

MT9805 DATASHEET

Version1.6

AEROSEMI

目录

MT9805	1
概述	1
端口配置	1
相比于 MT9811 的区别	3
引脚功能	4
端口描述	6
绝对最大额定值	10
电参数	11
工作	20
工作原理	20
MT9805 内核状态描述	20
隔离 SPI 状态描述	21
功耗	22
ADC 操作	23
ADC 模式	23
7kHz 模式(标准)	23
27kHz 模式(快速)	23
26Hz 模式(滤波)	23
14kHz、3kHz、2kHz、1kHz 和 422Hz 模式	23
ADC 范围和分辨率	24
ADC 范围与电压基准值的关系	24
测量电池电压(ADCV 命令)	25
欠压/过压监视	26
辅助(GPIO)测量(ADAX 命令)	26
辅助(GPIO)测量数字冗余(ADAXD 命令)	26
测量电池电压和 GPIO(ADCVAX 命令)	26
数据采集系统诊断	27

测量内部器件参数(ADSTAT 命令).....	27
测量内部器件参数数字冗余(ADSTATD 命令).....	28
测量电池电压和 GPIO(ADCVSC 命令).....	28
数字冗余诊断.....	29
重叠单元测量(ADOL 命令).....	30
精度检查.....	31
多路复用器(MUX)译码器检查.....	31
数字滤波器检查.....	31
ADC 清除命令.....	32
导线开路检查(ADOW 命令).....	32
热停机.....	34
版本代码.....	34
看门狗和软件放电定时器.....	34
用于电池平衡的 S 引脚脉宽调制.....	36
采用 GPIO 的 I2C/SPI 主控器.....	37
COMM 寄存器.....	37
COMM 命令.....	39
使用 S 端口控制寄存器发送脉冲.....	40
S 引脚静默.....	41
串行接口概述.....	41
四线式串行外设接口(SPI)物理层外部连接.....	41
定时.....	41
两线式隔离型接口(隔离 SPI)物理层.....	41
外部连接.....	42
选择偏置电阻器.....	42
隔离 SPI 脉冲细节.....	43
端口 A 针对 SPI 配置时 MT9805 的操作.....	43
唤醒串行接口.....	46
数据链路层.....	47

包误差检测码	47
广播命令	50
轮询(polling command)方法.....	50
总线协议	52
编程实例	66
唤醒串行接口	66
写配置寄存器	66
读电池寄存器组 A	67
启动电池电压 ADC 转换.....	67
清除电池电压寄存器	67
连接至 MT9805 的 I2C 受控器通信	67
与一个连接至 MT9805 的 SPI 受控器通信.....	69
应用	70
简单的线性稳压器	70
电池和 GPIO 输入的滤波	71
利用内部 MOSFET 进行电池平衡	72
利用外部 MOSFET 进行电池平衡	73
电池测量期间的放电控制	73
采用 MT9805 监视电池数目小于 18 的电池组	75
封装形式	76

MT9805

概述

MT9805 是一款多通道电池堆监控器，可测量多达 18 串电池电压，总测量误差(TME)小于 3.0 mV。MT9805 具有 0 V 至 5 V 的电池测量范围，适合大多数电池应用可在 290 us 内完成所有 18 个电池电压测量，并可以选择较低的数据采集速率以实现高降噪抑制。

可将多个 MT9805 器件串联，以便同时监测更长串的高压电池组。每个 MT9805 都有一个 isoSPI 接口，用于实现高速抗干扰局域通信。多个器件以菊花链形式连接，通过最顶端或底端的器件连接到主处理器。该菊花链可双向操作，即使通信路径出错，也能确保通信完整性。

电池堆可直接用于为 MT9805 供电，也可采用隔离电源对其供电。MT9805 包括针对每个电池的被动均衡，可对每个单元进行单独的脉宽调制(PWM)占空比控制。其他特性包括一个板载 5V 稳压器、9 个通用 I/O 接口以及 SLEEP 模式(在此模式下，功耗降至 7 uA)。

端口配置

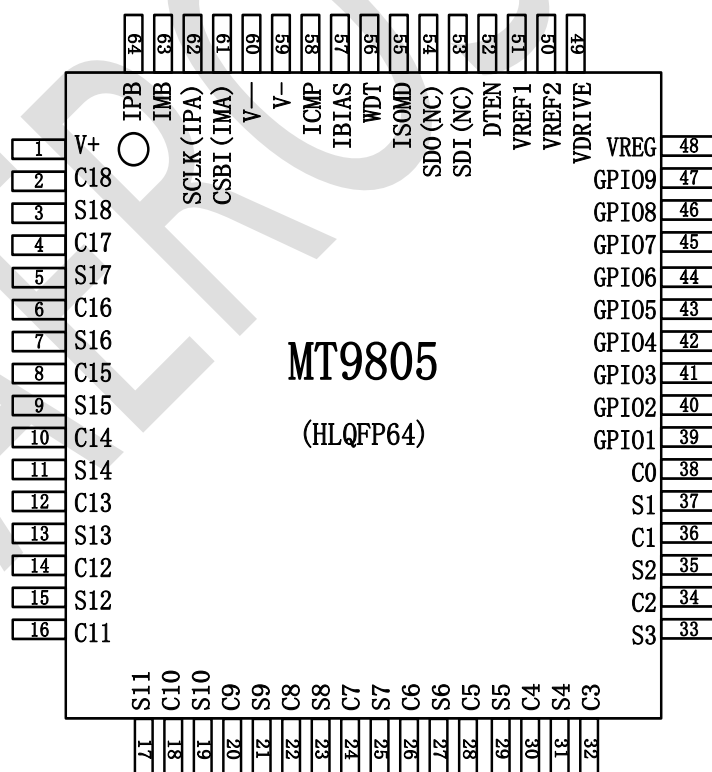


图 1 MT9805 端口配置图

系统框图

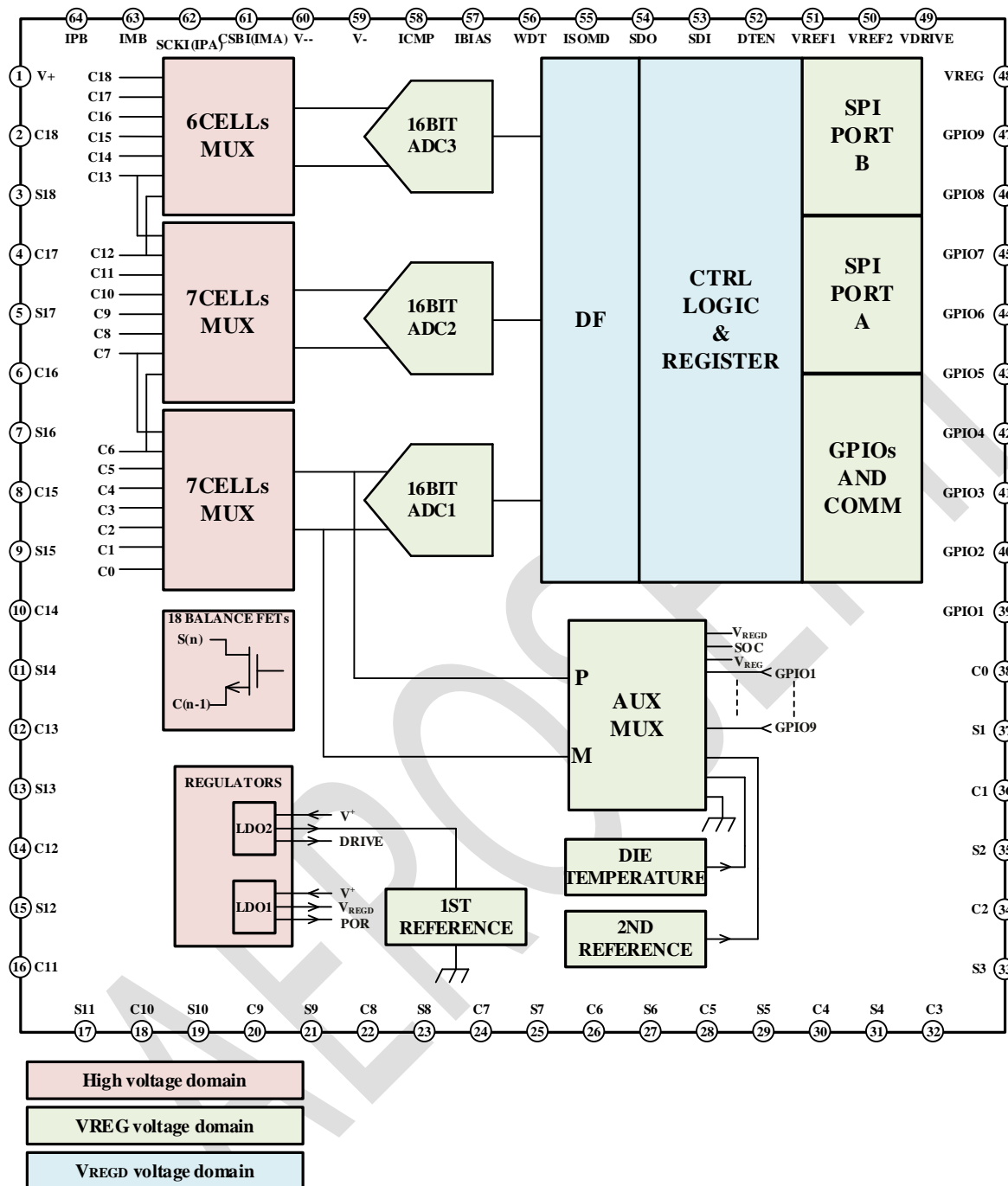


图 2 MT9805 内部结构示意图

相比于 MT9811 的区别

MT9805从功能和性能的角度是MT9811的升级版。表1总结了MT9805相比于MT9811的特性变更和增加情况。

表1 MT9805相比于MT9811的特性变更和增加情况

MT9805新增特性	优势	数据手册相关部分
MT9805有3个同步运行的ADC,而MT9811只有2个ADC。	每个转换周期可以测量3个电池单元。	ADC操作
除了3个ADC数字滤波器之外,还有冗余的第4个滤波器。	检查所有数字滤波器是否无故障。	说明参见具有数字冗余的ADC转换部分以及表25中的PS位[1:0]
利用ADOL命令,使用ADC1和ADC2测量电池7,使用ADC2和ADC3测量电池13。	检查ADC2与ADC1一样准确,并检查ADC3与ADC2一样准确。	重叠单元测量(ADOL命令)
放电计时期间可以使能监视特性。当电池电压达到可编程欠压阈值时,电池平衡可以自动终止。	改善电池平衡。	放电定时器监控器
内部放电MOSFET可以支持100 mA的平衡电流(V _{cell} =4V时)	更快的电池平衡,尤其是电池电压较低时。	使用内部MOSFET的电池平衡
在不影响TME的情况下,C0引脚电压的允许范围为0 V至1V。	C0不必直接连接到V ₋ 。	ADC直流规格
均衡静默MUTE和取消均衡静默UNMUTE命令允许主机关断或开启放电引脚(S引脚),而不会覆盖寄存器值。	更好地控制S引脚之间的时序。关闭和开启电池测量。	S引脚静音
辅助测量具有开路诊断特性	改善故障检测。	辅助开路检查(AXOW命令)
添加了四个GPIO引脚,总共有九个。	可以测量更多的温度或其他传感器。	辅助(GPIO)测量(ADAX命令)和辅助开路检查(AXOW命令)
MT9805的菊花链可以双向工作(两个端口均可为主机或从机)。	冗余通信路径。	可反向isoSPI

引脚功能

MT9805 提供了一个用于底端电池的输入(C0)。这一简单的引脚功能差异为提升第一节电池(cell1)的测量精度、增强 SPI 噪声耐受性及简化布线提供了可能。

C0 至 C18: 电池输入。

S1 至 S18: 平衡输入/输出。在 MT9805 内部 S(n)和 C(n-1)之间连接了 18 个 N-MOSFET, 用于对电池进行放电。

V+: 正电源引脚。

V-: 负电源引脚。V-引脚和 V--引脚必须在 IC 的外部短接在一起。

V--: 负电源引脚。V-引脚和 V--引脚必须在 IC 的外部短接在一起。

VREF2: 经过缓冲的第二基准电压。采用一个外部 1uF 电容器进行旁路。

VREF1: ADC 基准电压。采用一个外部 1uF 电容器进行旁路。

GPIO1~GPIO9: 通用 I/O。可用作数字输入或数字输出, 或者用作具有一个从 V- 至 5V 之测量范围的模拟输入。GPIO[3:5]可用作一个 I²C 或 SPI 端口。

DTEN: 软件定时器启用。把该引脚连接至 VREG 以使能软件定时器。

VDRIVE: 将一个 NPN 的基极连接至该引脚。把该 NPN 的集电极连接至 V+, 发射极连接至 VREG。

VREG: 5V 稳压器输入。采用一个外部 1uF 电容器进行旁路。

ISOMD: 串行接口模式。把 ISOMD 连接至 VREG 即可针对两线式隔离型 SPI 接口模式来配置 MT9805 的引脚 61 至 64。而将 ISOMD 连接至 V- 则可针对四线式 SPI 模式来配置 MT9805。

WDT: 看门狗定时器输出引脚。该引脚是一个漏极开路 NMOS 数字输出。其可被置于不连接状态或通过一个 1M 电阻器连接至 VREG。如果 MT9805 在 2 秒之内未接到一个唤醒信号图 25, 则看门狗定时器电路将使 MT9805 复位并使 WDT 引脚变为高阻态。

CSBI (IMA)、SCLK (IPA)、SDI (NC)、SDO(NC): 四线式串行外设接口(SPI)。低电平有效芯片选择(CSBI)、串行时钟(SCLK)和串行数据输入(SDI)是数字输入。串行数据输出(SDO)是一个漏极开路 NMOS 输出引脚。SDO 需要一个 2k 上拉电阻器。

IPA、IMA: 隔离型两线式串行接口端口 A。IPA(+)和 IMA(-)是一对差分输入/输出。

IPB、IMB: 隔离型两线式串行接口端口 B。IPB(+)和 IMB(-)是一对差分输入/输出。

IBIAS: 隔离型接口电流偏置。通过一个电阻分压器把 IBIAS 连接至 V- 以设定接口输出电流水平。当隔离 SPI 接口被使能时, IBIAS 引脚输出电压为 2V。IPA/IMA 或 IPB/IMB 输出驱动电流被设定为从 IBIAS 引脚供应之电流 (IB)的 20 倍。

ICMP: 隔离型接口比较器电压门限设定。把该引脚连接至介于 IBIAS 和 V^- 之间的电阻分压器以设定隔离 SPI 接收器比较器的电压门限。比较器门限被设定为 ICMP 引脚电压的 1/2。

表 2 串行端口引脚

/	MT9805(可进行菊花链连接)	
	ISOMD=VREG	ISOMD= V^-
端口 B(引脚 57 至 58 引脚 63 至 64)	IPB	IPB
	IMB	IMB
	ICMP	ICMP
	IBIAS	IBIAS
端口 A(引脚 53 至 54 引脚 61 至 62)	(NC)	SDO
	(NC)	SDI
	IPA	SCLK
	IMA	CSBI

端口描述

名称	编号	I/O	描述
V+	1	I	电源正输入端
C18	2	I	第 18 节电池正输入端
S18	3	I/O	第 18 节电池均衡端
C17	4	I	第 17 节电池正输入端
S17	5	I/O	第 17 节电池均衡端
C16	6	I	第 16 节电池正输入端
S16	7	I/O	第 16 节电池均衡端
C15	8	I	第 15 节电池正输入端
S15	9	I/O	第 15 节电池均衡端
C14	10	I	第 14 节电池正输入端
S14	11	I/O	第 14 节电池均衡端
C13	12	I	第 13 节电池正输入端
S13	13	I/O	第 13 节电池均衡端
C12	14	I	第 12 节电池正输入端
S12	15	I/O	第 12 节电池均衡端
C11	16	I	第 11 节电池正输入端
S11	17	I/O	第 11 节电池均衡端
C10	18	I	第 10 节电池正输入端
S10	19	I/O	第 10 节电池均衡端

C9	20	I	第 9 节电池正输入端
S9	21	I/O	第 9 节电池均衡端
C8	22	I	第 8 节电池正输入端
S8	23	I/O	第 8 节电池均衡端
C7	24	I	第 7 节电池正输入端
S7	25	I/O	第 7 节电池均衡端
C6	26	I	第 6 节电池正输入端
S6	27	I/O	第 6 节电池均衡端
C5	28	I	第 5 节电池正输入端
S5	29	I/O	第 5 节电池均衡端
C4	30	I	第 4 节电池正输入端
S4	31	I/O	第 4 节电池均衡端
C3	32	I	第 3 节电池正输入端
S3	33	I/O	第 3 节电池均衡端
C2	34	I	第 2 节电池正输入端
S2	35	I/O	第 2 节电池均衡端
C1	36	I	第 1 节电池正输入端
S1	37	I/O	第 1 节电池均衡端
C0	38	I	第 1 节电池负输入端
GPIO1	39	I/O	通用输入/输出 1
GPIO2	40	I/O	通用输入/输出 2

GPIO3	41	I/O	通用输入/输出 3
GPIO4	42	Ground	通用输入/输出 4
GPIO5	43	Ground	通用输入/输出 5
GPIO6	44	I/O	通用输入/输出 6
GPIO7	45	I/O	通用输入/输出 7
GPIO8	46	O	通用输入/输出 8
GPIO9	47	O	通用输入/输出 9
VREG	48	I	线性稳压器输入端口
VDRIVE	49	I	驱动 NPN 基极的输出端口
VREF2	50	O	经过缓冲的第二基准电压
VREF1	51	O	ADC 基准电压
DTEN	52	I	软件定时器启用输入端口
SDI(NC)	53	I	四线式串行外设接口(SPI)
SDO(NC)	54	O	四线式串行外设接口(SPI)
ISOMD	55	I	串行接口模式
WDT	56	I	看门狗定时器输出引脚
IBIAS	57	O	隔离型接口电流偏置
ICMP	58	I	隔离型接口比较器电压门限设定
V--	59	Ground	模拟电路“地”
V-	60	Ground	模拟电路“地”
CSBI(IMA)	61	I/O	四线式串行外设接口(SPI)，隔离型两线式串行接口端

			口 A 差分输入/输出
SCLK(IPA)	62	I/O	四线式串行外设接口(SPI)，隔离型两线式串行接口端 口 A 差分输入/输出
IMB	63	I/O	隔离型两线式串行接口端口 B 差分输入/输出
IPB	64	I/O	隔离型两线式串行接口端口 B 差分输入/输出

绝对最大额定值

绝对最大额定值				
参数		最小值	最大值	单位
芯片电源电压 (V+至 V-)		16	105	V
相对 V-电压	Cn	0	7.5*n	V
	Sn	0	7.5*n	V
	C0	0	0.3	V
	IPA/IMA/IPB/IMB	0	VREG+0.3	V
	VDRIVE	0	7	V
C(n)与 C(n-1)之间		-0.3	7.5	V
S(n)与 C(n-1)之间		-0.3	7.5	V
C18 与 C15 之间		-0.3	20	V
C15 与 C9 之间		-0.3	20	V
C9 与 C6 之间		-0.3	20	V
C6 与 C3 之间		-0.3	20	V
C3 与 C0 之间		-0.3	20	V
流入/流出引脚电流(除 VREG,IPA,IMA,IPB,IMB,S(n)以外)			10	mA
IPA,IMA,IPB,IMB 流入/流出引脚的电流			30	mA
工作温度范围		-40	125	°C
贮存温度		-65	150	°C
结温			150	°C

注意，等于或超出上述绝对最大额定值可能会导致产品永久性损坏。这只是额定最值，不表示在这些条件下或者在任何其它超出本技术规范操作章节中所示规格的条件下，器件能够正常工作。长期在超出最大额定值条件下工作会影响产品的可靠性。电特性参数

