chgrin_ie: 充电器接入触发后, 使能 Alarm 中断输出。清除 chgrin_flag 后, 清除 Alarm。
IE2[4]	0xC4	ldoff_ie: 负载移除触发后, 使能 Alarm 中断输出。清除 ldoff_flag 后, 清除 Alarm。
IE2[3]	0xC4	ldon_ie: 休眠负载触发后, 使能 Alarm 中断输出。清除 ldon_flag 后, 清除 Alarm。
IE2[2]	0xC4	dischg_ie: 放电状态触发后, 使能 Alarm 中断输出。清除 dischg_flag 后, 清除 Alarm。
IE2[1]	0xC4	chg_ie: 充电状态触发后, 使能 Alarm 中断输出。清除 chg_flag 后, 清除 Alarm。
IE2[0]	0xC4	idle_ie: 静置状态触发后, 使能 Alarm 中断输出。清除 idle_flag 后, 清除 Alarm。
STATUS2[9]	0xC5	sleep_status: 芯片休眠状态, 反映当前检测状态。
STATUS2[6]	0xC5	chgroff_status: 芯片充电器移除状态, 反映当前检测状态。
STATUS2[5]	0xC5	chgrin_status: 芯片充电器接入状态, 反映当前检测状态。
STATUS2[4]	0xC5	ldoff_status: 芯片负载移除状态, 反映当前检测状态。
STATUS2[3]	0xC5	ldon_status: 芯片休眠负载状态, 反映当前检测状态。
STATUS2[2]	0xC5	dischg_status: 芯片放电状态状态, 反映当前检测状态。
STATUS2[1]	0xC5	chg_status: 芯片充电状态状态, 反映当前检测状态。
STATUS2[0]	0xC5	idle_status: 芯片静置状态状态, 反映当前检测状态。
FLAG3[7]	0xC6	adc2_done_flag: 当每次 ADC2 完成一次转换, 将此旗标置为 1, MCU 写 1 清 0。
FLAG3[5]	0xC6	auto_cochk_done_flag: 当每次内部完成自动断线检测时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG3[4]	0xC6	auto_chk_done_flag: 当每次内部完成自动安全事件检测时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG3[2]	0xC6	sw_cochk_done_flag: 当完成一次 MCU 请求的断线检测时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG3[1]	0xC6	sw_chk_done_flag: 当完成一次 MCU 请求的安全事件检测时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG3[0]	0xC6	sw_adc1_done_flag: 当完成一次 MCU 请求的某一个通道或者同步 ADC 转换时, 将此旗标置为 1, MCU 写 1 清 0。
IE3[7]	0xC7	adc2_done_ie: ADC2 采样完成触发后, 使能 Alarm 中断输出。清除 adc2_done_flag 后, 清除 Alarm。
IE3[5]	0xC7	auto_cochk_done_ie: 自动断线检测完成触发后, 使能 Alarm 中断输出。清除 auto_cochk_done_flag 后, 清除 Alarm。
IE3[4]	0xC7	auto_chk_done_ie: 自动保护检测完成触发后, 使能 Alarm 中断输出。清除 auto_chk_done_flag 后, 清除 Alarm。
IE3[2]	0xC7	sw_cochk_done_ie: 手动断线诊断检测完成触发后, 使能 Alarm 中断输出。清除 sw_cochk_done_flag 后, 清除 Alarm。
IE3[1]	0xC7	sw_chk_done_ie: 手动保护检测完成触发后, 使能 Alarm 中断输出。清除 sw_chk_done_flag 后, 清除 Alarm。
IE3[0]	0xC7	sw_adc1_done_ie: 手动 ADC1 采样完成触发后, 使能 Alarm 中断输出。清除 sw_adc1_done_flag 后, 清除 Alarm。
FLAG4[11]	0xC8	ts2_utd_flag: THM2 放电低温保护旗标
FLAG4[10]	0xC8	ts2_utc_flag: THM2 充电低温保护旗标
FLAG4[9]	0xC8	ts2_otc_flag: THM2 充电过温保护旗标
FLAG4[8]	0xC8	ts2_otd_flag: THM2 放电过温保护旗标