ADC DC 规格					
参数	条件	最小值	典型值	最大值	单位
测量分辨率 LSB	电源电压, V+相对于 V-		0.1		mV
ADC 偏移电压			0.1		V
ADC 增益误差			0.01	0.02	%
标准模式中的总测量误差(TME)	C(n)至 C(n-1), GPIO(n)至 V-=0		±0.2		mV
	C(n)至 C(n-1), GPIO(n)至 V-=2.0			±1.4	mV
	C(n)至 C(n-1), GPIO(n)至 V-=3.3			±2.5	mV
	C(n)至 C(n-1), GPIO(n)至 V-=4.2			±3.0	mV
	C(n)至 C(n-1), GPIO(n)至 V-=5.0		±1		mV
	电池总和		±0.2	±0.75	%
	VREG 引脚		±0.1	±0.25	%
	内部温度		±5		°C
	VREF2 引脚		±0.02	±0.1	%
	数字电源电压 VREGD		±0.1	±1	%
滤波模式中的总测量误差(TME)	C(n)至 C(n-1), GPIO(n)至 V-=0		±0.1		mV
	C(n)至 C(n-1), GPIO(n)至 V-=2.0			±1.4	mV
	C(n)至 C(n-1), GPIO(n)至 V-=3.3			±2.5	mV
	C(n)至 C(n-1), GPIO(n)至 V-=4.2			±3.0	mV
	C(n)至 C(n-1), GPIO(n)至 V-=5.0		±1		mV
	电池总和		±0.2	±0.75	%
	VREG 引脚		±0.1	±0.25	%
	内部温度		±5		°C

	VREF2 引脚		± 0.02	± 0.1	%
	数字电源电压 VREGD		± 0.1	± 1	%
快速模式中的总测量误差(TME)	C(n)至 C(n-1), GPIO(n)至 V ₋ =0		± 2		mV
	C(n)至 C(n-1), GPIO(n)至 V ₋ =2.0			± 8	mV
	C(n)至 C(n-1), GPIO(n)至 V ₋ =3.3			± 10	mV
	C(n)至 C(n-1), GPIO(n)至 V ₋ =4.2			± 12.5	mV
	C(n)至 C(n-1), GPIO(n)至 V ₋ =5.0		± 10		mV
	电池总和		± 0.3	± 1	%
	VREG 引脚		± 0.3	± 1	%
	内部温度		± 5		°C
	VREF2 引脚		± 0.1	± 0.25	%
	数字电源电压 VREGD		± 0.2	± 2	%
输入范围	C(n), n=1 至 18	C(n-1)		C(n-1)+5	V
	C0		0		
	GPIO(n), n=1 至 9	0		5	V
未对输入进行测量时的输入漏电流 I _L	C(n), n=1 至 18		10	± 250	nA
	GPIO(n), n=1 至 9		10	± 250	nA
对输入进行测量时的输入漏电流 I _L	C(n), n=1 至 18		± 2		μA
	GPIO(n), n=1 至 9		± 2		μA
导线开路检测期间的输入电流	VREF1 引脚, 无负载	70	100	130	μA
电压基准规格					
第一基准电压 VREF1	VREF1 引脚, 无负载	3.2	3.4	3.6	V
第一基准电压 VREF1 TC	VREF1 引脚, 无负载		3		ppm/°C
第二基准电压 VREF2	VREF2 引脚, 无负载	2.990	3	3.010	V
	VREF2 引脚, 5.1k 负载至 V ₋	2.952	2.957	3.002	V

第二基准电压 VREF2 TC	VREF2 引脚，无负载			10		ppm/°C
一般 DC 规格						
V+电流 I _{VP}	状态：内核 =SLEEP，隔离 SPI=IDLE 状态：	VREG=0V		2.2		μA
		VREG=5V		1.4		μA
	内核=STANDBY			26.4		μA
	内核=REFUP			326.7		μA
	内核=MEASURE			420.3		μA
VREG 电源电流 I _{REG(CORE)}	状态：内核 =SLEEP，隔离 SPI=IDLE	VREG=5V		0.9		μA
	状态：内核=STANDBY			17.8		μA
	状态：内核=REFUP			0.3		mA
	状态：内核=MEASURE			13.2		mA
VREG 电源电流(假 如隔离 SPI 处于 READY/ACTIVE 状 态) 注：ACTIVE 状态 电流 假设 t _{CLK} =1us I _{REG(ISOSPI)}	MT9805-1：ISOMD=0， R _{B1} +R _{B2} =2k	READY	4.4	5.5	6.7	mA
		ACTIVE	6.8	8.2	9.7	mA
	MT9805-1：ISOMD=1， R _{B1} +R _{B2} =2k	READY	7.8	9.4	11.4	mA
		ACTIVE	12.2	13.6	16.0	mA
	MT9805-1：ISOMD=0， R _{B1} +R _{B2} =20k	READY	1.3	2.3	3.4	mA
		ACTIVE	1.8	2.8	4.0	mA
	MT9805-1：ISOMD=1， R _{B1} +R _{B2} =20k	READY	2.5	4.0	5.9	mA
		ACTIVE	3.2	4.9	7.0	mA
V+电源输入	满足 TME 规格(注 6)		16	60	90	V
VREG 电源电压	TME 电源抑制<1mV/V		4.5	5	5.5	V
VDRIVE 输出电压	供应 1uA 电流		5.2	5.6	6.0	V
	供应 500uA 电流		5.1	5.6	6.1	V
数字电源电压 VREGD			2.7	3.0	3.6	V

放电开关导通电阻	VCELL=3V		4	10	Ω
热停机温			150		$^{\circ}\text{C}$
看门狗定时器引脚为低电平 $V_{OL(WDT)}$	WDT 引脚吸收 4mA 电流			0.4	V
通用 I/O 引脚为低电平 $V_{OL(GPIO)}$	GPIO 引脚吸收 4mA 电流(用作数字输出)			0.4	V
测量+校准周期时间(当从标准模式中的 REFUP 状态开始时) t_{CYCLE}	测量 18 节电池	4.22	4.85	6	ms
	测量 3 节电池	0.70	0.81	0.86	ms
	测量 18 节电池和 2 个 GPIO 输入	5.63	6.47	6.85	ms
测量+校准周期时间(当从滤波模式中的 REFUP 状态开始时) t_{CYCLE}	测量 18 节电池	266.70	306.55	324.94	ms
	测量 3 节电池	44.45	51.09	54.16	ms
	测量 18 节电池和 2 个 GPIO 输入	355.60	408.73	433.26	ms
测量+校准周期时间(当从快速模式中的 REFUP 状态开始时) t_{CYCLE}	测量 18 节电池	2.31	2.65	2.81	ms
	测量 3 节电池	0.38	0.44	0.47	ms
	测量 18 节电池和 2 个 GPIO 输入	3.07	3.53	3.75	ms
稳压器启动时间 t_{WAKE}	VREG 由驱动引脚产生		200	400	us
看门狗或软件放电定时器 t_{SLEEP}	DTEN 引脚=0 或 DCTO[3:0]=0000	1.7	2	2.3	s
	DTEN 引脚=1 且 DCTO[3:0]≠0000	0.5		120	min
基准 VREF1 唤醒时间 t_{REFUP}	状态: 内核=STANDBY	2.7	3.5	4.4	ms
	状态: 内核=REFUP			0	ms
ADC 时钟频率 f_s			2.2		MHz
SPI 接口 DC 规格					
SPI 引脚数字输入电压为高电平 $V_{IH(SPI)}$	引脚 CSBI、SCLK、SDI	2.3			V
SPI 引脚数字输入电压为低电平 $V_{IL(SPI)}$	引脚 CSBI、SCLK、SDI			0.8	V
配置引脚数字输入	引脚 ISOMD, SWTEN, GPIO1 至 GPIO5	2.7			V

电压为高电平 $V_{IH(CFG)}$					
配置引脚数字输入 电压为低电平 $V_{IL(CFG)}$	引脚 ISOMD, SWTEN, GPIO1 至 GPIO5			1.2	V
数字输入电流 I_{LEAK}	引脚 CSBI, SCLK, SDI, ISOMD, SWTEN			± 1	μA
数字输出为低电平 $V_{OL(SDO)}$	引脚 SDO 吸收 1mA 电流			0.3	V
隔离 SPI DC 规格					
IBIAS 引脚上的电 压 V_{IBIAS}	READY/ACTIVE 状态	1.9	2.0	2.1	V
	IDLE 状态		0		V
隔离式接口偏置电 流 I_B	RBIAS=2k 至 20k	0.1		1.0	mA
隔离式接口电流增 益 A_{IB}	$V_A \leq 1.6V, I_B = 1mA$	22	25	28	mA/mA
	$V_A \leq 1.6V, I_B = 0.1mA$	22	25	28	mA/mA
	$V_A = V_{IP} - V_{IM} $			1.6	V
ICMP 引脚上的门 限设定电压 V_{ICMP}	$V_{TCMP} = A_{TCMP} \cdot V_{ICMP}$	0.2		1.5	V
ICMP 引脚上的输 入漏电流 $I_{LEAK(ICMP)}$	$V_{ICMP} = 0V$ 至 V_{REG}			± 1	μA
IP 和 IM 引脚上的 漏电流 $I_{LEAK(IP/IM)}$	IDLE 状态, V_{IP} 或 $V_{IM} = 0V$ 至 V_{REG}			± 1	μA
接收器比较器门限 电压增益 A_{TCMP}	$V_{CM} = V_{REG}/2$ 至 $V_{REG} - 0.2V$, $V_{ICMP} = 0.2V$ 至 $1.5V$	0.4	0.5	0.6	V/V
接收器共模偏置 V_{CM}	IP/IM 未执行驱动操作	3.2			V
接收器输入电阻 R_{IN}	单端至 IPA, IMA, IPB, IMB	27	35	43	k Ω
隔离 SPI 空闲/唤醒规格					
差分唤醒电压 V_{WAKE}	$t_{DWELL} = 240ns$	200			mV

唤醒检测之前 V_{WAKE} 的停留时间 t_{DWELL}	$V_{WAKE}=200mV$	240			ns
识别唤醒之后的启动时间 t_{READY}				10	us
空闲超时周期 t_{IDLE}		4.3	5.5	6.7	ms
隔离 SPI 脉冲定时规格					
芯片选择半脉冲宽度 $t_{1/2PW(CS)}$		120	150	180	ns
芯片选择脉冲反向延迟 $t_{INV(CS)}$				200	ns
数据半脉冲宽度 $t_{1/2PW(D)}$		40	50	60	ns
数据脉冲反向延迟 $t_{INV(D)}$				70	ns
SPI 定时要求					
SCLK 周期 t_{CLK}		1			us
SDI 建立时间(在 SCLK 上升沿之前) t_1		25			ns
SDI 保持时间(在 SCLK 上升沿之后) t_2		25			ns
SCLK 为低电平 t_3	$t_{CLK}=t_3+t_4 \geq 1s$	200			ns
SCLK 为高电平 t_4	$t_{CLK}=t_3+t_4 \geq 1s$	200			ns
CSBI 上升沿至 CSBI 下降沿 t_5		0.65			us
SCLK 上升沿至 CSBI 上升沿 t_6		0.8			us
CSBI 下降沿至 SCLK 上升沿 t_7		1			us
隔离 SPI 定时规格					
SCLK 下降沿至 SDO 有效 t_8				60	ns

SCLK 上升沿至短 ±1 脉冲传输 t_9				50	ns
CSBI 转换至长±1 脉冲传输 t_{10}				60	ns
CSBI 上升沿至 SDO 上升 t_{11}				200	ns
数据返回延迟 t_{RTN}			430	525	ns
芯片选择菊链延迟 $t_{DSY(CS)}$			150	200	ns
数据菊链延迟 $t_{DSY(D)}$			300	360	ns
数据菊链滞后(相对 于芯片选择) t_{LAG}		0	35	70	ns
数据至芯片选择脉 冲调节器 $t_{6(GOV)}$		0.8		1.05	μs

特性曲线

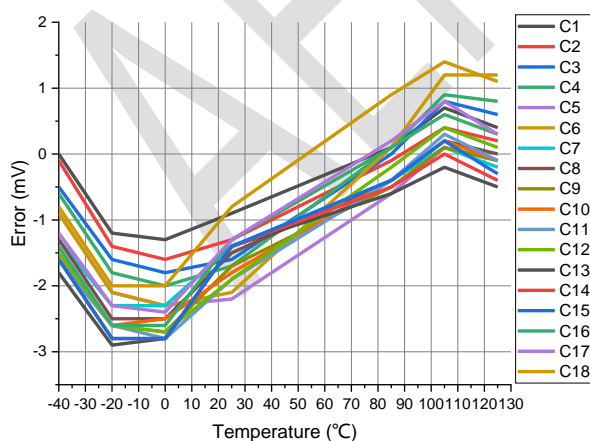


图 3 滤波模式测量误差 (Cell=3V)

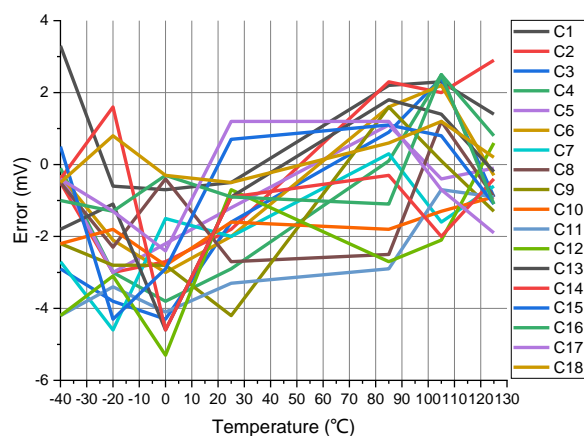


图 4 快速模式测量误差 (Cell=3V)

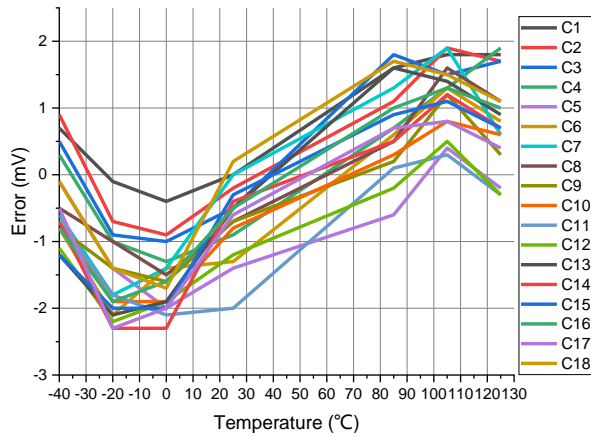


图 5 标准模式测量误差 (Cell=3V)

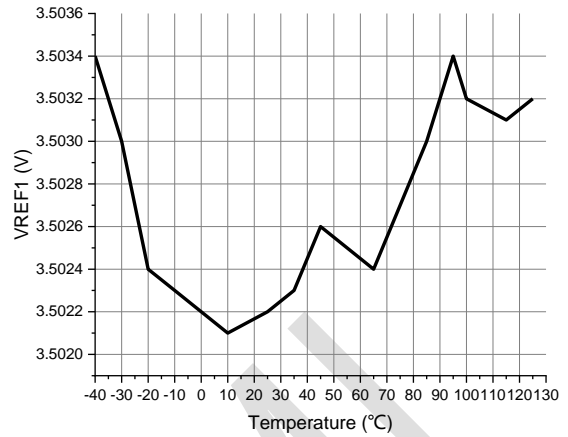


图 6 VREF1 在不同温度下的电压

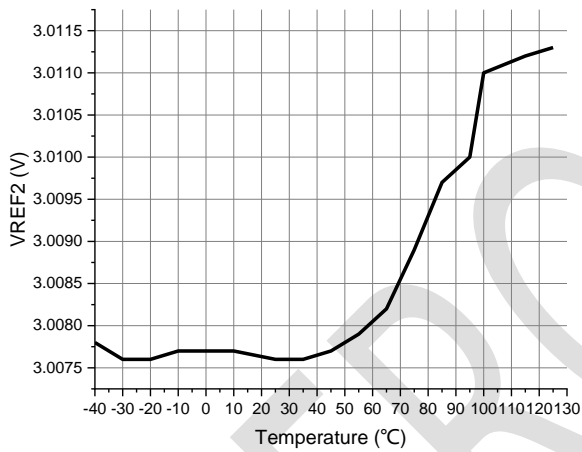


图 7 VREF2 在不同温度下的电压

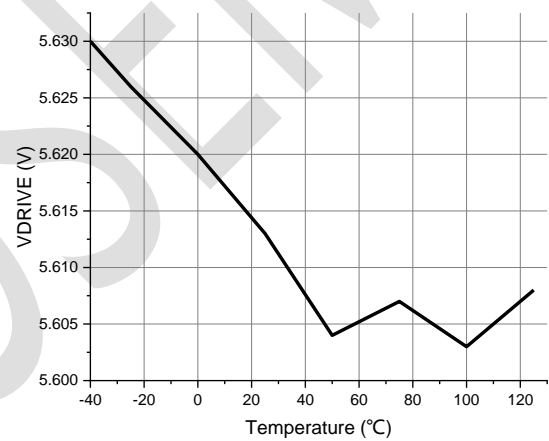


图 8 VDRIVE 在不同温度下的电压

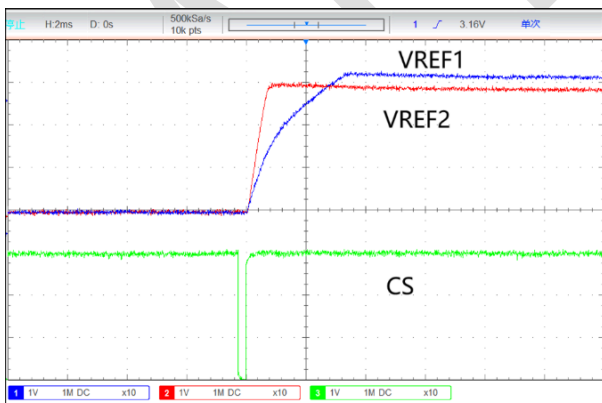


图 9 VREF1 和 VREF2 启动过程

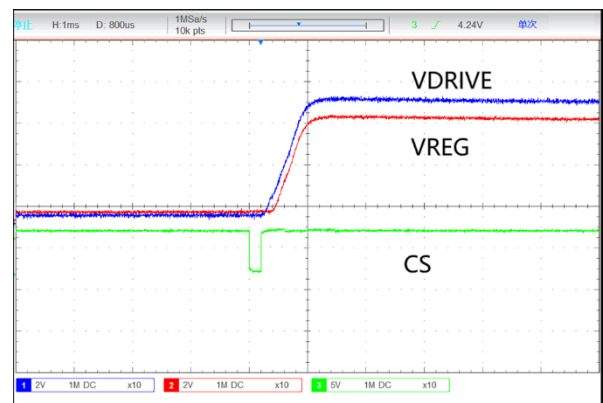


图 10 VDRIVE 和 VREG 启动过程

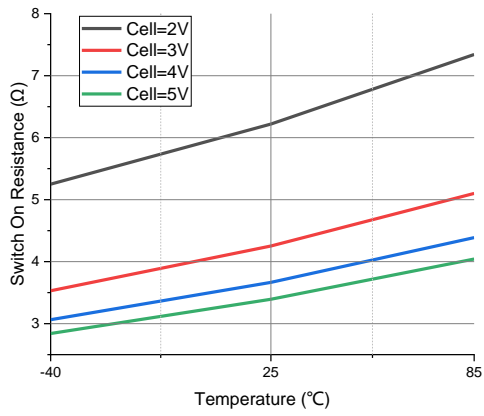


图 11 不同温度下内部均衡开关导通电阻

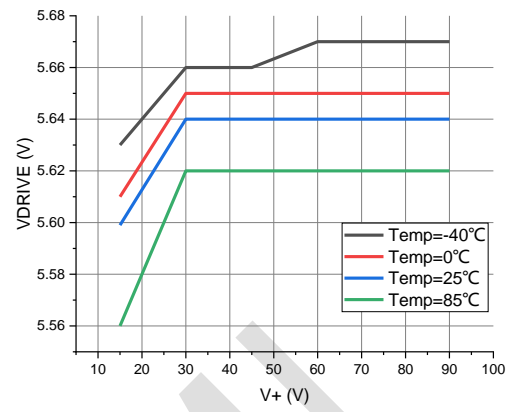


图 12 不同 V+ 输入下的 VDRIVE 电压

工作

工作原理

MT9805 的操作分为两个单独的部分：内核电路和隔离 SPI 电路。这两个部分都具有独立的操作状态以及一个停机超时。

MT9805 内核状态描述

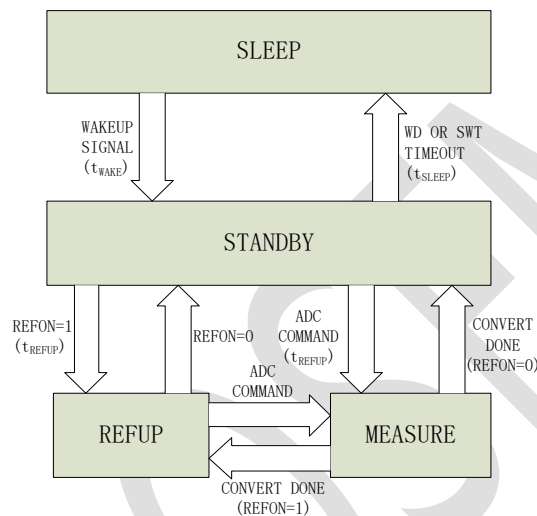


图 13 MT9805 内核工作状态

SLEEP 状态

基准和 ADC 断电。看门狗定时器已超时。软件放电定时器被停用或暂停。电源电流被减小至最低水平。隔离 SPI 端口将处于 IDLE 状态。

如果接收到一个 WAKEUP 信号，则 MT9805 将进入 STANDBY 状态。

STANDBY 状态

基准和 ADC 被关断。看门狗定时器和/或软件放电定时器开始计时。VDRIVE 引脚通过一个外部晶体管将 VREG 引脚供电至 5V，或者也可以利用一个外部电源来给 VREG 供电。

如果接收到一个有效的 ADC 命令或配置寄存器组的 REFON 位中被设定为 1，则 IC 暂停工作(持续时间为 t_{REFUP})以使基准上电，并随后进入 REFUP 或 MEASURE 状态。否则，在经历了 t_{SLEEP} 之后(此时看门狗和软件放电定时器均已计时终止)，MT9805 将返回 SLEEP 状态。假如软件放电定时器被停用，则仅有看门狗定时器与此相关。

REFUP 状态

为达到该状态，配置寄存器组中的 REFON 位必须设定为 1(采用 WRCFGA 命令)。ADC 被关断，基准被上电

以使 MT9805 能够更快地启动 ADC 转换(相比于从 STANDBY 状态)。当接收到一个有效的 ADC 命令时, IC 变至 MEASURE 状态以开始执行转换操作。否则, MT9805 将在 REFON 位被设定为 0(采用 WRCFGA 和命令手动完成, 或在看门狗定时器计时终止时自动实现)时返回 STANDBY 状态。(若两个定时器均计时终止, 则 MT9805 将直接进入 SLEEP 状态)。

MEASURE 状态

MT9805 在该状态中执行 ADC 转换操作。基准和 ADC 工作。ADC 转换完成之后, MT9805 将转换至 REFUP 或 STANDBY 状态(这取决于 REFON 位)。通过设定 REFON=1 利用 REFUP 状态, 可更快地启动另外的 ADC 转换。注: 非 ADC 命令不会引起内核状态转换。只有一个 ADC 转换或诊断命令可以把内核置于 MEASURE 状态。

隔离 SPI 状态描述

IDLE 状态

隔离 SPI 端口断电。隔离 SPI 端口 A 或端口 B 接收到一个 WAKEUP 信号时, 隔离 SPI 进入 READY 状态。如果内核处于 STANDBY 状态, 则该转换过程需要 t_{READY} , 因为 VDRIVE 和 VREG 引脚已被施加偏压。如果内核在隔离 SPI 接收到一个 WAKEUP 信号时处于 SLEEP 状态, 则隔离 SPI 将在 t_{WAKE} 时间之后转换至 READY 状态。

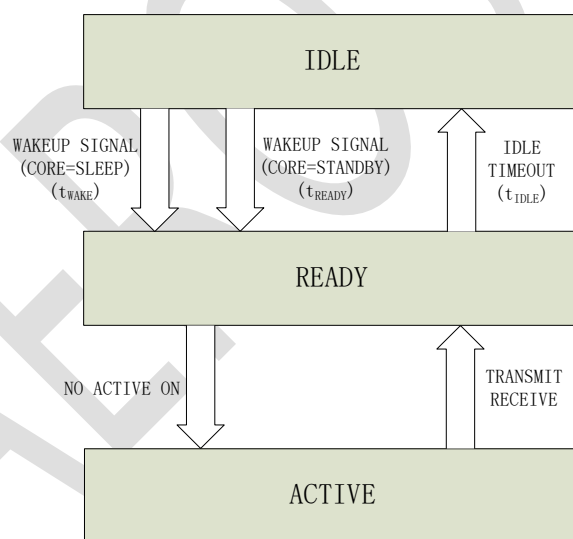


图 14 MT9805 隔离 SPI 工作状态

READY 状态

在 READY 模式下, isoSPI 端口处于随时准备通讯的状态。该状态中的串行接口电流取决于器件 ISOMD 引脚的状态以及 $R_{BIAS}=R_{B1}+R_{B2}$ (连接至 IBIAS 引脚的外部电阻器)。

如果在端口 A 或端口 B 上超过 $t_{IDLE}=5.5ms$ 没有动作(即: 无 WAKEUP 信号), 则 MT9805 进入 IDLE 状态。当串行接口发送或接收数据时, MT9805 变至 ACTIVE 状态。

ACTIVE 状态

MT9805 采用一个或两个隔离 SPI 端口来发送/接收数据。串行接口在该状态中消耗最大的功率。电源电流随时钟频率而增大，随着隔离 SPI 脉冲的密度增加。

功耗

MT9805 通过两个引脚来供电：V+和 VREG。V+输入所需的电压高于或等于顶端电池电压，而且其负责为内核电路的高电压组件供电。V+可以直接连接至电池组的顶端电池，或者至一个外部电源。VREG 输入需要 5V 电压并用于为其余的内核电路及隔离 SPI 电路供电。VREG 输入可通过一个由已调 VDRIVE 输出引脚驱动的外部晶体管供电。或者，也可以采用一个外部电源来给 VREG 供电。

$$I_{REG}=I_{REG(CORE)}+I_{REG(隔离\ SPI)}$$

表 3 内核电源电流

状态		I _{V+}	I _{REG(CORE)}
SLEEP	VREG=0V	7uA	0uA
	VREG=5V	4uA	3uA
STANDBY		32uA	30uA
REFUP		550uA	900uA
MEASURE		900uA	16mA

在 SLEEP 状态中，如果由一个外部电源供电，则 VREG 引脚将吸收大约 3uA 的电流。否则，将由 V+引脚提供所需的电流。

表 4 隔离 SPI 电源电流公式

隔离 SPI 状态	ISOMD 连接	I _{REG(隔离 SPI)}
IDLE	不适用	0mA
READY	VREG	$2.8mA+5 \cdot I_B$
	V-	$1.6mA+3 \cdot I_B$
ACTIVE	VREG	写： $2.8mA+5 \cdot I_B+(2 \cdot I_B+0.4mA) \cdot (1\mu s/t_{CLK})$
		读： $2.8mA+5 \cdot I_B+(3 \cdot I_B+0.5mA) \cdot (1\mu s/t_{CLK})$
	V-	$1.6mA+3 \cdot I_B+(2 \cdot I_B+0.2mA) \cdot (1\mu s/t_{CLK})$

功耗随着操作状态而改变，表 3 和表 4 给出了用于粗略估计每种状态中的电源引脚电流。V+引脚电流仅取决于内核状态(而不是隔离 SPI 状态)。VREG 引脚电流则同时取决于内核状态及隔离 SPI 状态，且因此可被分为两个分量。隔离 SPI 接口仅从 VREG 引脚吸收电流。

ADC 操作

在 MT9805 的内部有三个 ADC。当测量 18 个电池时这三个 ADC 同时运作。只有一个 ADC 用于测量通用输入端口。当需要区分三个电路时(比如在时序图中)，下面的讨论将采用 ADC1、ADC2 和 ADC3 进行指代。

ADC 模式

配置寄存器组中的 ADCOPT 位(CFGRA0[0])和转换命令中的模式选择位 MD[1:0]共同为 ADC 提供了 8 种操作模式，其对应于不同的过采样比(OSR)。表 5 汇总了这些模式的精度。在每种模式中，ADC 首先执行校准，然后进行测量。模式的名称基于 ADC 测量的-3dB 带宽。

表 5 ADC 滤波器带宽和精度

模式	-3dB 滤波器 BW	-40dB 滤波器 BW	3.3V、25°C 条 件下的 TME 规 格	3.3V、-40°C、 85°C 条件下的 TME 规格
27kHz(快速模式)	27kHz	84kHz	±4.7mV	±4.7mV
14kHz	13.5kHz	42kHz	±4.7mV	±4.7mV
7kHz(标准模式)	6.8kHz	21kHz	±1.2mV	±2.2mV
3kHz	3.4kHz	10.5kHz	±1.2mV	±2.2mV
2kHz	1.7kHz	5.3kHz	±1.2mV	±2.2mV
1kHz	845Hz	2.6kHz	±1.2mV	±2.2mV
422Hz	422Hz	1.3kHz	±1.2mV	±2.2mV
26Hz(滤波模式)	26Hz	82Hz	±1.2mV	±2.2mV

7kHz 模式(标准)

在该模式中，ADC 具有高分辨率和低 TME(总测量误差)。是速度与精度的最优组合作为标准操作模式。

27kHz 模式(快速)

在该模式中，ADC 具有最大的吞吐量，但 TME(总测量误差)略有增加。因此这种模式也被称为快速模式。速度的提高源自于过采样比的下降。这将导致噪声和平均测量误差的增加。

26Hz 模式(滤波)

在该模式中，通过增加 OSR 将 ADC 数字滤波器-3dB 频率降低至 26Hz。该模式由于其低的-3dB 频率的原因也被称为滤波模式。其精度类似于具有较低噪声的 7kHz(标准)模式。

14kHz、3kHz、2kHz、1kHz 和 422Hz 模式

14kHz、3kHz、2kHz、1kHz 和 422Hz 模式提供了用于将 ADC 数字滤波器-3dB 频率分别设定在 13.5kHz、3.4kHz、1.7kHz、845Hz 和 422Hz 的额外选项。14kHz 模式的精度类似于 27kHz(快速)模式。3kHz、2kHz、1kHz 和 422Hz 模

式的精度类似于 7kHz(标准)模式。

这些模式的转换时间列于表 7。如果内核处于 STANDBY 状态，则需一个附加的 t_{SETUP} 时间以在 ADC 转换开始之前对基准进行上电。倘若配置寄存器组中的 REFON 位被设定为 1，则基准在 ADC 转换操作之间保持上电，于是内核在一个 t_{REFUP} 延迟之后处于 REFUP 状态。这样，后续的 ADC 命令在开始 ADC 转换之前将没有 t_{REFUP} 延迟。

ADC 范围和分辨率

C 输入和 GPIO 输入具有相同的范围和分辨率。MT9805 内部的 ADC 具有一个 $-0.82V$ 至 $5.73V$ 的近似范围。负读数被取整至 $0V$ 。数据的格式为 16 位无符号整数，其中 LSB 代表 $100\mu V$ 。因此， $0x80E8$ (十进制为 33,000)的读数表示一个 $3.3V$ 的测量值。

表 6 ADC 测量范围和精度

模式	全标度	规定范围	精准范围	LSB	格式	最大噪声	无噪声分辨率
27kHz(快速)	- 0.82 至 5.73	0V 至 5V	0.5V 至 4.5V	100 μV	无符号 16 位	$\pm 4mV$	10Bits
14kHz						$\pm 1mV$	12Bits
7kHz(标准)						$\pm 250\mu V$	14Bits
3kHz						$\pm 150\mu V$	14Bits
2kHz						$\pm 100\mu V$	15Bits
1kHz						$\pm 100\mu V$	15Bits
422kHz							
26Hz(滤波)						$\pm 50\mu V$	16Bits

$\Delta\Sigma$ ADC 具有与输入电压相关的量化噪声，特别是在低过采样比(OSR)条件下，例如：在快速模式中。在某些 ADC 模式中，当输入电压接近 ADC 范围的上限和下限时，量化噪声增加。

ADC 的规定范围为 $0V$ 至 $5V$ 。在表 6 中，ADC 的精准范围被定义为 $0.5V$ 至 $4.5V$ 中的随意值。这是那种甚至在较低 OSR 模式中量化噪声都将相对恒定的范围。表 6 汇总了该范围中的总噪声(对于所有 8 种 ADC 操作模式)。另外，表中还给出了无噪声分辨率。例如：标准模式中的 14 位无噪声分辨率意味着，对于一个 DC 输入，上部的 14 个位将是无噪声的，但第 15 和 16 个最低有效位(LSB)将跳变。

ADC 范围与电压基准值的关系

典型的 $\Delta\Sigma$ ADC 具有一个恰为电压基准值两倍的范围，而 ADC 测量误差与电压基准的误差成正比。MT9805 ADC 并不是典型的 ADC。VREF1 的绝对值被调高或调低以补偿 ADC 中的增益误差。当在 7kHz(标准)模式中测量 $3.300V$ 时，总测量误差的 $25^{\circ}C$ 规格值为 $\pm 1.2mV$ ，而 VREF1 的 $25^{\circ}C$ 规格值则为 $3.400V \pm 200mV$ 。

测量电池电压(ADCV 命令)

ADCV 命令用于启动电池输入(引脚 C0 至 C18)的测量。该命令具有用于选择测量信道数目和 ADC 模式的选项。

图 15 示出了用于测量所有 18 节电池的 ADCV 命令的定时。在接收到测量所有 18 节电池的 ADCV 命令之后，ADC3 顺序测量上部的 6 节电池，ADC2 则顺序测量中部的 6 节电池，而 ADC1 则顺序测量下部的 6 节电池。在电池测量完成之后，对每个通道进行校准以消除任何的偏移误差。

图 16 示出了仅测量 3 节电池的 ADCV 命令的定时。

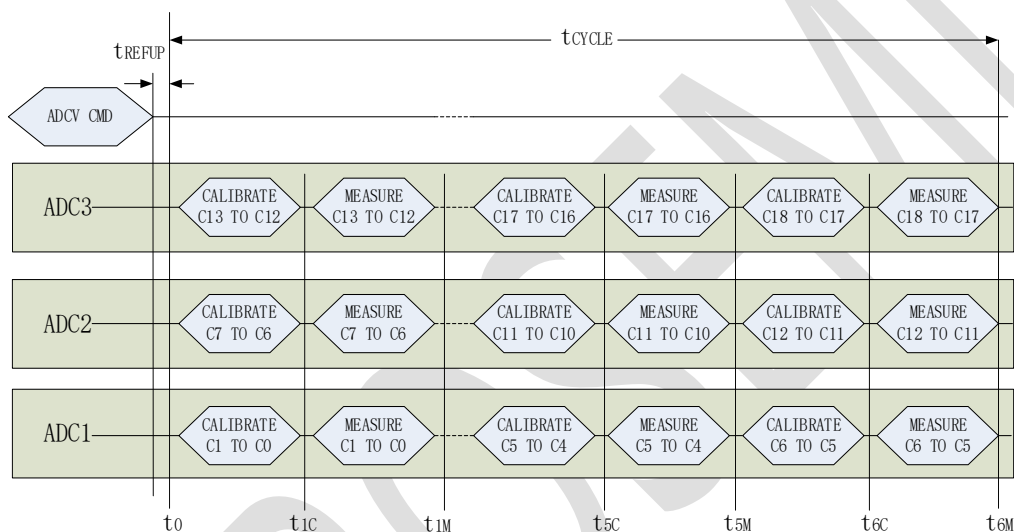


图 15 用于测量所有 18 节电池的 ADCV 命令的定时

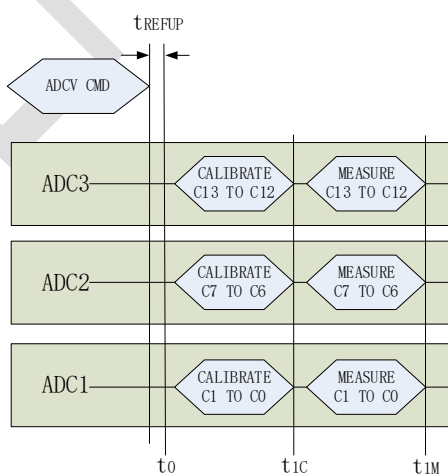


图 16 用于测量 3 节电池的 ADCV 命令的定时

欠压/过压监视

每当测量 C 输入时，将测量结果与存储器中存储的欠压和过压门限进行比较。如果电池电压的测量读数高于过压限值，则存储器中的一个位被设定为标记。同样，低于欠压门限的测量结果也会导致设定一个标记。过压和欠压门限存储在配置寄存器组 A 中。标记则存储于状态寄存器组 B 和辅助寄存器组 D 之中。

辅助(GPIO)测量(ADAX 命令)

ADAX 命令用于启动 GPIO 输入的测量。该命令具有用于选择要测量的 GPIO 输入(GPIO1-9)及 ADC 模式的选项。另外，ADAX 命令还负责测量第二基准。在 ADAX 命令中有单独测量每个 GPIO 和第二基准或者在单个命令周期中测量所有 9 个 GPIO 和第二基准的选项。所有的辅助测量均相对于 V_{引脚}电压。通过把温度传感器连接至 GPIO，该命令可用于读取外部温度。这些传感器可以采用同样由 ADAX 命令进行测量的第二基准来供电，从而获得精准的比例式测量结果。

图 17 示出了用于测量所有 GPIO 和第二基准的 ADAX 命令的定时。由于所有的 10 项测量均由 ADC1 单独完成，因此 ADAX 命令的转换时间与 ADCV 命令的相似。

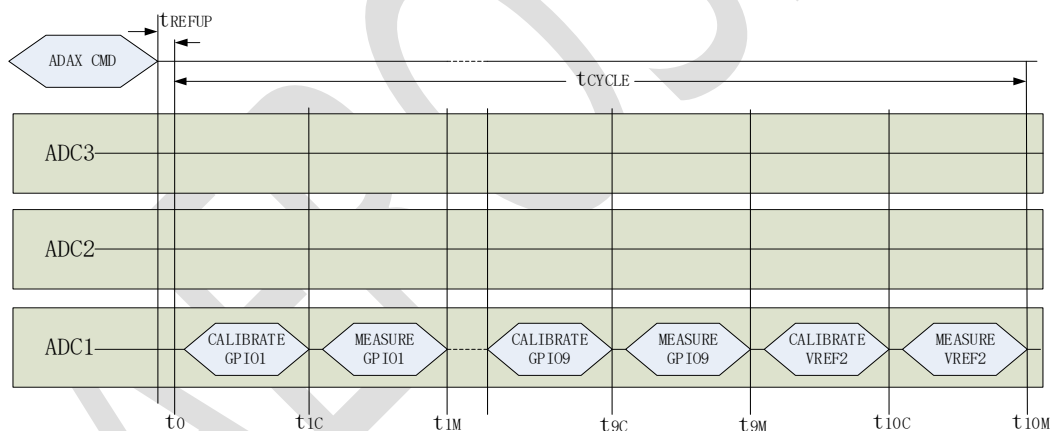


图 17 测量所有 GPIO 和第二基准的 ADAX 命令的定时

辅助(GPIO)测量数字冗余(ADAXD 命令)

ADAXD 命令的操作与 ADAX 命令类似，只是使用数字冗余执行附加诊断。测量结果以冗余计算。具体参见“数字冗余诊断”小节。

测量电池电压和 GPIO(ADCVAX 命令)