Continued on next page

FLAG4[7]	0xC8	ts1_utd_flag: THM1 放电低温保护旗标
FLAG4[6]	0xC8	ts1_utc_flag: THM1 充电低温保护旗标
FLAG4[5]	0xC8	ts1_otc_flag: THM1 充电过温保护旗标
FLAG4[4]	0xC8	ts1_otd_flag: THM1 放电过温保护旗标
FLAG4[3]	0xC8	ts0_utd_flag: THM0 放电低温保护旗标
FLAG4[2]	0xC8	ts0_utc_flag: THM0 充电低温保护旗标
FLAG4[1]	0xC8	ts0_otc_flag: THM0 充电过温保护旗标
FLAG4[0]	0xC8	ts0_otd_flag: THM0 放电过温保护旗标
STATUS4[11]	0xC9	ts2_utd_status: 芯片 TS2 放电低温保护状态, 反映当前检测状态。
STATUS4[10]	0xC9	ts2_utc_status: 芯片 TS2 充电低温保护状态, 反映当前检测状态。
STATUS4[9]	0xC9	ts2_otc_status: 芯片 TS2 充电高温保护状态, 反映当前检测状态。
STATUS4[8]	0xC9	ts2_otd_status: 芯片 TS2 放电高温保护状态, 反映当前检测状态。
STATUS4[7]	0xC9	ts1_utd_status: 芯片 TS1 放电低温保护状态, 反映当前检测状态。
STATUS4[6]	0xC9	ts1_utc_status: 芯片 TS1 充电低温保护状态, 反映当前检测状态。
STATUS4[5]	0xC9	ts1_otc_status: 芯片 TS1 充电高温保护状态, 反映当前检测状态。
STATUS4[4]	0xC9	ts1_otd_status: 芯片 TS1 放电高温保护状态, 反映当前检测状态。
STATUS4[3]	0xC9	ts0_utd_status: 芯片 TS0 放电低温保护状态, 反映当前检测状态。
STATUS4[2]	0xC9	ts0_utc_status: 芯片 TS0 充电低温保护状态, 反映当前检测状态。
STATUS4[1]	0xC9	ts0_otc_status: 芯片 TS0 充电高温保护状态, 反映当前检测状态。
STATUS4[0]	0xC9	ts0_otd_status: 芯片 TS0 放电高温保护状态, 反映当前检测状态。
STATUS5[3]	0xCA	pdsg_en 对应预放电管状态, 1 为导通, 0 为关闭。
STATUS5[2]	0xCA	pchg_en 对应预充电管状态, 1 为导通, 0 为关闭。
STATUS5[1]	0xCA	chg_en 对应充电管状态, 1 为导通, 0 为关闭。
STATUS5[0]	0xCA	dsg_en 对应放电管状态, 1 为导通, 0 为关闭。

12.3.5 MCU 控制 MOSFET

硬件上参与 MOS 控制的引脚:

- ELOCK:** 内部有弱下拉电阻, 当其为低电平 0 时, 放电 MOS 会被关闭, 这个引脚主要的应用场景是当用户用钥匙启动时 (ELOCK 置为高电平), 放电 MOS 才会被允许打开, 当车钥匙拔出时, 放电 MOS 被关闭
- EDSG:** 内部有弱下拉电阻, 可以输入外部信号来控制放电 MOS, 例如输入 PWM 波。其逻辑可以通过寄存器设置。在 sleep 状态下也可配置为 PWM 控制放电管。
- DEFAULT:** 内部有弱下拉电阻, 此引脚的作用是系统刚上电时, 内部状态机还没有工作, 此时放电 MOS 的状态依赖于此引脚。当内部状态机工作起来后, 状态机取得控制权。整个时间大约为 5ms。
- PCHG/PDSG:** 可以配置为预充电, 也可以配置为预放电。