ADCVAX 命令组合了 18 节电池测量与 2 个 GPIO 测量(GPIO1 和 GPIO2)。当电流传感器连接至 GPIO1 和 GPIO2

输入时，该命令可简化电池电压和电流测量的同步。图 18 示出了 ADCVAX 命令的定时。见有关“命令”的章节以了解 ADCVAX 命令格式。

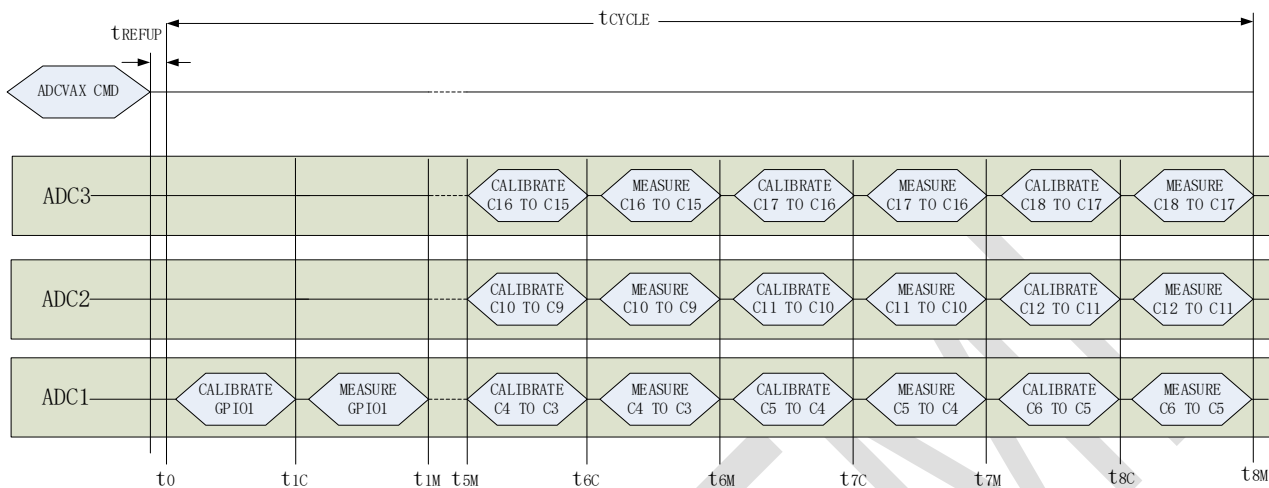


图 18 ADCVAX 命令的定时

数据采集系统诊断

电池监视数据采集系统由多路复用器、ADC、第一基准、数字滤波器和存储器组成。为了确保长期的可靠性能，有几个诊断命令可用于验证这些电路的正确运作。

测量内部器件参数(ADSTAT 命令)

ADSTAT 命令是一种用于测量以下内部器件参数的命令：所有电池的总和(SOC)、内部芯片温度(ITMP)、仿真电(VA)和数字电源(VD)。

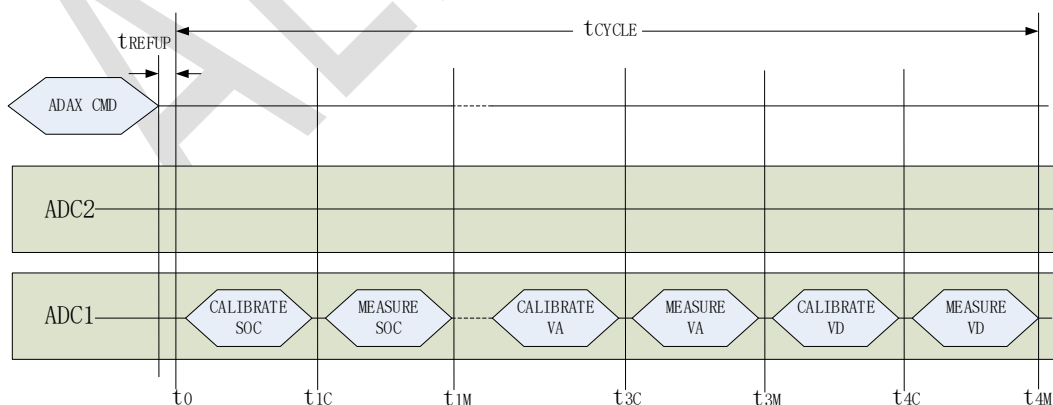


图 19 测量 SOC、ITMP、VA、VD 的 ADSTAT 命令的定时

这些参数在下一节描述。所有 8 种 ADC 模式均可用于这些转换。见有关“命令”的章节以了解 ADSTAT 命令格

式。图 19 示出了测量所有 4 种内部器件参数的 ADSTAT 命令的定时。

电池测量结果的总和：所有电池测量结果的总和是 C18 和 C0 之间的电压(具有一个 30:1 的衰减)。电池测量结果总和(SC)的 16 位 ADC 值存储在状态寄存器组 A 当中。利 SC 值，可由下式得出所有电池电压测量结果之和：

$$\text{所有电池之和} = \text{SOC} \cdot 30 \cdot 100\mu\text{V}$$

内部芯片温度：ADSTAT 命令能够测量内部芯片温度。芯片温度测量结果(ITMP)的 16 位 ADC 值存储在状态寄存器组 A 中。根据 ITMP 可利用下式计算出实际的芯片温度：

$$\text{内部芯片温度}(\text{°C}) = (\text{ITMP}) \cdot 100\mu\text{V} / (7.5\text{mV}) \cdot \text{°C} - 273\text{°C}$$

电源测量：ADSTAT 命令还用于测量模拟电源(VREG)和数字电源(VREGD)。

模拟电源测量结果(VA)的 16 位 ADC 值存储在状态寄存器 A 中。数字电源测量结果(VD)的 16 位 ADC 值存储在状态寄存器组 B 中。利用 VA 和 VD 可由下式求出电源测量结果：

$$\text{模拟电源测量(VREG)} = \text{VA} \cdot 100\mu\text{V}$$

$$\text{数字电源测量(VREGD)} = \text{VD} \cdot 100\mu\text{V}$$

VREG 的标称范围为 4.5V 至 5.5V。VREGD 的标称范围为 2.7V 至 3.6V。

测量内部器件参数数字冗余(ADSTATD 命令)

ADSTATD 命令的操作与 ADSTAT 命令类似，只是使用数字冗余执行附加诊断。具体参见“数字冗余诊断”小节。

测量电池电压和 GPIO(ADCVSC 命令)

ADCVAX 命令组合了 18 节电池测量与电池包总压测量。图 20 示出了 ADCVSC 命令的定时。见有关“命令”的章节以了解 ADCVSC 命令格式。

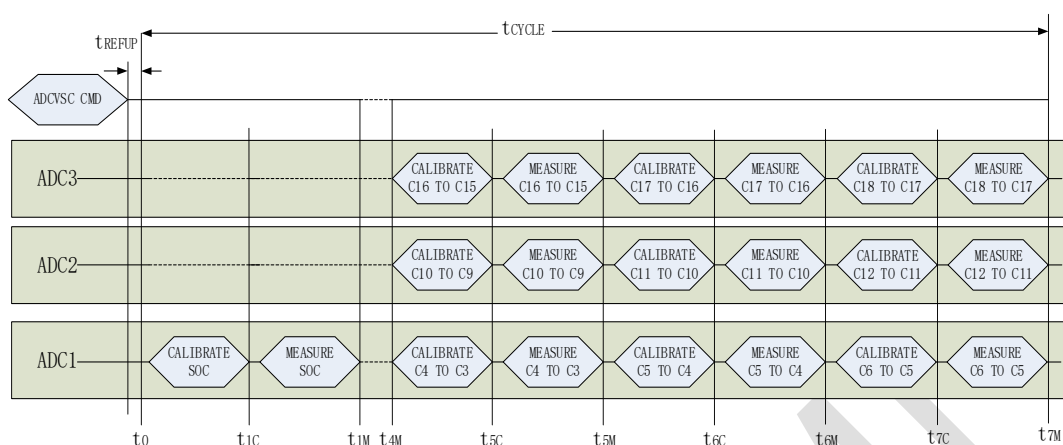


图 20 ADCVSC 命令的定时

数字冗余诊断

每个 ADC 都带有数字滤波器，MT9805 提供第 4 个数字滤波器，用于数字冗余诊断。除了 ADAX 和 ADSTAT 以外所有 ADC 采集和自测试指令。

包括 ADCV,ADOW,CVST,ADOL,ADAXD,AXOW,AXST,ADSTATD,STATST,ADCVAX 和 ADCVSC, 都可以工作于数字冗余诊断。在每次测量结束时，比较两个结果，如果出现任何不匹配，则值为 0xFF0X($\geq 6.528V$) 写入结果寄存器。该值超出 ADC 的箝位范围，主机应将其识别为故障指示。最后四位用于指示结果值的哪个字节不匹配，如表 9 所示。

因为只有一个冗余的数字滤波器，一次只能对一个 ADC 应用冗余。默认情况下，MT9805 自动选择 ADC 路径冗余。用户可以选择 ADC 冗余路径通过写入 PS[1:0]，配置寄存器组 B 中的 Bits[1:0]，表 8 给出了冗余数字滤波器诊断通路可能的选择。

表 8 ADC 数字冗余通道选择

	PS[1:0]=00		PS[1:0]=01		PS[1:0]=10		PS[1:0]=11	
采集	通道选择	冗余采集	通道选择	冗余采集	通道选择	冗余采集	通道选择	冗余采集
通道 1, 7, 13	ADC1	通道 1	ADC1	通道 1	ADC2	通道 7	ADC3	通道 13
通道 2, 8, 14	ADC2	通道 8	ADC1	通道 2	ADC2	通道 8	ADC3	通道 14
通道 3, 9, 15	ADC3	通道 15	ADC1	通道 3	ADC2	通道 9	ADC3	通道 15
通道 4, 10, 16	ADC1	通道 4	ADC1	通道 4	ADC2	通道 10	ADC3	通道 16
通道 5, 11, 17	ADC2	通道 11	ADC1	通道 5	ADC2	通道 11	ADC3	通道 17
通道 6, 12, 18	ADC3	通道 18	ADC1	通道 6	ADC2	通道 12	ADC3	通道 18

通道 7 (ADOL)	ADC2	通道 7	ADC1	通道 7	ADC2	通道 7	ADC3	/
通道 13 (ADOL)	ADC2	通道 13	ADC1	/	ADC2	通道 13	ADC3	通道 13
GPIO1~9	ADC1	GPIO1~9	ADC1	GPIO1~9	ADC2	/	ADC3	/
SC	ADC1	SC	ADC1	SC	ADC2	/	ADC3	/
ITMP	ADC1	ITMP	ADC1	ITMP	ADC2	/	ADC3	/
VA	ADC1	VA	ADC1	VA	ADC2	/	ADC3	/
VD	ADC1	VD	ADC1	VD	ADC2	/	ADC3	/

表 9 数字冗余说明

寄存器结果	说明
16'b1111_1111_0000_0XXX	数字滤波器输出结果的 15~12 位相同
16'b1111_1111_0000_1XXX	数字滤波器输出结果的 15~12 位有不同
16'b1111_1111_0000_X0XX	数字滤波器输出结果的 11~8 位相同
16'b1111_1111_0000_X1XX	数字滤波器输出结果的 11~8 位有不同
16'b1111_1111_0000_XX0X	数字滤波器输出结果的 7~4 位相同
16'b1111_1111_0000_XX1X	数字滤波器输出结果的 7~4 位有不同
16'b1111_1111_0000_XXX0	数字滤波器输出结果的 3~0 位相同
16'b1111_1111_0000_XXX1	数字滤波器输出结果的 3~0 位有不同

重叠单元测量(ADOL 命令)

ADOL 命令用于 ADC1 和 ADC2 同时测量第 7 节电池电压，ADC2 和 ADC3 同时测量第 13 节电池电压。主机可以将两个结果相互比较。ADC1 的结果会储存在电池电压寄存器 C 的 C7V[15:0]，ADC2 的结果会储存在电池电压寄存器 C 的 C8V[15:0]，ADC2 的结果会储存在电池电压寄存器 C 的 C13V[15:0]，ADC3 的结果会储存在电池电压寄存器 C 的 C14V[15:0]。ADOL 采集时序图如图 21 所示。

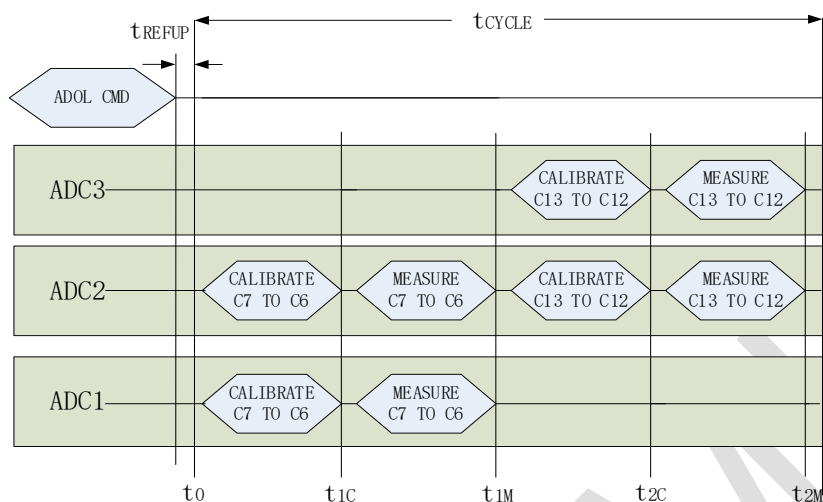


图 21 ADOL 命令的定时

精度检查

测量一个独立的电压基准是验证数据采集系统精度的最佳方法。MT9805 包括用于该目的第二个基准。ADAX 命令将启动第二基准的测量。测量结果被置于辅助寄存器组 B 中。测量结果的范围取决于 ADC 测量精度和第二基准的精度，包括热迟滞和长期漂移。如果出现 2.980 至 3.020 范围之外的读数，则表示系统超出了其规定的容差范围。

多路复用器(MUX)译码器检查

诊断命令 DIAGN 负责确保每个多路复用器通道的正确运作。该命令循环通过所有的通道，并在有任何通道译码器发生故障的情况下在状态寄存器组 B 中将 MUXFAIL 位设定 1。如果通道译码器通过了测试，则 MUXFAIL 位被设定 0。另外，在上电(POR)时或一个 CLRSTAT 命令之后，MUXFAIL 位也被设定为 1。

DIAGN 命令需要大约 400us(假如内核处于 REFUP 状态)和大约 4.5ms(假如内核处于 STANDBY 状态)的时间完成。在“轮询方法”部分中描述的轮询法可用于确定 DIAGN 命令的完成。

数字滤波器检查

$\Delta\Sigma$ ADC 由一个 1 位脉冲密度调制器和一个位于其后的数字滤波器组成。当具有较高输入电压时，经过脉冲密度调制的输出位流具有相对较高“1”的百分比。数字滤波器把该高频 1 位数据流转换为单个 16 位字。

自测试命令用于验证数字滤波器和存储器的操作。1 位脉冲密度调制器的输出被一个 1 位测试信号所取代。该测试信号通过数字滤波器并被转换为一个 16 位值。1 位测试信号将经历与来自调制器的常规 1 位脉冲相同的数字

转换，因此任何自测试命令的转换时间与对应的常规 ADC 转换命令完全相同。16 位 ADC 值与常规 ADC 转换命令存储在相同的寄存器组之中。测试信号设计用于在寄存器中放置交替的 0-1 图形。

ADC 清除命令

MT9805 具有 3 个清除命令—CLRCELL、CLRAUX 和 CLRSTAT。这些命令用于清除存储了所有 ADC 转换结果的寄存器。

CLRCELL 命令负责清除电池电压寄存器组 A、B、C、D、E 和 F。这些寄存器中的所有字节均由 CLRCELL 命令设定为 0xFF。

CLRAUX 命令负责清除辅助寄存器组 A、B、C 和 D，除了辅助寄存器组 D 的最后 4 位。这些寄存器中的所有字节皆由 CLRAUX 命令设定为 0xFF。

表 10 自测试命令转换结果

命令	自测试选项	不同 ADC 模式中的输出结果		寄存器
		14KHz、27KHz	7KHz、3KHz、2KHz、1KHz、422Hz、26Hz	
CVST	ST[1:0]=10	0x9554	0x9555	CVA~
	ST[1:0]=01	0x6AAC	0x6AAA	CVD
AXST	ST[1:0]=10	0x9554	0x9555	AUXA
	ST[1:0]=01	0x6AAC	0x6AAA	AUXB
STATST	ST[1:0]=10	0x9554	0x9555	STATA
	ST[1:0]=01	0x6AAC	0x6AAA	STATB

CLRSTAT 命令用于清除状态寄存器组 A 和 B(状态寄存器 B 中的 REVCODE 除外)。REVCODE 的回读将回复器件的版本代码。RSVD 位始终回读 0。状态寄存器组 B 中的所有 OV 标记、UV 标记、MUXFAIL 位、RSVD 位和 THSD 位都由 CLRSTAT 命令设定为 1。THSD 位在 RDSTATB 命令之后被设定为 0。存储了 SOC、ITMP、VA 和 VD 的寄存器全部由 CLRSTAT 命令设定为 0xFF。

导线开路检查(ADOW 命令)

ADOW 命令用于检查 MT9805 中的 ADC 与外部电池之间的任何导线开路。该命令和 ADCV 命令一样在 C 引脚输入端上执行 ADC 转换，仅有的区别是两个内部电流源在其被测量的过程中吸收或供应电流至两个 C 引脚。

ADOW 命令的上拉(PUP)位负责确定电流源是吸收还是供应 100uA 电流。

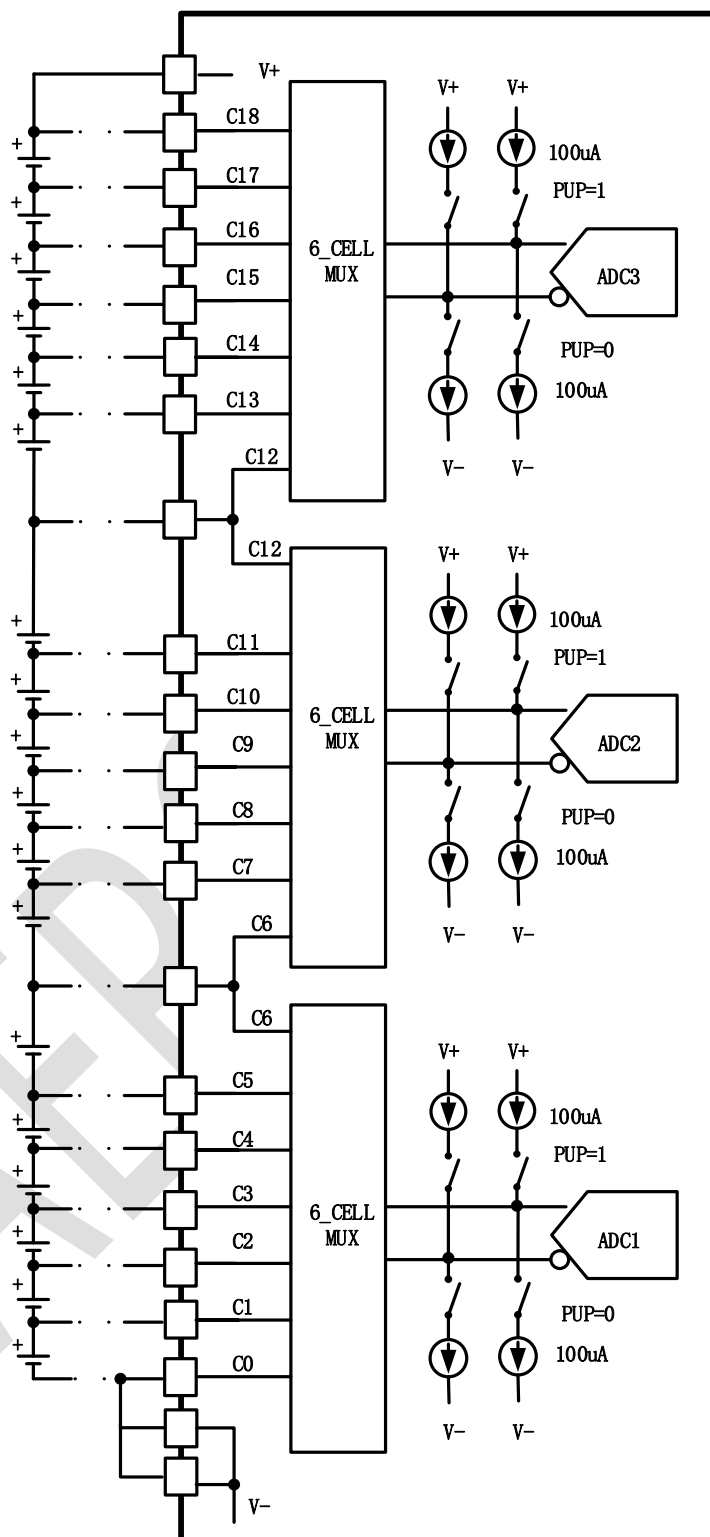


图 22 导线开路检测电路

热停机

为防止 MT9805 发生过热，该 IC 内置了一个热停机电路。如果在芯片上检测到的温度超过了大约 150°C，则热停机电路跳变并使配置寄存器组复位至其默认状态。这将切断所有的放电开关。当出现热停机事件时，状态寄存器组 B 中的 THSD 位将变至高电平。当在状态寄存器组 B 上完成了一个读取操作(RDSTATB 命令)之后，该位即被清除。CLRSTAT 命令将 THSD 位设定为高电平以进行诊断，但并不使配置寄存器组复位。

版本代码

辅助寄存器组 D 包含一个 16 位芯片代码 0x9805。在所有的场合中，当计算数据读取的包错误检测码时必须使用所有位的数值。

看门狗和软件放电定时器

当超过 2 秒钟的时间没有唤醒信号时，看门狗定时器计时终止如图 23 所示。在所有的场合中这都将使配置寄存器 A 字节 CFGR0-CFGR3 复位。当软件定时器停用时，由看门狗定时器使 CFGRA4,CFGRA5,CFGRB0 和 CFGRB1 复位。当看门狗定时器计时周期结束后，WDT 引脚由外部上拉电阻拉至高电平。看门狗定时器始终处于使能状态，并在每一个有效命令执行之后复位。软件放电定时器用于在可编程的持续时间里将放电开关保持于接通状态。如果软件定时器处于使用之中，那么当看门狗定时器启动时放电开关不会被关断。

如欲启用软件定时器，则需把 DTEN 引脚连接至 VREG 的高电平。放电开关可在编程持续时间(其由写至配置寄存器的 DCTO 值确定)里保持导通状态。表 12 给出了各种不同的时间设定值和对应的 DCTO 值。表 13 汇总了在一个看门狗定时器或软件定时器事件之后配置寄存器组的状态。

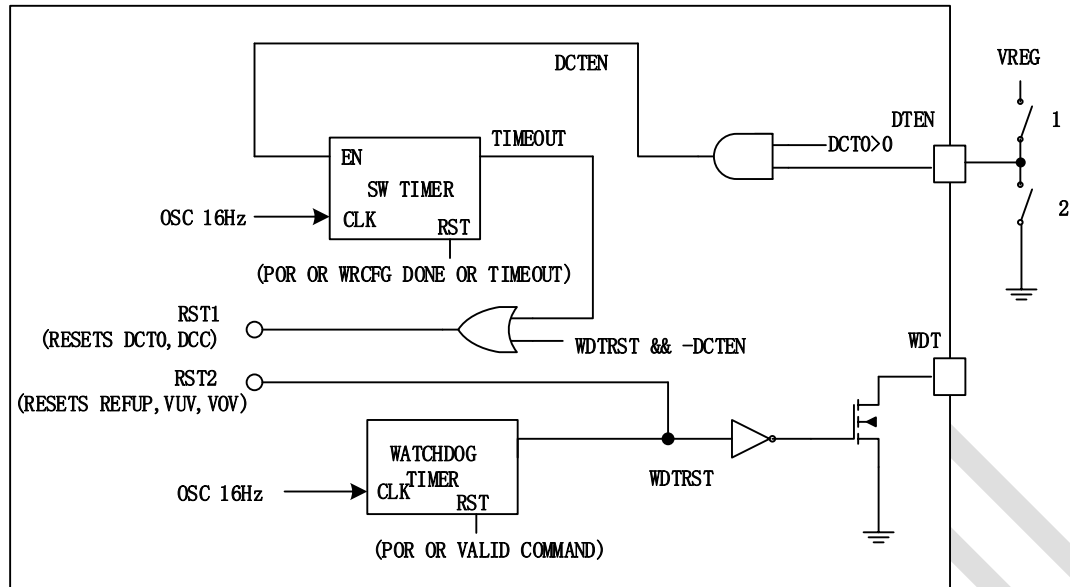


图 23 看门狗和软件放电定时器

与看门狗定时器不同，当存在一个有效命令时，软件定时器不会被复位。软件定时器只能在一个有效的 WRCFG(写配置寄存器)命令之后才能被复位。存在这样一种可能性，即软件定时器将在某些命令的执行过程之中计时终止。

表 12 DCTO 设定值

DCTO	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
分钟	停用	0.5	1	2	3	4	5	10	15	20	30	40	60	75	90	120

表 13 看门狗超时配置寄存器状态

	看门狗定时器	软件定时器
SWREN=0,DCTO=XXXX	复位 CFGRA0-5, CFGRB0-1 和 SCTRL	停用
SWREN=1,DCTO=0000	复位 CFGRA0-5, CFGRB0-1 和 SCTRL	停用
SWREN=1,DCTO!=0000	复位 CFGRA0-3, CFGRB 的 GPIO 位	复位 CFGRA4-5,SCTRL 和 CFGRB 剩余位

如果软件定时器在 WRCFGA 命令的执行过程当中启动，则配置寄存器将按表 14 所述复位。然而，在有效 WRCFG 命令结束时，新的数据将被复制到配置寄存器。当软件定时器启动时新的数据不会丢失。

表 14 软件定时时间

DCTO	剩余时间（分钟）
0	停用
1	$0 < \text{定时} \leq 0.5$
2	$0.5 < \text{定时} \leq 1$
3	$1 < \text{定时} \leq 2$
4	$2 < \text{定时} \leq 3$
5	$3 < \text{定时} \leq 4$
6	$4 < \text{定时} \leq 5$
7	$5 < \text{定时} \leq 6$
8	$6 < \text{定时} \leq 10$
9	$10 < \text{定时} \leq 20$
A	$20 < \text{定时} \leq 30$
B	$30 < \text{定时} \leq 40$
C	$40 < \text{定时} \leq 60$
D	$60 < \text{定时} \leq 75$
E	$75 < \text{定时} \leq 90$
F	$90 < \text{定时} \leq 120$

如果软件定时器在 RDCFG 命令的执行过程当中启动，则配置寄存器组将按表 14 所述复位。因此，从字节 CFGRA4、CFGRA5，CFGRB0-1 读回的数据有可能遭到损坏。

用于电池平衡的 S 引脚脉宽调制

对于电池放电的额外控制，主机可配置 S 引脚，以使用脉冲宽度模块进行操作。当看门狗定时器未超时时，配置寄存器组中的 DCC 位控制 S 引脚。当看门狗定时器超时时，在 PWM 操作期间，DCC 位必须设置为 1，PWM 功能才能工作。一旦 PWM 操作开始，PWM 寄存器中的配置使所有 S 端口周期性开启或关断，以实现表 15 所示的所需占空比。每个 PWM 信号以 30 秒周期运行。对于每个循环，占空比可以以 $1/15=6.67\%$ （2 秒）的增量从 0% 编程到 100%。

表 15 S 端口 PWM 占空比

DCC BIT	PWMn[3:0]	开启时间（s）	关断时间（s）	占空比
---------	-----------	---------	---------	-----

	(n=1,2...12)			
0	4'bXXXX	0	连续关断	0%
1	4'b1111	连续开启	0	100%
1	4'b1110	28	2	93.3%
1	4'b1101	26	4	86.7%
1	4'b1100	24	6	80%
1	4'b1011	22	8	73.3%
1	4'b1010	20	10	66.7%
1	4'b1001	18	12	60%
1	4'b1000	16	14	53.3%
1	4'b0111	14	16	46.7%
1	4'b0110	12	18	40%
1	4'b0101	10	20	33.3%
1	4'b0100	8	22	26.7%
1	4'b0011	6	24	20%
1	4'b0010	4	26	13.3%
1	4'b0001	2	28	6.7%
1	4'b0000	0	连续关断	0%

采用 GPIO 的 I2C/SPI 主控器

MT9805 的 I/O 端口 GPIO3、GPIO4 和 GPIO5 可用作一个 I2C 或 SPI 主控器端口，可与 I2C 或 SPI 受控器件通信。就 I2C 主控器而言，GPIO4 和 GPIO5 分别形成了 I2C 接口的 SDA 和 SCL 端口。至于 SPI 主控器，GPIO3、GPIO4 和 GPIO5 分别成为了 SPI 接口的芯片选择(CSBM)、时钟(SCKM)和数据(SDIOM)端口。GPIO 是漏极开路输出，因此需要在这些端口上布设一个外部上拉电阻器来充当 I2C 或 SPI 主控器。还有一个重要之处是应在 CFG 寄存器组中将 GPIO 位写为“1”，这样一来这些端口就不会被器件在内部拉至低电平。

COMM 寄存器

MT9805 具有一个如表 16 所示的 6 字节 COMM 寄存器。该寄存器存储了至一个受控器的 I2C 或 SPI 通信所需的全部数据和控制位。COMM 寄存器包含了即将发送至从属器件或从从属器件接收的 3 个 $D_n[7:0]$ 数据字节。 $ICOM_n[3:0]$ 规定了发送/接收该数据字节之前的控制动作。 $FCOM_n[3:0]$ 规定了发送/接收该数据字节之后的控制动作。

如果 COMM 寄存器中的 ICOMn[3:0]位被设定为 0，则器件变为一个 I2C 主控器；而倘若该位被设定为 1，那么器件将变为一个 SPI 主控器。表 17 描述了用于 ICOMn[3:0]和 FCOMn[3:0]的有效写代码，以及它们在器件被用作一个 I2C 主控器时的运行方式。表 18 描述了用于 ICOMn[3:0]和 FCOMn[3:0]的有效代码，以及它们在器件被用作一个 SPI 主控器时的运行方式。

表 16 COMM 寄存器内存映射

寄存器	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
COMM0	RD/WR	ICOM0[3:0]				D0[7:4]			
COMM1	RD/WR	D0[3:0]				FCOM0[3:0]			
COMM2	RD/WR	ICOM1[3:0]				D1[7:4]			
COMM3	RD/WR	D1[3:0]				FCOM1[3:0]			
COMM4	RD/WR	ICOM2[3:0]				D2[7:4]			
COMM5	RD/WR	D2[3:0]				FCOM2[3:0]			

请注意，只有表 17 和 18 列出的代码对于 ICOMn[3:0]和 FCOMn[3:0]是有效的。如果把未列于表 17 和 18 的任何其他代码写至 ICOMn[3:0]和 FCOMn[3:0]，则有可能导致在 I2C 和 SPI 端口上出现意外的运行方式。

表 17 用于 ICOMn[3:0]和 FCOMn[3:0]的写代码(在 I2C 主控器上)

控制位	代码	动作	描述
ICOMn[3:0]	0110	启动 (START)	在 I2C 端口上生成一个 START 信号，随后进行数据传输
	0001	停止 (STOP)	在 I2C 端口上生成一个 STOP 信号
	0000	空白 (BLANK)	直接在 I2C 端口上继续进行数据传输
	0111	无传输 (No Transmit)	释放 SDA 和 SCL 并忽略其余的数据
FCOMn[3:0]	0000	主控器确认 (ACK)	主控器在第 9 个时钟周期上生成一个肯定确认(ACK)信号
	1000	主控器否认 (NACK)	主控器在第 9 个时钟周期上生成一个否定确认(NACK)信号
	1001	主控器否认+停止	主控器生成一个否定确认(NACK)信号，后随停止(STOP)信号

表 18 用于 ICOMn[3:0]和 FCOMn[3:0]的写代码(在 SPI 主控器上)

控制位	代码	动作	描述
-----	----	----	----

ICOMn[3:0]	1000	CSBM 为低电平	在 SPI 端口(GPIO3)上生成一个 CSBM 低电平信号
	1001	CSBM 为高电平	在 SPI 端口(GPIO3)上生成一个 CSBM 高电平信号
	1111	无传输(No Transmit)	释放 SPI 端口并忽略其余的数据
FCOMn[3:0]	X000	CSBM 为低电平	在字节传输结束时将 CSBM 保持在低电平
	1001	CSBM 为高电平	在字节传输结束时将 CSBM 转换至高电平

COMM 命令

三个命令可帮助实现至从属器件的 I2C 或 SPI 通信：

WRCOMM、STCOMM、RDCOMM。

WRCOMM 该命令用于将数据写到 COMM 寄存器。此命令可将 6 个数据字节写至 COMM 寄存器。PEC 必需在数据的末端写入。如果 PEC 不匹配，则 COMM 寄存器中的所有数据在 CSB 变至高电平时被清除为“1”。有关写命令格式的更多详情请见“总线协议”部分。

STCOMM 该命令负责在 GPIO 端口上启动 I2C/SPI 通信。COMM 寄存器包含 3 个即将发送至受控器的数据字节。在该命令的执行期间，存储在 COMM 寄存器中的数据字节被传输至从属的 I2C 或 SPI 器件，而从 I2C 或 SPI 器件接收的数据则被存储在 COMM 寄存器中。对于 I2C 通信，该命令采用 GPIO4(SDA)和 GPIO5(SCL)，而对于 SPI 通信，该命令则采用 GPIO3(CSBM)、GPIO4(SDIOM)和 GPIO5(SCKM)。

STCOMM 命令之后将跟随 24 个时钟周期，以在把 CSB 保持于低电平的情况下将每个数据字节传输至从属器件。例如：如欲把 3 个数据字节传输至受控器，则发送 STCOMM 命令及其 PEC，之后是 72 个时钟周期。在 STCOMM 命令的 72 个时钟周期的末端将 CSB 拉至高电平。

在 I2C 或 SPI 通信期间，从从属器件接收的数据在 COMM 寄存器中进行更新。

RDCOMM 从从属器件接收的数据可采用 RDCOMM 命令从 COMM 寄存器读回。该命令读回 6 个后随 PEC 的数据字节。

表 19 描述了当把器件用作一个 I2C 主控器时 ICOMn[3:0]和 FCOMn[3:0]的可用回读代码。Dn[3:0]包含由 I2C 主控器发送的数据字节或者从 I2C 受控器接收的数据字节。在器件被用作 SPI 主控器的场合，用于 ICOMn[3:0]和 FCOMn[3:0]的回读代码始终分别为 0111 和 1111。Dn[3:0]包含由 SPI 主控器发送的数据字节或者从 SPI 受控器接收的数据字节。

采用这些命令可将任何数目的字节以 3 个字节为一组发送至受控器。GPIO 端口将不会在不同的 STCOMM 命令之间被复位。不过，假如命令之间的等待时间超过了 2 秒，则看门狗将超时并使端口复位至默认值。

表 19 用于 ICOMn[3:0]和 FCOMn[3:0]的读代码(在 I2C 主控器上)

控制位	代码	描述
ICOMn[3:0]	0110	主控器生成一个启动(START)信号
	0001	主控器生成一个停止(STOP)信号
	0000	消隐(BLANK)、SDA 在字节之间保持于低电平
	0111	消隐(BLANK)、SDA 在字节之间保持于高电平
FCOMn[3:0]	0000	主控器生成一个肯定确认(ACK)信号
	0111	受控器生成一个肯定确认(ACK)信号
	1111	受控器生成一个否定确认(NACK)信号
	0001	受控器生成一个肯定确认(ACK)信号，主控器
	1001	生成一个停止(STOP)信号

如欲采用一个 I2C 主控器来传输若干个数据字节，则在整个数据流的起点只需要一个 START 信号。在数据流的末端则仅需一个 STOP 信号。所有的中间数据组皆可在数据字节之前使用一个 BLANK 代码，并在数据字节之后酌情采用一个 ACK/NACK 信号。SDA 和 SCL 将不会在不同的 STCOMM 命令之间被复位。

如欲采用一个 SPI 主控器来传输若干个数据字节，则在第一个数据字节的起点发送一个 CSBM 低电平信号。对于在 FCOMn[3:0]上使用适合代码的中间数据组，CSBM 可以保持低电平或被拉至高电平。在最后一个数据字节的末端发送一个 CSBM 高电平信号。CSBM、SDIOM 和 SCKM 将不会在不同的 STCOMM 命令之间被复位。

使用 S 端口控制寄存器发送脉冲

MT9805 的 S 引脚可用作简单的串行接口。用于控制外部主动均衡 DC/DC 转换器。S 端口控制寄存器 SCTRLn[3:0] (n=1,2...18) 控制 S 端口发送 1 到 7 个脉冲之间的脉冲序列。表示 S 端口由 SCTRLn[3:0] (n=1,2...18) 详细说明。S 引脚脉冲频率为 6.44kHz (155μS 周期)，脉冲宽度将为 77.6μs。如果命令 PEC 匹配，则在最后一个命令 PEC 时钟之后发送 STCTRL 命令时，S 引脚脉冲开始。然后，主机可以继续对 SCLK 进行计时，以便轮询脉冲的状态。此轮询功能类似于 ADC 轮询功能。数据输出将保持逻辑低，直到 S 引脚脉冲序列完成。当 S 引脚脉冲正在进行时，将忽略新的 STSCTRL 或 WRSCTRL 命令。如果正确接收 WRSCTRL 命令和命令 PEC，但数据 PEC 不匹配，则 S 控制寄存器组将被清除。如果配置寄存器组中的 DCC 位被设置为 1，MT9805 将驱动所选的 S 引脚为低电平，而不考虑 S 控制寄存器组。当使用 S 控制寄存器组时，主机应将 DCC 位设置为 0。CLRSCTRL 命令可用于将 S 控制寄存器组快速重置为所有 0，并强制脉冲发生器释放对 S 引脚的控制。

S 引脚静默

可以通过发送静默命令 MUTE 禁用 S 引脚，通过发送取消静默命令 UNMUTE 重新启用 S 引脚。静默和非静默命令不需要任何后续数据，因此命令通过 MT9805 设备的堆栈快速传播。此操作允许主机在不干扰寄存器内容的情况下快速 ($<100\mu\text{s}$) 禁用和重新启用放电。静默状态报告在配置寄存器组 B 中的只读 mute 位中。

串行接口概述

在 MT9805 上有两类串行端口，即：一个标准的四线式串行外设接口(SPI)和一个两线式隔离型接口(隔离 SPI)。根据 ISOMD 引脚的状态，可以把引脚 53, 54, 61, 62 配置为两线式或四线式串行端口，第二个 ISOSPI 使用引脚 57, 58, 63, 64。

四线式串行外设接口(SPI)物理层外部连接

把 ISOMD 连接至 V-可针对四线式 SPI 来配置串行端口 A。SDO 引脚是一个漏极开路输出，其需要一个连接至合适电源电压的上拉电阻器。

定时

四线式串行端口配置为在一个采用 CPHA=1 和 CPOL=1 的 SPI 系统中运作。因此，SDI 上的数据在 SCLK 的上升沿期间必须保持稳定。时序规格见图 24。最大数据速率 1Mbps。

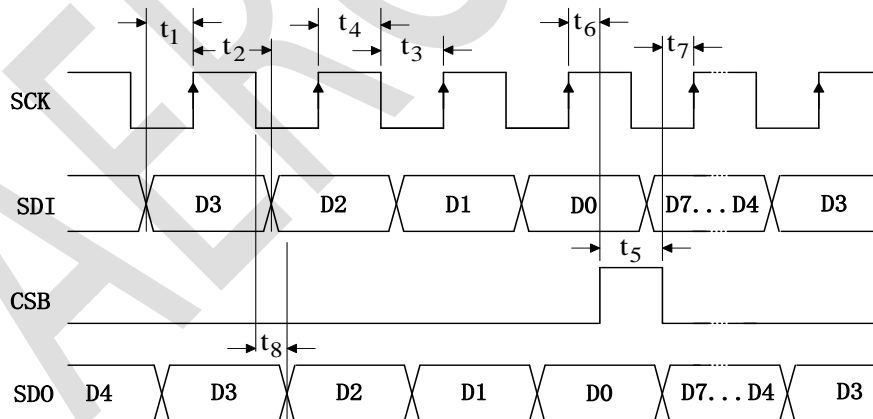


图 24 四线式串行外设接口的时序图

两线式隔离型接口(隔离 SPI)物理层

两线式接口提供了一种采用简单的双绞线电缆来实现 MT9805 器件互连的方法。该接口专门针对低包差错率(此时电缆会遭受高 RF 磁场的影响)而设计。隔离通过一个外部变压器实现。

标准的 SPI 信号被编码为差分脉冲。发送脉冲的强度和接收器的门限电平由两个外部电阻器设定。用户可通过

调整电阻器的阻值在功率耗散与抗噪声性能之间进行权衡。

图 25 示出了隔离 SPI 电路的工作原理。一个 2V 基准负责驱 IBIAS 引脚。外部电阻器 R_{B1} 和 R_{B2} 产生基准电流 I_B 。该电流用于设定发送器的驱动强度。另外, R_{B1} 和 R_{B2} 还在 ICMP 引脚上形成了 2V 基准的一个分压器。这设定了接收器电路的门限电压。