在非 MCU 参与下，即非 OTP 寄存器是默认值的情况下，MOS 只按照 OTP 里面设置的功能进行 MOS 的控制。

在有 MCU 参与下，MCU 可以取得更大的权限控制 MOS。主要的控制方式如下面寄存器描述：

SWOPTION[1]	0xB1	sw_rls: 当设为 1 时，所有影响充放电管的事件除了满足硬件条件外，还需要 MCU 来清除旗标来完成释放。 默认值为 0。
SWCFG[4]	0xB5	pdsg_off_option: (default=0) 0 表示当 UV 和 OHT 发生时才关预放电管， 1 表示任何关 DSG 的事件发生就关预放电管。
SWCFG[1:0]	0xB5	edsg_md[1: 0]: 00: 维持硬件配置 edsg_ctrl 的选择； 01: 输入为低关断放电管； 10: 输入为高关断放电管； 11: 输入为 PWM 波 (高导通, 低关断)。 默认值为 00。
SWFET[2]	0xB8	sw_pfet_ctrl: 1 表示允许打开预充 (或放) 电管, 0 表示 MCU 强制关闭预充 (或放) 电管。默认值为 1。
SWFET[1]	0xB8	sw_cfet_ctrl: 1 表示允许打开充电管, 0 表示 MCU 强制关闭充电管。默认值为 1。
SWFET[0]	0xB8	sw_dfet_ctrl: 1 表示允许打开放电管, 0 表示 MCU 强制关闭放电管。默认值为 1。
PDSGON[4]	0xB9	sw_pdsg_en: (default=0) 1 表示开启预放电功能 (预充自动失效，也就是说预放优先级高于预充)， 0 表示关闭预放电功能。
PDSGON[0]	0xB9	sw_force_pdsg_on: (default=0) 给 MCU 一个超级控制权， 1 表示强制打开预放电管 (此时必须配合 sw_pfet_ctrl=1)， 0 表示由本来设置的条件来决定。
STATUS5[3]	0xCA	pdsg_en 对应预放电管状态， 1 为导通, 0 为关闭。
STATUS5[2]	0xCA	pchg_en 对应预充电管状态， 1 为导通, 0 为关闭。
STATUS5[1]	0xCA	chg_en 对应充电管状态， 1 为导通, 0 为关闭。
STATUS5[0]	0xCA	dsg_en 对应放电管状态， 1 为导通, 0 为关闭。

将 edsg_md[1:0] 设置为 ‘11’，则相关保护使用 0xB6 的参数进行保护。

12.3.6 库仑计

芯片内部集成了一个库仑计，库仑计是按照 ADC2 的转换周期进行累加的。

当用户想知道当前充放多少电，就可以通过芯片内部的库仑计进行计算，而不需要频繁的去读取电流由 MCU 去做累加。减轻了 MCU 的负担。

具体的寄存器如下：

FLAG2[15]	0xC3	ccuf_flag: 当库仑计计数器发生下溢时，将此旗标置为 1，MCU 写 1 清 0。
FLAG2[14]	0xC3	ccof_flag: 当库仑计计数器发生上溢时，将此旗标置为 1，MCU 写 1 清 0。

Continued on next page

IE2[15]	0xC4	ccuf_ie: 库仑计下溢出触发后, 使能 Alarm 中断输出。清除 ccuf_flag 后, 清除 Alarm。
IE2[14]	0xC4	ccof_ie: 库仑计上溢出触发后, 使能 Alarm 中断输出。清除 ccof_flag 后, 清除 Alarm。
ADC2EN[0]	0xD0	sw_adc2_en: ADC2 是专用于库仑计的 ADC, 在标称 32768Hz 下工作。1: 启用, 0: 停用。
ADC2EN[1]	0xD0	adc2_lsb: 0 选择 18 位精度, LSB=2 μ V, 转换时间为 62.5ms; 1 选择 20 位精度, LSB=0.5 μ V, 转换时间为 125ms。
ADC2ZERO[7:0]	0xD2	零区控制范围, 以 20 位的 LSB 为基本单位。默认值为 8。
CCH16[15:0]	0xDA	库仑计共 32 位, 带符号补码表示。 高 16 位在此寄存器, 低 16 位在下一个寄存器。 上溢出和下溢出都会产生旗标。此寄存器可被 MCU 修改计数的起点, 缺省值为 0。 表示的范围和精度将在进一步核算后给出。
CCL16[15:0]	0xDB	库仑计低 16 位。

库仑计使用 ADC2 的转换结果寄存器 ADC2D, 每个转换周期 (例如 20bit 精度时, 转换周期是 125ms) 累加一次。可以用此寄存器通过公式转换为充放电的能量值。