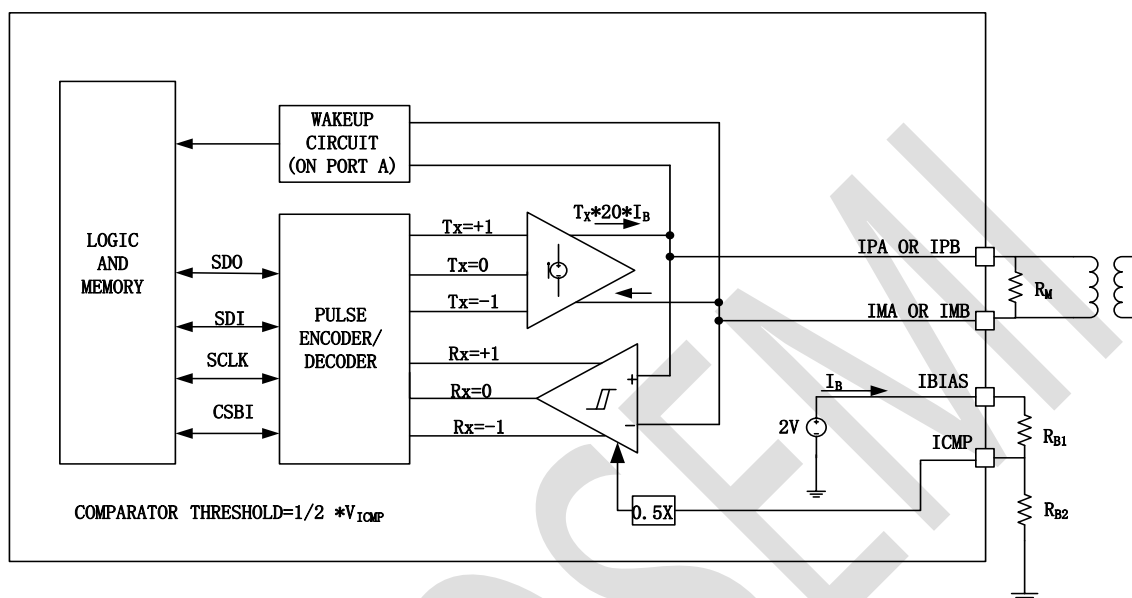


图 25 隔离 SPI 接口

外部连接

MT9805 具有两个被称为端口 B 和端口 A 的串行端口。端口 B 始终被配置为一个两线式接口。端口 A 可以是一个两线式或四线式接口，这取决于 ISOMD 引脚的连接。

选择偏置电阻器

可调的信号幅度允许系统通过牺牲功耗指针来换取通信坚固性的提高，而可调节的比较器门限则使系统能够对信号损失实施补偿。

隔离 SPI 发送器驱动电流和比较器电压门限由一个位于 IBIAS 和 V^- 之间的电阻分压器($R_{BIAS}=R_{B1}+R_{B2}$)设定。经分压后的电压连接至 ICMP 引脚，其将比较器门限设定为该电压(V_{ICMP})的 1/2。当任一隔离 SPI 接口被使能(非 IDLE)时，IBIAS 保持于 2V，从而致使从 IBIAS 引脚流出一个电流 I_B 。IP 和 IM 引脚驱动电流为 $25 \cdot I_B$ 。

例如，如果分压电阻器 R_{B1} 为 2.8k，且电阻器 R_{B2} 为 1.21k(这样 $R_{BIAS}=4k$)，则：

$$I_B = \frac{2V}{R_{B1} + R_{B2}} = 0.5mA$$

$$I_{DRV} = I_{IP} = I_{IM} = 25 \times I_B = 12.5mA$$

$$V_{ICMP} = 2V \times \frac{R_{B2}}{R_{B1} + R_{B2}} = I_B \times R_{B2} = 603\text{mV}$$

$$V_{TCMP} = 0.5 \times V_{ICMP} = 302\text{mV}$$

在该例中，脉冲驱动电流 I_{DRV} 将为 12.5mA，而接收器比较器将检测 IP-IM 幅度大于 $\pm 302\text{mV}$ 的脉冲。

假如隔离势垒采用由双绞线连接并在每一端终接了 120Ω 电阻器的 1:1 变压器，则传输差分信号幅度(\pm)将为：

$$V_A = I_{DRV} \times \frac{R_M}{2} = 0.75\text{V}$$

隔离 SPI 脉冲细节

两个 MT9805 器件能够通过一个隔离势垒来回地发送和接收差分脉冲以实现通信。发送器可输出三种电压电平： $+V_A$ 、 0V 和 $-V_A$ 。负载电阻器两端的 IP 供电电流和 IM 吸收电流产生一个正输出。IP 吸收电流和 IM 供电电流则产生一个负电压。当两个输出均关断时，负载电阻将强制差分输出为 0V 。

如欲启用软件定时器，则需把 SWTEN 引脚连接至 VREG 的高电平。现在，放电开关可在编程持续时间(其由写至配置寄存器的 DCTO 值确定)里保持导通状态。表 20 隔离 SPI 脉冲类型

脉冲类型	第一种电平	第二种电平	终止电平
长+1	$+V_A(150\text{ns})$	$-V_A(150\text{ns})$	0V
长-1	$-V_A(150\text{ns})$	$+V_A(150\text{ns})$	0V
短+1	$+V_A(50\text{ns})$	$-V_A(50\text{ns})$	0V
短-1	$-V_A(50\text{ns})$	$+V_A(50\text{ns})$	0V

为了使用该两线式接口，主机控制器不必生成隔离 SPI 脉冲。系统中的第一个 MT9805 可采用其端口 A 上的四线 SPI 接口与微控制器进行通信，然后使用其端口 B 上的两线式隔离 SPI 接口菊链连接至其他的 MT9805。或者，也可以使用一个 MT9820 将 SPI 信号转换为隔离 SPI 脉冲。

表 21 MT9805 端口 B(主控器)隔离 SPI 端口功能

通信事件（端口 A SPI）	传输脉冲（端口 B 隔离 SPI）
CSBI 上升	长+1
CSBI 下降	长-1
SCLK 上升沿, SDI=1	短+1
SCLK 上升沿, SDI=0	短-1

端口 A 针对 SPI 配置时 MT9805 的操作

当 MT9805 在端口 A 作为一个 SPI(ISOMD=V-)的情况下运作时，SPI 将检测 4 种通信事件之一：CSBI 下降、

CSBI 上升、SCLK 上升(SDI=0)和 SCLK 上升(SDI=1)。每种事件被转换为 4 种脉冲之一，以通过 MT9805 菊链进行传输。长脉冲用于传输 CSBI 变化，而短脉冲则用于传输数据，如表 22 中所示。

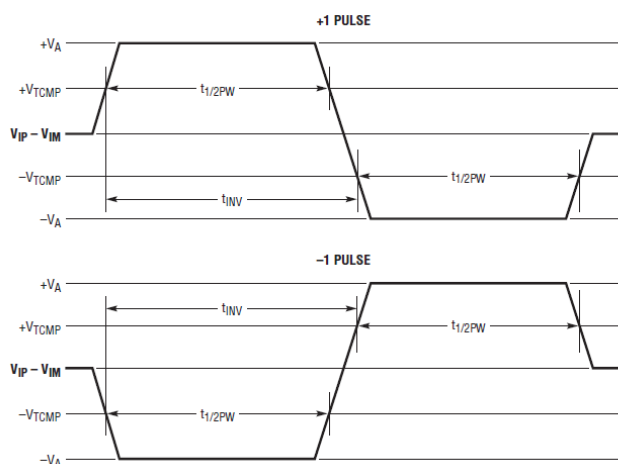


图 16 隔离 SPI 脉冲细节

在隔离势垒的另一侧(即：在电缆的另一端)，第二个 MT9805 设置 ISOMD=VREG。其端口 A 或者端口 B 是从属隔离 SPI 接口的作用。它接收每一个传输脉冲并在内部重构 SPI 信号，如表 22 所示。此外，在一个 READ 命令期间该端口还可以传输返回数据脉冲。

表 22 数据脉冲说明

接受脉冲（端口 A 隔离 SPI）	内部 SPI 端口动作	返回脉冲
长+1	驱动 CSBI 至高电平	无
长-1	驱动 CSBI 至低电平	无
短+1	设定 SDI=1	短-1 脉冲 (若读取位为 0)
	SCLK 脉冲	
短-1	设定 SDI=0	若不处于读取模式则不返回脉冲，当读取位为 1 时，返回短+1 脉冲，
	SCLK 脉冲	

较低的隔离 SPI 端口(端口 A)从不传输长(CSBI)脉冲。此外，从属隔离 SPI 端口将只传输短-1 脉冲，而绝不会传输一个+1 脉冲。主控器端口将空响应认作一个逻辑“1”。这使得可在单根电缆上连接多个从属器件而不会有发生冲突的风险(多分支)。

当 MT9805 在端口 A 配置为 isoSPI 的情况下运行时，可以从端口 A 或端口 B 启动通信，MT9805 可以根据通信方向将端口 A 或 B 配置为从属或主端口。可逆 isoSPI 功能允许在菊花链设备堆栈中从两个方向进行通信，实现双向闭环通信。

当 MT9805 处于睡眠状态时，设备对端口 A 或端口 B 上的有效唤醒信号做出响应。对于 ISOMD 引脚的任何一种配置都是如此。

当 MT9805 处于就绪状态时，可以通过在端口 A 或端口 B 上发送一个长的-1 isoSPI 脉冲（CSB 下降）来启动通信。MT9805 自动将接收长的-1 isoSPI 脉冲的端口配置为从端口，将另一个端口配置为主端口。isoSPI 脉冲通过主端口传输到菊花链中的其余设备。

在激活状态下，MT9805 处于通信中间，内部 SPI 端口的 CSB 较低。在通信结束时，从端口上的长+1 脉冲（CSB 上升）将设备返回到就绪状态。尽管它不是正常通信例程的一部分，但 MT9805 允许端口 A 和端口 B 在活动状态下交换。该特征对于主控制器来说是有用的，以恢复对 MT9805 的从端口的控制，而与端口的当前状态无关。此操作可以通过在设备传输的最后一个 isoSPI 信号延迟 t_{BLOCK} 后，在主端口上发送一个长的-1 isoSPI 脉冲来完成。发送到 t_{BLOCK} 内部主端口的任何长 isoSPI 脉冲都会被设备拒绝。

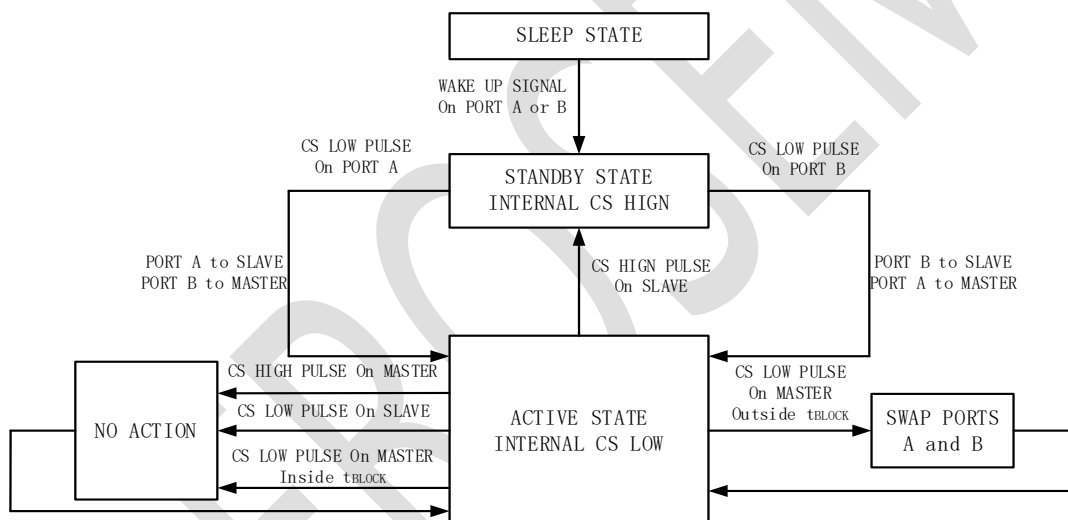


图 26 双向 ISOSPI 状态

图 27 示出了一个发送至菊花链式 MT9805 器件的 READ 命令的隔离 SPI 时序图。ISOMD 引脚连接至底端器件上的 V₋，因此其端口 A 被配置为一个 SPI 端口(CSBI、SCLK、SDI 和 SDO)。图中示出了三个堆栈器件的隔离 SPI 信号，各标注了端口(A 或 B)和器件型号。请注意，ISO B1 和 ISO A2 实际上是相同的信号，但示于连接器件 1 和 2 的传输电缆的每一端上。同样，ISO B2 和 ISO A3 也是相同的信号，但具有示于器件 2 和 3 之间的电缆延迟。

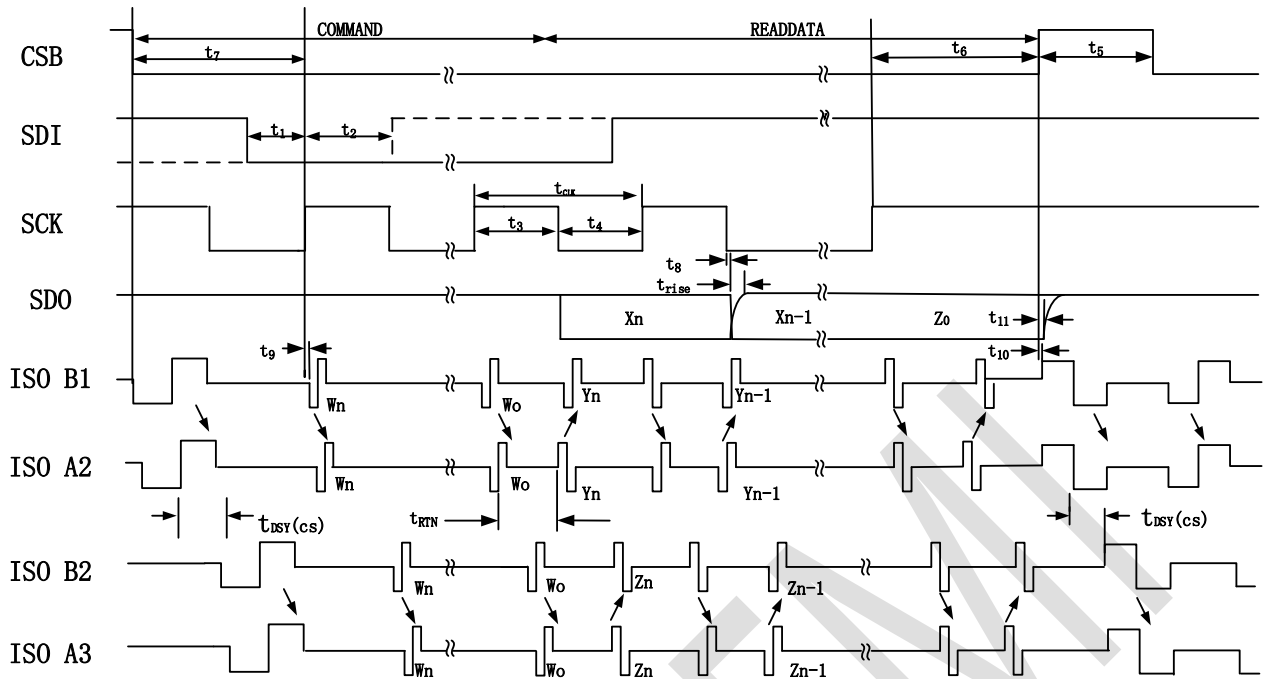


图 27 隔离 SPI 时序图

W_n 至 W_0 位指的是 READ 命令的 16 位命令代码和 16 位 PEC。在 W_0 位的末端，3 个器件对 READ 命令进行译码，并移出在时钟 SCLK 的下一个上升沿上有效的数据。 X_n 至 X_0 位指的是由器件 1 移出的数据。 Y_n 至 Y_0 位指的是由器件 2 移出的数据， Z_n 至 Z_0 位指的是由器件 3 移出的数据。所有的数据均以一种菊花链的方式从器件 1 上的 SDO 端口读回。

唤醒串行接口

如果在端口 A 或者端口 B 上持续 tIDLE 的时间没有动作，则串行接口(SPI 或隔离 SPI)将进入低功率 IDLE 状态。WAKEUP 电路负责监视引脚 61 和 64 上的动作。

假如 ISOMD=V₋，则端口 A 处于 SPI 模式。CSBI 或 SCLK 引脚上的动作将唤醒 SPI 接口。如果 ISOMD=VREG，则端口 A 处于隔离 SPI 模式。IPA-IMB 上的差分动作将唤醒隔离 SPI 接口。当隔离 SPI 状态在 tWAKE 或 tREADY 之内变至 READY 时，MT9805 将做好通信的准备。

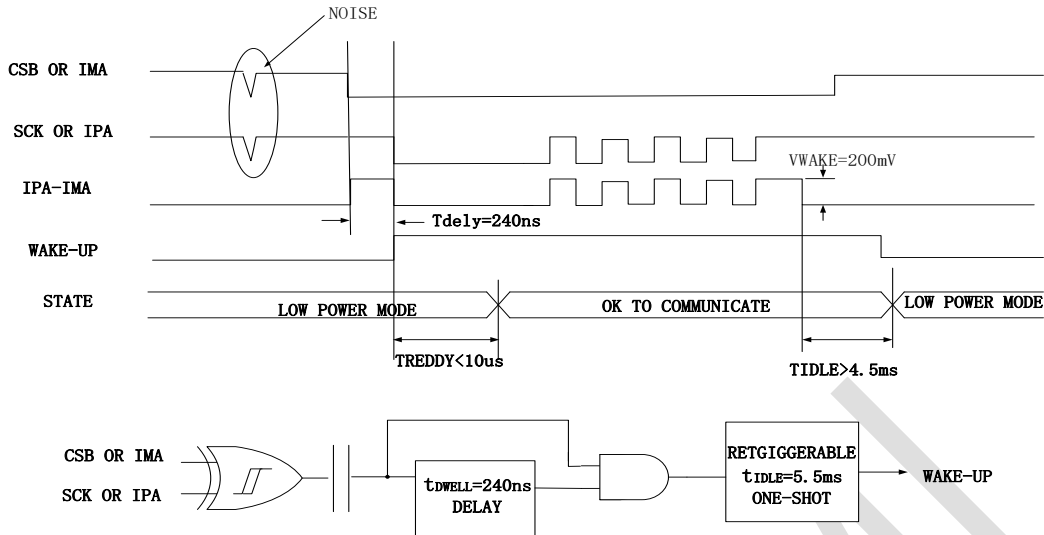


图 28 唤醒检测和 IDLE 定时器

MT9805 在做好了通信准备之后于端口 B 上发送一个长+1 脉冲。在菊链式配置中，该脉冲唤醒堆栈中的下一个器件，这接着将唤醒再下一个器件。如果堆栈中有“N”个器件，则所有器件将在 $N \cdot t_{WAKE}$ 或 $N \cdot t_{READY}$ 的时间之内(取决于内核状态)上电。对于大的堆栈，时间 $N \cdot t_{WAKE}$ 有可能等于或大于 t_{IDLE} 。在该场合中，在等待的时间超过了 $N \cdot t_{WAKE}$ 之后，主机可能发送另一个虚字节并等待 $N \cdot t_{READY}$ 的时间，以确保所有的器件均处于 READY 状态。图 28 示出了定时和功能上等效的电路。共模信号将不会唤醒串行接口。该接口专为在接收一个大信号单端脉冲或一个低幅度对称脉冲之后唤醒而设计。差分信号 $|SCLK(IPA) - CSBI(IMA)|$ 在 $t_{DWELL} = 240ns$ 的最小持续时间里必须至少为 $V_{WAKE} = 200mV$ ，才会被认作是一个适合对串行接口进行上电的唤醒信号。

数据链路层

MT9805 上的所有数据皆以字节组进行传输。每个字节包 8 个位。字节的传输从最高有效位(MSB)开始。CSBI 必须在整个命令序列期间(包括在一个命令字节和后续数据之间)保持低电平。在执行写命令时，数据在 CSBI 的上升沿被锁定。

包误差检测码

包错误检测码(PEC)是一种 15 位循环冗余校验(CRC)值，其针对一个寄存器组中的所有位并按照它们的传递顺序进行计算，计算过程采用了 000000000010000 的初始 PEC 种子值和下面的特征多项式： $x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$ 。为了计算该 15 位 PEC 值，可以建立一种简单的程序：

1. 将 PEC 初始化为 000000000010000(PEC 是一个 15 位寄存器组)
2. 对于每个输入 PEC 寄存器组的 DIN 位，设定：

$IN0 = DIN \text{ XOR } PEC[14]$
 $IN3 = IN0 \text{ XOR } PEC[2]$

$IN4=IN0 \text{ XOR } PEC[3]$

$IN7=IN0 \text{ XOR } PEC[6]$

$IN8=IN0 \text{ XOR } PEC[7]$

$IN10=IN0 \text{ XOR } PEC[9]$

$IN14=IN0 \text{ XOR } PEC[13]$

3.按下面的设置更新 15 位 PEC:

$PEC[14]=IN14,$

$PEC[13]=PEC[12],$

$PEC[12]=PEC[11],$

$PEC[11]=PEC[10],$

$PEC[10]=IN10,$

$PEC[9]=PEC[8],$

$PEC[8]=IN8,$

$PEC[7]=IN7,$

$PEC[6]=PEC[5],$

$PEC[5]=PEC[4],$

$PEC[4]=IN4,$

$PEC[3]=IN3,$

$PEC[2]=PEC[1],$

$PEC[1]=PEC[0],$

$PEC[0]=IN0$

4.返回第 2 步直到所有数据均被转移为止。最终的 PEC(16 位)是 15 位的 PEC 寄存器的值，在其 LSB 上附加了一个值为 0 的位。

表 23 0x0001 的 PEC 计算

PEC[14]	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
PEC[13]	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0
PEC[12]	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	1
PEC[11]	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	1	1
PEC[10]	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	1	1	1

PEC[9]	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	1
PEC[8]	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0
PEC[7]	0	0	0	1	0	0	0	0	0	0	1	1	1	0	1	1	1
PEC[6]	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
PEC[5]	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
PEC[4]	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1
PEC[3]	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0
PEC[2]	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
PEC[1]	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
PEC[0]	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
IN14	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
IN10	0	0	0	0	0	1	0	0	0	0	1	1	0	1	1	1	PEC
IN8	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	
IN7	0	0	1	0	0	0	0	0	0	0	1	1	1	0	1	1	
IN4	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	
IN3	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	
IN0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	
DIN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
周期	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

表 23 列举了计算一个 16 位字(0x0001)的 PEC 的例子。当在 LSB 上填充了一个 0 位之后, 0x0001 的 PEC 计算值为 0x3D6E。对于更长的数据流, PEC 在发送至 PEC 寄存器的最后一位的末端有效。MT9805 可计算任何接收命令或数据的 PEC, 并将之与该命令或数据之后的 PEC 进行比较。只有在 PEC 匹配的情况下, 命令或数据才会被视为有效。另外, MT9805 还在其移出数据的末端附加计算的 PEC。表 24 列出了给 MT9805 写入数据或从 MT9805 读取数据时的 PEC 格式。

表 24 读/写 PEC 格式

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
PEC0	RD/WR	PEC[14:7]							
PEC1	RD/WR	PEC[6:0]							

当给 MT9805 写入任何命令时, 命令字节 CMD0 和 CMD1 以及 PEC 字节 PEC0 和 PEC1 将按下列顺序在端口

A 上发送: CMD0, CMD1, PEC0, PEC1

在菊链式 MT9805 器件接收到一个广播写命令之后, 数据将被发送至每个器件, 随后发送的是 PEC。例如: 当把配置寄存器组写至两个菊链式器件(主器件为 P, 堆栈器件为 S)时, 数据将按如下顺序发送至端口 A 上的主器件:

CFGR0(S), ..., CFGR5(S), PEC0(S), PEC1(S), CFGR0(P), ..., CFGR5(P), PEC0(P), PEC1(P)

在一个用于菊链式器件的读命令之后, 每个器件将在端口 A 上移出其数据和它为其数据计算的 PEC, 然后在端口 B 上移出接收数据。例如, 当从两个菊链式器件(主器件为 P, 堆栈器件为 S)读取状态寄存器组 B 时, 主器件将按以下顺序在端口 A 上发送数据:

STBR0(P), ..., STBR5(P), PEC0(P), PEC1(P), STBR0(S), ..., STBR5(S), PEC0(S), PEC1(S)

广播命令

广播命令是一种总在线的所有器件都将对其做出响应的命令。该见“总线协议”以了解广播命令格式。利用广播命令可同时向所有的器件发送命令。

在并行配置中, 这适用于 ADC 转换器和轮询(polling command)命令。当所有器件都被写入相同的数据时, 其还可与写命令一起使用。广播读命令不应在并行配置中使用。

菊链式配置只支持广播命令。链中的所有器件同时接收命令字节。例如, 为在一个器件堆栈中启动 ADC 转换将发送一个 ADCV 命令, 所有的器件将同时启动转换操作。对于读和写命令, 发送单个命令, 然后堆栈器件实际上变成了一个级联式移位寄存器, 其中的数据通过每个器件移至堆栈中的下一个器件。

轮询(polling command)方法

确定 ADC 转换是否完成的最简单方法, 是控制器启动 ADC 转换并等待规定的转换时间, 然后读取结果。如果使用以 SPI 模式通信的单个 MT9805 (ISOMD 引脚接低电平), 则有两种轮询方法。第一种方法是在发送 ADC 转换命令后将 CSB 保持为低电平。发出转换命令后, 当器件忙于执行转换时, SDO 线被拉到低电平。当器件完成转换时, SDO 被拉高。但是, 即使器件尚未完成转换, 当 CSB 变为高电平时, SDO 也会变为高电平。这种方法的问题在于, 在等待 ADC 转换完成期间, 控制器无法自由执行其他串行通信。

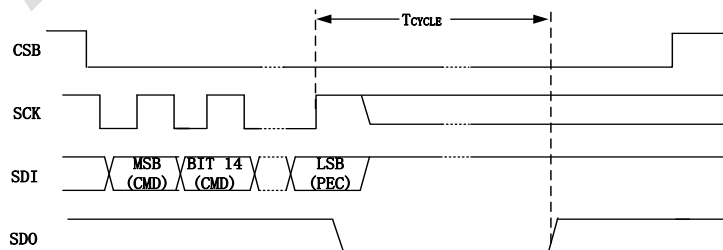


图 29 ADC 转换命令之后的 SDO 轮询 (单个 MT9805)

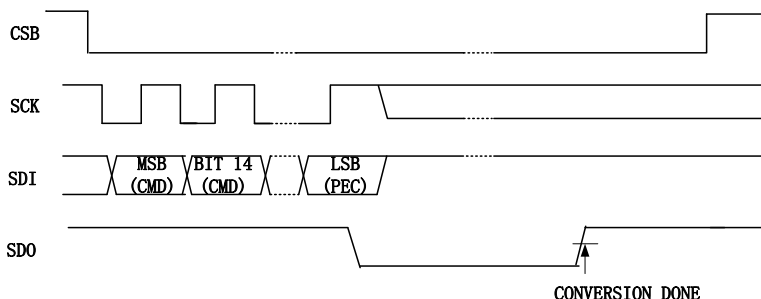


图 30 采用 PLADC 命令的 SDO 轮询（单个 MT9805）

第二种方法克服了此限制。控制器可以发送 ADC 启动命令，执行其他任务，然后发送轮询 ADC 转换器状态 (PLADC) 命令来确定 ADC 转换的状态（参见图 30）。发出 PLADC 命令后，如果器件正忙于执行转换，则 SDO 变为低电平。转换结束时，SDO 被拉高。但是即使器件尚未完成转换，当 CSB 变为高电平时，SDO 也会变为高电平。

如果使用以 isoSPI 模式通信的单个 MT9805，则低端端口只有在响应其收到主机 isoSPI 脉冲时才发送数据脉冲。因此，按照上述任意一种轮询方法发出命令后，isoSPI 数据脉冲都将被发送至该器件以更新转换状态。只需为 SCK 引脚提供时钟，便可使用 MT9820 发送这些脉冲。响应此脉冲，如果 MT9805 仍在忙于执行转换，则发送回一个低电平 isoSPI 脉冲；如果已完成转换，则发送回一个高电平数据脉冲。如果将 CSB 高电平 isoSPI 脉冲发送到器件，则器件退出轮询命令。

在有 N 个堆叠器件的菊花链配置中，同样可以使用这两种轮询方法。如果底部器件以 SPI 模式进行通信，则底部器件的 SDO 会指示整个堆叠的转换状态。也就是说 SDO 会保持低电平，直到堆叠中的所有器件都完成转换为止。在第一种轮询方法中，发送 ADC 转换命令后，通过 SCK 发送时钟脉冲，同时 CSB 保持低电平。SDO 状态仅在 SCK 上的 N 个时钟脉冲结束时才有效。在前 N 个时钟脉冲期间，菊花链中底部 MT9805 输出 0 或低电平数据脉冲。在 N 个时钟脉冲之后，对于随后的每个时钟脉冲，来自底部 MT9805 的输出数据都会更新（参见图 31）。在第二种方法中，发送 PLADC 命令后，通过 SCK 发送时钟脉冲，同时 CSB 保持低电平。与第一种方法类似，SDO 状态仅在 SCK 上的 N 个时钟周期后才有效，并在随后的每个时钟周期之后更新（参见图 32）。

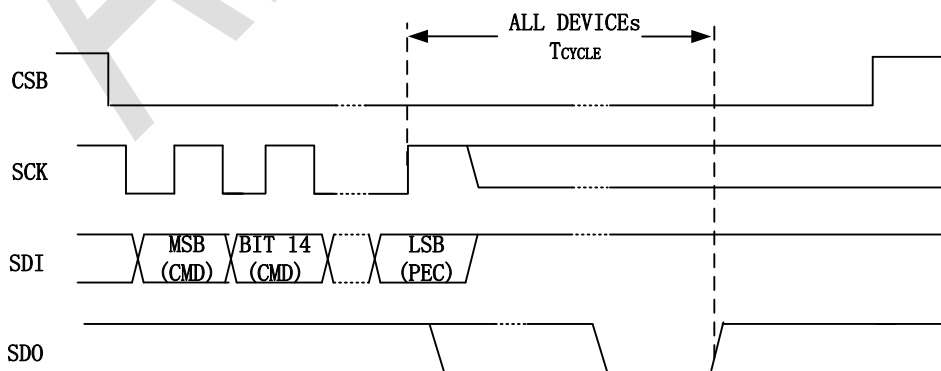


图 31 ADC 转换命令之后的 SDO 轮询（菊花链配置）

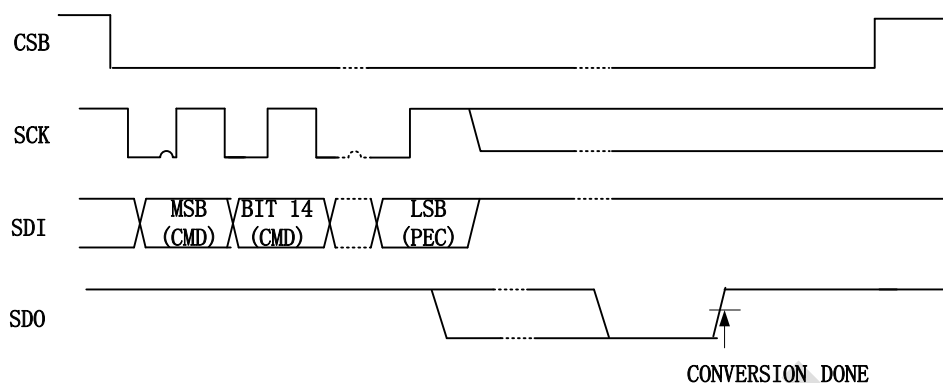


图 32 采用 PLADC 命令的 SDO 轮询（菊花链配置）

如果底部器件以 isoSPI 模式通信，则将 isoSPI 数据脉冲发送到该器件以更新转换状态。采用 MT9820，只需为 SCK 引脚提供时钟即可实现此操作。转换状态仅在底部 MT9805 器件接收到 N 个 isoSPI 数据脉冲之后有效，对于随后的每个 isoSPI 数据脉冲，状态都会更新。如果堆叠中有器件正忙于执行转换，该器件将返回低电平数据脉冲；如果所有器件都空闲，则返回高电平数据脉冲。

总线协议

协议格式：表 25 至 29 给出了广播命令和地址命令的协议格式。

命令格式：广播命令和地址命令的格式分别示于表 30。11 位命令代码 CC[10:0] 对于一个广播命令或一个地址命令是相同的。表 31 给出了所有命令代码的清单。对于 CMD0[7]至 CMD0[3]，广播命令具有一个数值 0。用于广播的 PEC 必须在整个 16 位命令(CMD0 和 CMD1)上计算。表 31 列出了针对 MT9805 的所有命令及其选项。

表 25 广播轮询命令

8	8	8	8	
CMD0	CMD1	PEC0	PEC1	轮询数据

表 26 广播写命令

8	8	8	8	8		8	8	8	8		8
CMD0	CMD1	PEC0	PEC1	数据字节低	...	数据字节高	PEC0	PEC1	移出字节 1	...	移出字节 n

表 27 地址写命令

8	8	8	8	8		8	8	8
CMD0	CMD1	PEC0	PEC1	数据字节低	...	数据字节高	PEC0	PEC1

表 28 广播读命令

8	8	8	8	8		8	8	8	8		8
---	---	---	---	---	--	---	---	---	---	--	---

CMD0	CMD1	PEC0	PEC1	数据 字节 低	。。。	数字 字节 高	PEC0	PEC1	移出 字节 1	。。。	移出 字节 n
------	------	------	------	---------------	-----	---------------	------	------	---------------	-----	---------------

表 29 地址读命令

8	8	8	8	8		8	8	8
CMD0	CMD1	PEC0	PEC1	数据字节低	。。。	数字字节高	PEC0	PEC1

表 30 广播地址命令格式

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CMD0	WR	0	0	0	0	0	CMD[10:8]		
CMD1	WR	CMD[7:0]							

表 31 命令代码

命令描述	名称	CMD[10:0]										
		10	9	8	7	6	5	4	3	2	1	0
写配置寄存器组 A	WRCFGA 0x0001	0	0	0	0	0	0	0	0	0	0	1
写配置寄存器组 B	WRCFGB 0x0024	0	0	0	0	0	1	0	0	1	0	0
读配置寄存器组 A	RDCFGA 0x0002	0	0	0	0	0	0	0	0	0	1	0
读配置寄存器组 B	RDCFGB 0x0026	0	0	0	0	0	1	0	0	1	1	0
读电池电压寄存器组 A	RDCVA 0x0004	0	0	0	0	0	0	0	0	1	0	0
读电池电压寄存器组 B	RDCVB 0x0006	0	0	0	0	0	0	0	0	1	1	0
读电池电压寄存器组 C	RDCVC 0x0008	0	0	0	0	0	0	0	1	0	0	0
读电池电压寄存器组 D	RDCVD	0	0	0	0	0	0	0	1	0	1	0

	0x000A											
读电池电压寄存器组 E	RDCVE 0x0009	0	0	0	0	0	0	0	1	0	0	1
读电池电压寄存器组 F	RDCVF 0x000B	0	0	0	0	0	0	0	1	0	1	1
读辅助寄存器组 A	RDAUXA 0x000C	0	0	0	0	0	0	0	1	1	0	0
读辅助寄存器组 B	RDAUXB 0x000E	0	0	0	0	0	0	0	1	1	1	0
读辅助寄存器组 C	RDAUXC 0x000D	0	0	0	0	0	0	0	1	1	0	1
读辅助寄存器组 D	RDAUXD 0x000F	0	0	0	0	0	0	0	1	1	1	1
读状态寄存器组 A	RDSTATA 0x0010	0	0	0	0	0	0	1	0	0	0	0
读状态寄存器组 B	RDSTATB 0x0012	0	0	0	0	0	0	1	0	0	1	0
写 S 控制寄存器组	WRSCtrl 0x0014	0	0	0	0	0	0	1	0	1	0	0
写 PWM 寄存器组	WRPWM 0x0020	0	0	0	0	0	1	0	0	0	0	0
写 PWM/S 控制寄存器组	WRPSB 0x001C	0	0	0	0	0	0	1	1	1	0	0
读 S 控制寄存器组	RDSCtrl 0x0016	0	0	0	0	0	0	1	0	1	1	0
读 PWM 寄存器组	RDPWM 0x0022	0	0	0	0	0	1	0	0	0	1	0
读 PWM/S 控制寄存器组 B	RDPSB 0x001E	0	0	0	0	0	0	1	1	1	1	0

启动 S 控制脉冲轮询	STCTRL 0x0019	0	0	0	0	0	0	1	1	0	0	1
清除 S 控制寄存器组	CLRCTRL 0x0018	0	0	0	0	0	0	1	1	0	0	0
启动电池电压 ADC 转换和轮询状态	ADCV	0	1	MD[1:0]		1	1	DCP	0	CH[2:0]		
启动导线开路 ADC 转换和轮询状态	ADOW	0	1	MD[1:0]		PUP	1	DCP	1	CH[2:0]		
启动自测试电池电压转换和轮询状态	CVST	0	1	MD[1:0]		ST[1:0]		0	0	1	1	1
启动重叠测量 CELL7 和 CELL13	ADOL	0	1	MD[1:0]		0	0	DCP	0	0	0	1
启动 GPIO ADC 转换和轮询状态	ADAX	1	0	MD[1:0]		1	1	0	0	CHG[2:0]		
启动 GPIO 数字冗余 ADC 转换和轮询状态	ADAXD	1	0	MD[1:0]		0	0	0	0	CHG[2:0]		
启动 GPIO 导线开路 ADC 转换和轮询状态	AXOW	1	0	MD[1:0]		PUP	0	1	0	CHG[2:0]		
启动自测试 GPIO 转换和轮询状态	AXST	1	0	MD[1:0]		ST[1:0]		0	0	1	1	1
启动状态组 ADC 转换和轮询状态	ADSTAT	1	0	MD[1:0]		1	1	0	1	CHST[2:0]		
启动状态组数字冗余 ADC 转换和轮询状态	ADSTATD	1	0	MD[1:0]		0	0	0	1	CHST[2:0]		
启动自测试状态组转换和轮询状态	STATST	1	0	MD[1:0]		ST[1:0]		0	1	1	1	1
启动组合电池电压以及 GPIO1、GPIO2 转换和轮询状态	ADCVAX	1	0	MD[1:0]		1	1	DCP	1	1	1	1

启动组合电池电压及总电压转换和轮询状态	ADCVSC	1	0	MD[1:0]		1	1	DCP	0	1	1	1
清除电池电压寄存器组	CLRCCELL 0x0711	1	1	1	0	0	0	1	0	0	0	1
清除辅助寄存器组	CLRAUX 0x0712	1	1	1	0	0	0	1	0	0	1	0
清除状态寄存器组	CLRSTAT 0x0713	1	1	1	0	0	0	1	0	0	1	1
轮询 ADC 转换状态	PLADC 0x0714	1	1	1	0	0	0	1	0	1	0	0
诊断 MUX 和轮询状态	DIAGN 0x0715	1	1	1	0	0	0	1	0	1	0	1
写 COMM 寄存器组	WRCOMM 0x0721	1	1	1	0	0	1	0	0	0	0	1
读 COMM 寄存器组	RDCOMM 0x0722	1	1	1	0	0	1	0	0	0	1	0
启动 I2C/SPI 通信	STCOMM 0x0723	1	1	1	0	0	1	0	0	0	1	1
放电静默	Mute 0x0028	0	0	0	0	0	1	0	1	0	0	0
放电不静默	Unmute 0x0029	0	0	0	0	0	1	0	1	0	0	1

表 32 命令位描述

名称	描述	数值		
MD[1:0]	ADC 模式	MD	ADCOPT(CFGR0[0]=0)	
		00	422Hz 模式	1kHz 模式
		01	27kHz 模式(快速)	14kHz 模式
		10	7kHz 模式(标准)	3kHz 模式