12.3.7 休眠与唤醒

CFGLOCK[7:6]	0x1F	sleep_option[1:0]: 00: 不支持休眠。 01: 支持在不充不放状态下, 且没有 OT/UT 和 OHT(过热) 发生以及没有断线, 没有发生 OCC/OCD1/OCD2/SCD 和没有均衡 (也就是, 15 秒后进入休眠状态, 在休眠状态下, 充放电管都被强制关断, 所有的检测被停止, 只打开检测充电器和负载, 等待充电器或者接入负载唤醒, 当然如果有 OV 插入充电器也不会被唤醒的, 如果有 UV 插入负载也是不会被唤醒的。 10/11: 支持在不充不放状态下, 且没有 OT/UT 和 OHT(过热) 发生以及没有均衡, 15 秒后进入休眠状态, 但不强制关充放电管, 保持原来的状态, 内部 ADC 检测变为 2s 或 4s 做一次, 等待被充放电电流来唤醒。
CFGLOCK[4:0]	0x1F	idle_range[4:0]: 当电流值小于这个范围就属于不充不放状态。按 14 位电流的 LSB=40 μ V, 步长为 2LSB=80 μ V, 范围为 (2N+1), 也就是 40 μ V ~ 2520 μ V。
SWCFG[15]	0xB5	ldon_wkup_en: (default=1), 1 允许负载插入唤醒, 0 不允许。
SWCFG[14]	0xB5	chgrin_wkup_en: (default=1), 1 允许充电器插入唤醒, 0 不允许。
SWCFG[13]	0xB5	indsg_wkup_en: (default=1), 1 允许放电电流唤醒, 0 不允许。
SWCFG[12]	0xB5	inchg_wkup_en: (default=1), 1 允许充电电流唤醒, 0 不允许。
SLPWKUP[15:0]	0xBB	slpwkup_cmd: 如果在 active 模式下, 写 0xXXEE 可让硬件快速进入休眠模式, 而不管硬件的休眠条件是否已经满足; 在休眠模式下, 写 0xXXCC 可以快速唤醒, 包括唤醒由本器件硬件进入的休眠。 其它唤醒源的配置在寄存器 0xB5。
FLAG2[13]	0xC3	indsg_wkup_flag: 当在休眠模式 1 中检测到放电电流时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[12]	0xC3	ldon_wkup_flag: 在休眠模式 2 下, 当被负载接入唤醒时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[11]	0xC3	chgrin_wkup_flag: 在休眠模式 2 下, 当被充电插入唤醒时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[10]	0xC3	inchg_wkup_flag: 在休眠模式 1 中, 如果充电电流被检出进入充电状态时, 将此旗标置为 1, MCU 写 1 清 0。

Continued on next page

FLAG2[9]	0xC3	sleep_flag: 当进入 sleep 时, 将此旗标置为 1, MCU 写 1 清 0。
FLAG2[8]	0xC3	sw_wkup_flag: 当唤醒是由软件命令唤醒时, 将此旗标置为 1, MCU 写 1 清 0。
IE2[13]	0xC4	indsg_wkup_ie: 休眠放电电流唤醒触发后, 使能 Alarm 中断输出。清除 indsg_wkup_flag 后, 清除 Alarm。
IE2[12]	0xC4	ldon_wkup_ie: 休眠负载接入唤醒触发后, 使能 Alarm 中断输出。清除 ldon_wkup_flag 后, 清除 Alarm。
IE2[11]	0xC4	chgrin_wkup_ie: 休眠充电器接入唤醒触发后, 使能 Alarm 中断输出。清除 chgrin_wkup_flag 后, 清除 Alarm。
IE2[10]	0xC4	inchg_wkup_ie: 休眠充电电流唤醒触发后, 使能 Alarm 中断输出。清除 inchg_wkup_flag 后, 清除 Alarm。
IE2[9]	0xC4	sleep_ie: 休眠触发后, 使能 Alarm 中断输出。清除 sleep_flag 后, 清除 Alarm。
IE2[8]	0xC4	sw_wkup_ie: 软件唤醒触发后, 使能 Alarm 中断输出。清除 sw_wkup_flag 后, 清除 Alarm。
STATUS2[9]	0xC5	sleep_status: 芯片休眠状态, 反映当前检测状态。

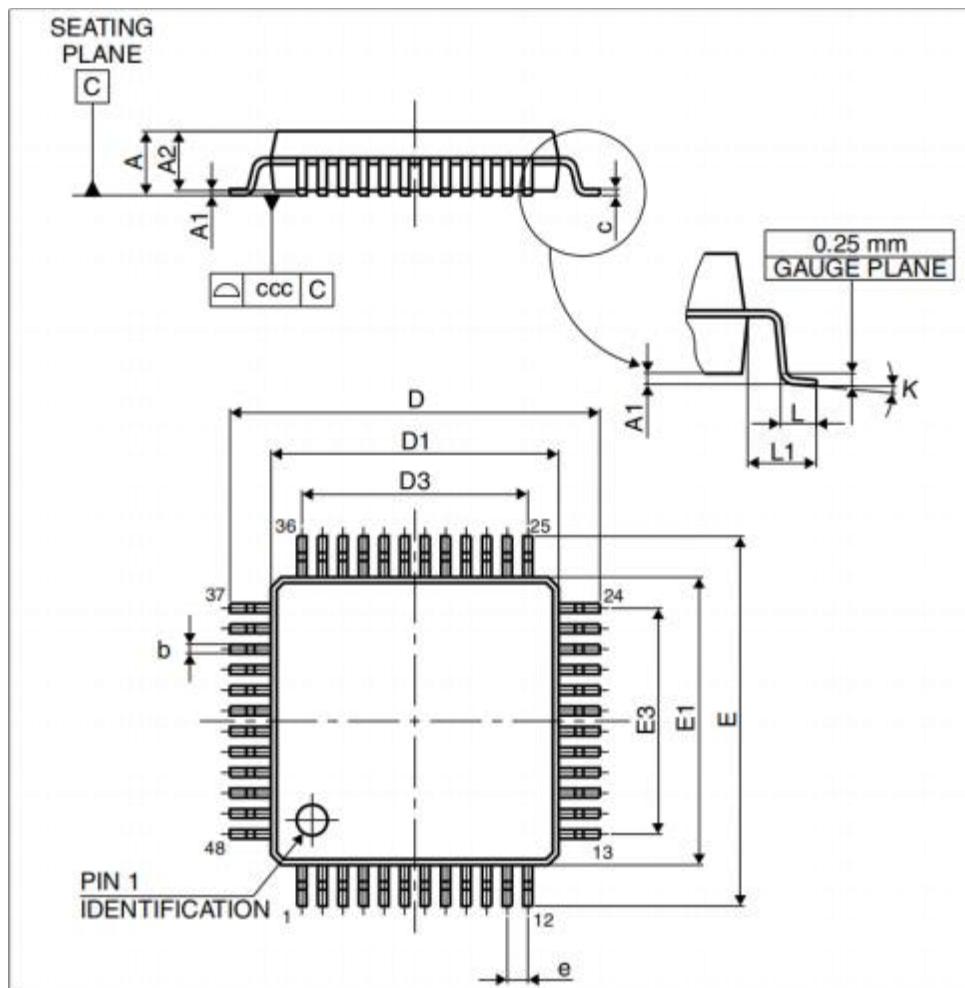
12.3.8 高压 BUCK 控制

高压 buck 可以配置成 3.6V 输出或者 5.4V 输出, 并且可以被 MCU 关闭, 并在一定条件下唤醒。Buck 关闭情况下, 适合 BMS 在运输、存储条件下使用。

具体配置的寄存器如下:

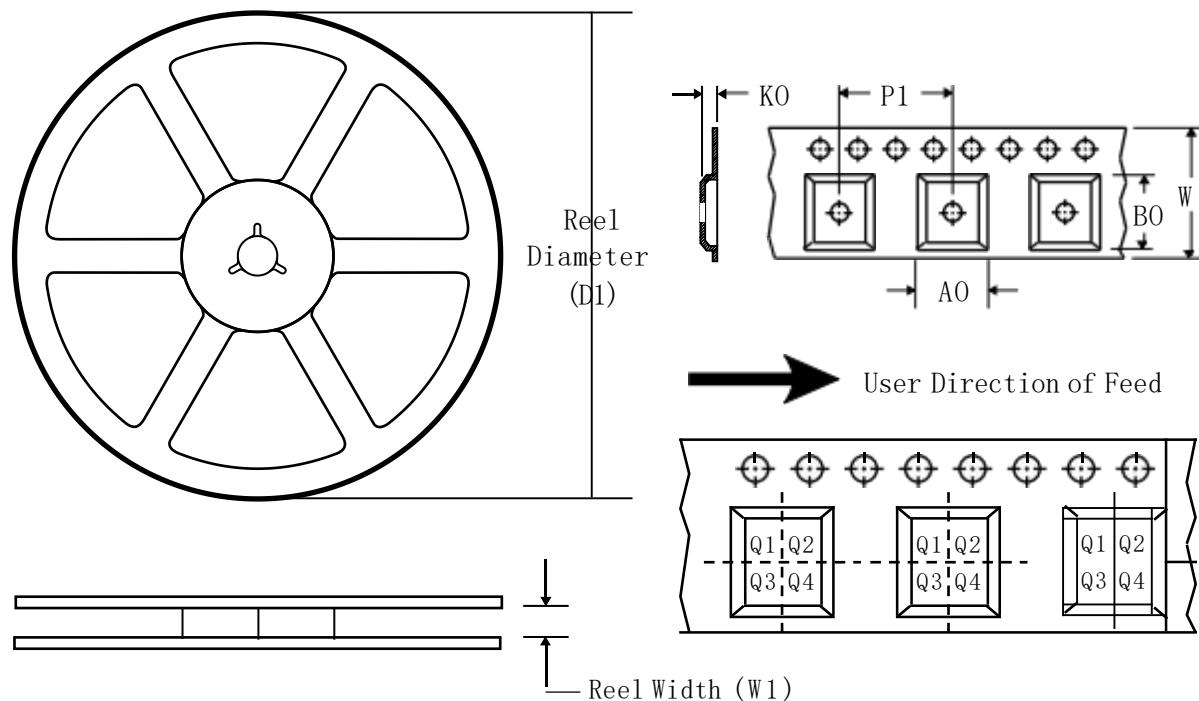
CFGLOCK[5]	0x1F	buck_sel: 0 低电压输出, 典型值 3.6V, 1 高电压输出, 典型值 5.4V。
BUCKDOWN[15:8]	0xBA	buck_down 命令, 写 0xXXA9 到此寄存器将关断 BUCK。 在关断之前, 会先把充放电管以及预充(放)关断, 内部的 ADC 和功能模块也被停止, 等待 16ms 才关掉 BUCK, 然后再等待 64ms 才打开充电器检测。 一旦检测到充电器, 就会打开 BUCK, 然后等待 4ms, 恢复现场。 当 OHT 发生后, 也会强制管 BUCK, 然后等待 OHT 释放, OHT 释放之后 16ms 才打开 BUCK, 大约 4ms 之后恢复现场(基本认为这个时候 BUCK 已完全开启)。

14 封装



符号	毫米 (millimeters)			英寸 (inches)			符号	毫米 (millimeters)			英寸 (inches)		
	最小值	典型值	最大值	最小值	典型值	最大值		最小值	典型值	最大值	最小值	典型值	最大值
A	-	-	1.6	-	-	0.063	E	8.8	9	9.2	0.3465	0.3543	0.3622
A1	0.05	-	0.15	0.002	-	0.0059	E1	6.8	7	7.2	0.2677	0.2756	0.2835
A2	1.35	1.4	1.45	0.0531	0.0551	0.0571	E3	-	5.5	-	-	0.2165	-
b	0.17	0.22	0.27	0.0067	0.0087	0.0106	e	-	0.5	-	-	0.0197	-
c	0.09	-	0.2	0.0035	-	0.0079	L	0.45	0.6	0.75	0.0177	0.0236	0.0295
D	8.8	9	9.2	0.3465	0.3543	0.3622	L1	-	1	-	-	0.0394	-
D1	6.8	7	7.2	0.2677	0.2756	0.2835	k	0°	3.5°	7°	0°	3.5°	7°
D3	-	5.5	-	-	0.2165	-	ccc	-	-	0.08	-	-	0.0031

15 卷带信息



产品名称	封装类型	D1 (mm)	W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin 1 位置
AMG8802-QPAR	QFP-48	330	24.5	9.4	9.4	2.1	16.00	24.00	Q2

16 订货信息

订单料号	封装类型	MSL 等级	工作温度	丝印	Eco	每卷数量
AMG8802-QPAR	QFP-48	3	-40°C-85°C	AMG8802	Green	2000

声明

迈巨微电子提供技术和可靠性数据（包括数据手册）、设计资源（包括参考设计）、应用程序或其他设计建议、WEB 工具、安全信息以及其他资源“按原样”并承担所有错误，并否认所有明示和暗示的保证，包括但不限于任何对适销性、特定用途适用性或不违反第三方的默示保证方知识产权。

迈巨微电子有限公司

Copyright © 2022, Amagitech Inc.