		11	26Hz 模式(滤波)				2kHz 模式				
DCP	允许放电	DCP									
		0	放电未得到允许								
		1	放电得到允许								
CH[2:0]	ADC 转换电池选择			8 种 ADC 模式中的总转换时间							
		CH		27kHz	14kHz	7kHz	3kHz	2kHz	1kHz	422Hz	26Hz
		000	所有电池	3ms	5ms	6ms	10ms	18ms	35ms	70ms	310ms
		001	电池 1,7,13	500μs	833μs	1ms	1.67ms	3ms	5.83ms	11.67ms	52ms
		010	电池 2,8,14								
		011	电池 3,9,15								
		100	电池 4,10,16								
		101	电池 5,11,17								
		110	电池 6,12,18								
PUP	导线开路/下拉	PUP									
		0	下拉电流								
		1	上拉电流								
ST[1:0]	自测试模式选择			自测试转换结果							
		ST		27kHz	14kHz	7kHz	3kHz	2kHz	1kHz	422Hz	26Hz
		01	自测试 1	0x6AAC	0x6AAC	0x6AAA	0x6AAA	0x6AAA	0x6AAA	0x6AAA	0x6AAA
		10	自测试 2	0x9554	0x9554	0x9555	0x9555	0x9555	0x9555	0x9555	0x9555
CHG[2:0]	ADC 转换 GPIO 选择			8 种 ADC 模式中的总转换时间							
		CHG		27KHz	14KHz	7KHz	3KHz	2KHz	1KHz	422Hz	26Hz
		000	GPIO 1-9, 第二基准	6ms	9ms	10ms	17ms	30ms	59ms	117ms	517ms
		001	GPIO 1, GPIO6	1100us	1.67ms	2.5ms	3.5ms	6ms	12ms	24ms	104ms
		010	GPIO 2, GPIO7								
		011	GPIO 3, GPIO8								
		100	GPIO 4, GPIO9								
		101	GPIO 5	550us	0.86ms	1.25ms	1.75ms	3ms	6ms	12ms	52ms

		110	第二基准	550us	0.86ms	1.25ms	1.75ms	3ms	6ms	12ms	52ms
CHST[2:0]	ADC 转换状态组选择			8 种 ADC 模式中的总转换时间							
		CHST		27KHz	14KHz	7KHz	3KHz	2KHz	1KHz	422Hz	26Hz
		000	SOC ITMP VA VD	2ms	3.4ms	4ms	6.7ms	12ms	23ms	47ms	207ms
		001	SOC	500us	830us	1ms	1.67ms	3ms	5.9ms	11.7ms	51.7ms
		010	ITMP								
		011	VA								
		100	VD								

表 33 配置寄存器组 A

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CFGR0	RD/WR	GPIO5	GPIO4	GPIO3	GPIO2	GPIO1	REFON	DTEN	ADCOPT
CFGR1	RD/WR	VUV[7:0]							
CFGR2	RD/WR	VOV[3:0]				VUV[11:8]			
CFGR3	RD/WR	VOV[11:4]							
CFGR4	RD/WR	DCC[8:1]							
CFGR5	RD/WR	DCT[3:0]				DCC[12:9]			

表 34 配置寄存器组 B

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CFGR0	RD/WR	DCC[16:13]				GPIO9	GPIO8	GPIO7	GPIO6
CFGR1	RD/WR	MUTE	FDRF	PS[1]	PS[0]	DTMEN	DCC0	DCC18	DCC17
CFGR2	RD/WR	RSVD0							
CFGR3	RD/WR								
CFGR4	RD/WR								
CFGR5	RD/WR								

表 35 电池电压寄存器组 A

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CVR0	RD	C1V[7:0]							
CVR1	RD	C1V[15:8]							
CVR2	RD	C2V[7:0]							

CVR3	RD	C2V[15:8]
CVR4	RD	C3V[7:0]
CVR5	RD	C3V[15:8]

表 36 电池电压寄存器组 B

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CVR0	RD	C4V[7:0]							
CVR1	RD	C4V[15:8]							
CVR2	RD	C5V[7:0]							
CVR3	RD	C5V[15:8]							
CVR4	RD	C6V[7:0]							
CVR5	RD	C6V[15:8]							

表 37 电池电压寄存器组 C

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CVR0	RD	C7V[7:0]							
CVR1	RD	C7V[15:8]							
CVR2	RD	C8V[7:0]							
CVR3	RD	C8V[15:8]							
CVR4	RD	C9V[7:0]							
CVR5	RD	C9V[15:8]							

表 38 电池电压寄存器组 D

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CVR0	RD	C10V[7:0]							
CVR1	RD	C10V[15:8]							
CVR2	RD	C11V[7:0]							
CVR3	RD	C11V[15:8]							
CVR4	RD	C12V[7:0]							
CVR5	RD	C12V[15:8]							

表 39 电池电压寄存器组 E

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
----	-------	------	------	------	------	------	------	------	------

CVR0	RD	C13V[7:0]
CVR1	RD	C13V[15:8]
CVR2	RD	C14V[7:0]
CVR3	RD	C14V[15:8]
CVR4	RD	C15V[7:0]
CVR5	RD	C15V[15:8]

表 40 电池电压寄存器组 F

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
CVR0	RD	C16V[7:0]							
CVR1	RD	C16V[15:8]							
CVR2	RD	C17V[7:0]							
CVR3	RD	C17V[15:8]							
CVR4	RD	C18V[7:0]							
CVR5	RD	C18V[15:8]							

表 41 辅助寄存器组 A

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
AUXR0	RD	G1V[7:0]							
AUXR1	RD	G1V[15:8]							
AUXR2	RD	G2V[7:0]							
AUXR3	RD	G2V[15:8]							
AUXR4	RD	G3V[7:0]							
AUXR5	RD	G3V[15:8]							

表 42 辅助寄存器组 B

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
AUXR0	RD	G4V[7:0]							
AUXR1	RD	G4V[15:8]							
AUXR2	RD	G5V[7:0]							
AUXR3	RD	G5V[15:8]							
AUXR4	RD	REF[7:0]							

AUXR5	RD	REF[15:8]
-------	----	-----------

表 43 辅助寄存器组 C

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
AUXR0	RD	G6V[7:0]							
AUXR1	RD	G6V[15:8]							
AUXR2	RD	G7V[7:0]							
AUXR3	RD	G7V[15:8]							
AUXR4	RD	G8V[7:0]							
AUXR5	RD	G8V[15:8]							

表 44 辅助寄存器组 D

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
AUXR0	RD	G9V[7:0]							
AUXR1	RD	G9V[15:8]							
AUXR2	RD	0x98							
AUXR3	RD	0x05							
AUXR4	RD	C16OV	C16UV	C15OV	C15UV	C14OV	C14UV	C13OV	C13UV
AUXR5	RD	RSVD1				C18OV	C18UV	C17OV	C17UV

表 45 状态寄存器组 A

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
STAR0	RD	SOC[7:0]							
STAR1	RD	SOC[15:8]							
STAR2	RD	ITMP[7:0]							
STAR3	RD	ITMP[15:8]							
STAR4	RD	VA[7:0]							
STAR5	RD	VA[15:8]							

表 46 状态寄存器组 B

名称	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
STBR0	RD	VD[7:0]							
STBR1	RD	VD[15:8]							

STBR2	RD	C4OV	C4UV	C3OV	C3UV	C2OV	C2UV	C1OV	C1UV
STBR3	RD	C8OV	C8UV	C7OV	C7UV	C6OV	C6UV	C5OV	C5UV
STBR4	RD	C12OV	C12UV	C11OV	C11UV	C10OV	C10UV	C9OV	C9UV
STBR5	RD	REV[3]	REV[2]	REV[1]	REV[0]	RSVD	RSVD	MUXFAIL	THSD

表 47 COMM 寄存器组

寄存器	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
COMM0	RD/WR	ICOM0[3:0]				D0[7:4]			
COMM1	RD/WR	D0[3:0]				FCOM0[3:0]			
COMM2	RD/WR	ICOM1[3:0]				D1[7:4]			
COMM3	RD/WR	D1[3:0]				FCOM1[3:0]			
COMM4	RD/WR	ICOM2[3:0]				D2[7:4]			
COMM5	RD/WR	D2[3:0]				FCOM2[3:0]			

表 48 S 控制寄存器组

寄存器	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
SCTRL0	RD/WR	SCTL2[3]	SCTL2[2]	SCTL2[1]	SCTL2[0]	SCTL1[3]	SCTL1[2]	SCTL1[1]	SCTL1[0]
SCTRL1	RD/WR	SCTL4[3]	SCTL4[2]	SCTL4[1]	SCTL4[0]	SCTL3[3]	SCTL3[2]	SCTL3[1]	SCTL3[0]
SCTRL2	RD/WR	SCTL6[3]	SCTL6[2]	SCTL6[1]	SCTL6[0]	SCTL5[3]	SCTL5[2]	SCTL5[1]	SCTL5[0]
SCTRL3	RD/WR	SCTL8[3]	SCTL8[2]	SCTL8[1]	SCTL8[0]	SCTL7[3]	SCTL7[2]	SCTL7[1]	SCTL7[0]
SCTRL4	RD/WR	SCTL10[3]	SCTL10[2]	SCTL10[1]	SCTL10[0]	SCTL9[3]	SCTL9[2]	SCTL9[1]	SCTL9[0]
SCTRL5	RD/WR	SCTL12[3]	SCTL12[2]	SCTL12[1]	SCTL12[0]	SCTL11[3]	SCTL11[2]	SCTL11[1]	SCTL11[0]

表 49 PWM 控制寄存器组

寄存器	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
PWM0	RD/WR	PWM2[3]	PWM2[2]	PWM2[1]	PWM2[0]	PWM1[3]	PWM1[2]	PWM1[1]	PWM1[0]
PWM1	RD/WR	PWM4[3]	PWM4[2]	PWM4[1]	PWM4[0]	PWM3[3]	PWM3[2]	PWM3[1]	PWM3[0]
PWM2	RD/WR	PWM6[3]	PWM6[2]	PWM6[1]	PWM6[0]	PWM5[3]	PWM5[2]	PWM5[1]	PWM5[0]
PWM3	RD/WR	PWM8[3]	PWM8[2]	PWM8[1]	PWM8[0]	PWM7[3]	PWM7[2]	PWM7[1]	PWM7[0]
PWM4	RD/WR	PWM10[3]	PWM10[2]	PWM10[1]	PWM10[0]	PWM9[3]	PWM9[2]	PWM9[1]	PWM9[0]
PWM5	RD/WR	PWM12[3]	PWM12[2]	PWM12[1]	PWM12[0]	PWM11[3]	PWM11[2]	PWM11[1]	PWM11[0]

表 50 PWM/S 控制寄存器组 B

寄存器	RD/WR	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
PWM0	RD/WR	PWM14[3]	PWM14[2]	PWM14[1]	PWM14[0]	PWM13[3]	PWM13[2]	PWM13[1]	PWM13[0]
PWM1	RD/WR	PWM16[3]	PWM16[2]	PWM16[1]	PWM16[0]	PWM15[3]	PWM15[2]	PWM15[1]	PWM15[0]
PWM2	RD/WR	PWM18[3]	PWM18[2]	PWM18[1]	PWM18[0]	PWM17[3]	PWM17[2]	PWM17[1]	PWM17[0]
PWM3	RD/WR	SCTL14[3]	SCTL14[2]	SCTL14[1]	SCTL14[0]	SCTL13[3]	SCTL13[2]	SCTL13[1]	SCTL13[0]
PWM4	RD/WR	SCTL16[3]	SCTL16[2]	SCTL16[1]	SCTL16[0]	SCTL15[3]	SCTL15[2]	SCTL15[1]	SCTL15[0]
PWM5	RD/WR	SCTL18[3]	SCTL18[2]	SCTL18[1]	SCTL18[0]	SCTL17[3]	SCTL17[2]	SCTL17[1]	SCTL17[0]

表 51 存储位描述

名称	描述	数值
GPIOx	GPIOx 引脚控制	写：0-GPIOx 引脚下拉电路导通；1-GPIOx 引脚下拉电路关断
		读：0-GPIOx 引脚处于逻辑“0”；1-GPIOx 引脚处于逻辑“1”
REFON	基准上电	1-基准保持上电状态直到看门狗超时为止
		0-基准再转换操作之后关断
DTEN	软件定时器状态	0：软件定时器关闭
		1：软件定时器使能
ADCOPT	ADC 模式选择位	0-MD[1:0]选择 27KHz、7KHz、422Hz 或 26Hz 模式
		1-MD[1:0]选择 14KHz、3KHz、1KHz 或 2KHz 模式
VUV	欠压比较电压	比较电压 = $(VUV+1) * 16 * 100\mu V$
		默认：0x000
VOV	过压比较电压	比较电压 = $VOV * 16 * 100\mu V$
		默认：0x000
DCC[x]	对电池 x 放电	x=1 至 18
		1-接通电池 x 的短路开关；0-断开电池 x 的短路开关（默认）
		x=0
		1-开启 GPIO9 的下拉；0-关闭 GPIO9 的下拉（默认）
DCTO	软件定时器设置	
MUTE	静默状态	1：静默使能
		0：静默不使能

FDRF	强制数字冗余诊断失效	1: 强制数字冗余诊断失效 0: 不强制数字冗余诊断失效
PS[1:0]	数字冗余诊断通道选择	11: 冗余仅应用于 ADC3 数字路径 10: 冗余仅应用于 ADC2 数字路径 01: 冗余仅应用于 ADC1 数字路径 00: 在电池转换期间将冗余依次应用于 ADC1、ADC2 和 ADC3 数字路径, 在 AUX 和 STATUS 转换期间将冗余应用于 ADC1
DTMEN	使能放电计时器 监控器	1: 如果 DTEN 引脚置为有效, 则使能放电定时器监视功能 0: 禁用放电定时器监视功能。如果 DTEN 引脚置为有效, 则使能正常的放电定时器功能
CxV	电池 x 电压	x=1 至 18 电池电压=CxV*100uV
		上电复位后和清零命令后 0xFFFF
GxV	电池 x 电压	x=1 至 9 GPIO 电压=CxV*100uV
		上电复位后和清零命令后 0xFFFF
REF	第二基准电压	第二基准电压=REF*100uV
		范围 2.980V~3.020V
SOC	电池测量总和	总压=SOC*100uV*30
ITMP	内部芯片温度	内部温度=ITMP*100uV/7.5mV/°C-273°C
VA	模拟电源电压	模拟电源电压=VA*100uV
VD	数字电源电压	数字电源电压=VD*100uV
CxOV	电池 x 过压标记	x=1 至 18 0-电池未过压; 1-电池过压
CxUV	电池 x 欠压标记	x=1 至 18 0-电池未欠压; 1-电池欠压
REV	版本代码	器件版本代码
RSVD0	保留位	读: 始终位 0
RSVD1	保留位	读: 始终位 1
MUXFAIL	多路复用器自测试结果	读: 0-多路复用器自测试通过 1-多路复用器自测试未通过
THSD	热停机状态	读: 0-热停机未发生; 1-热停机发生
SCTLn[x]	S 引脚控制位	0000: S 连续高电平
		0001: 发送 1 个脉冲

		0010: 发送 2 个脉冲 0011: 发送 3 个脉冲 0100: 发送 4 个脉冲 0101: 发送 5 个脉冲 0110: 发送 6 个脉冲 0111: 发送 7 个脉冲 1xxx: S 连续低电平																																																																																													
PWMn[x]	PWM 均衡控制		<table><tr><th>DCC BIT</th><th>PWMn[3:0] (n=1,2...18)</th><th>开启时间（s）</th><th>关断时间（s）</th><th>占空比</th></tr><tr><td>0</td><td>4'bXXXX</td><td>0</td><td>连续关断</td><td>0%</td></tr><tr><td>1</td><td>4'b1111</td><td>连续开启</td><td>0</td><td>100%</td></tr><tr><td>1</td><td>4'b1110</td><td>28</td><td>2</td><td>93.3%</td></tr><tr><td>1</td><td>4'b1101</td><td>26</td><td>4</td><td>86.7%</td></tr><tr><td>1</td><td>4'b1100</td><td>24</td><td>6</td><td>80%</td></tr><tr><td>1</td><td>4'b1011</td><td>22</td><td>8</td><td>73.3%</td></tr><tr><td>1</td><td>4'b1010</td><td>20</td><td>10</td><td>66.7%</td></tr><tr><td>1</td><td>4'b1001</td><td>18</td><td>12</td><td>60%</td></tr><tr><td>1</td><td>4'b1000</td><td>16</td><td>14</td><td>53.3%</td></tr><tr><td>1</td><td>4'b0111</td><td>14</td><td>16</td><td>46.7%</td></tr><tr><td>1</td><td>4'b0110</td><td>12</td><td>18</td><td>40%</td></tr><tr><td>1</td><td>4'b0101</td><td>10</td><td>20</td><td>33.3%</td></tr><tr><td>1</td><td>4'b0100</td><td>8</td><td>22</td><td>26.7%</td></tr><tr><td>1</td><td>4'b0011</td><td>6</td><td>24</td><td>20%</td></tr><tr><td>1</td><td>4'b0010</td><td>4</td><td>26</td><td>13.3%</td></tr><tr><td>1</td><td>4'b0001</td><td>2</td><td>28</td><td>6.7%</td></tr><tr><td>1</td><td>4'b0000</td><td>0</td><td>连续关断</td><td>0%</td></tr></table>	DCC BIT	PWMn[3:0] (n=1,2...18)	开启时间（s）	关断时间（s）	占空比	0	4'bXXXX	0	连续关断	0%	1	4'b1111	连续开启	0	100%	1	4'b1110	28	2	93.3%	1	4'b1101	26	4	86.7%	1	4'b1100	24	6	80%	1	4'b1011	22	8	73.3%	1	4'b1010	20	10	66.7%	1	4'b1001	18	12	60%	1	4'b1000	16	14	53.3%	1	4'b0111	14	16	46.7%	1	4'b0110	12	18	40%	1	4'b0101	10	20	33.3%	1	4'b0100	8	22	26.7%	1	4'b0011	6	24	20%	1	4'b0010	4	26	13.3%	1	4'b0001	2	28	6.7%	1	4'b0000	0	连续关断	0%		
DCC BIT	PWMn[3:0] (n=1,2...18)	开启时间（s）	关断时间（s）	占空比																																																																																											
0	4'bXXXX	0	连续关断	0%																																																																																											
1	4'b1111	连续开启	0	100%																																																																																											
1	4'b1110	28	2	93.3%																																																																																											
1	4'b1101	26	4	86.7%																																																																																											
1	4'b1100	24	6	80%																																																																																											
1	4'b1011	22	8	73.3%																																																																																											
1	4'b1010	20	10	66.7%																																																																																											
1	4'b1001	18	12	60%																																																																																											
1	4'b1000	16	14	53.3%																																																																																											
1	4'b0111	14	16	46.7%																																																																																											
1	4'b0110	12	18	40%																																																																																											
1	4'b0101	10	20	33.3%																																																																																											
1	4'b0100	8	22	26.7%																																																																																											
1	4'b0011	6	24	20%																																																																																											
1	4'b0010	4	26	13.3%																																																																																											
1	4'b0001	2	28	6.7%																																																																																											
1	4'b0000	0	连续关断	0%																																																																																											

编程实例

下面的例子采用了一种 3 个 MT9805 器件堆栈的配置：S1、S2、S3。器件 S1 上的端口 A 被配置为 SPI 模式 (ISOMD 引脚为低电平)。器件 S2 和 S3 上的端口 A 被配置为 isoSPI 模式 (ISOMD 引脚为高电平)。S1 上的端口 B 连接至 S2 上的端口 A。S2 上的端口 B 连接至 S3 上的端口 A。微控制器通过 S1 上的端口 A 与堆栈进行通信。

唤醒串行接口

1. 发送一个虚字节。CSB 和 SCK 上的动作将唤醒器件 S1 上的串行接口。
2. 等待 $3 \cdot t_{\text{WAKE}}$ 的时间以给所有的器件 (S1、S2 和 S3) 上电。在大型堆栈中，有些器件在唤醒之后可能会变至 IDLE 状态，对此可实施步骤 3 和 4：
3. 发送第二个虚字节。
4. 等待 $3 \cdot t_{\text{READY}}$ 的时间。
5. 发送命令。

写配置寄存器

1. 把 CSB 拉至低电平
2. 发送 WRCFG 命令 (0x00 0x01) 及其 PEC (0x3D 0x6E)
3. 发送器件 S3 的 CFGR0 字节，然后是 CFGR1(S3)，...CFGR5(S3)，CFGR0(S3) 至 CFGR5(S3) 的 PEC
4. 发送器件 S2 的 CFGR0 字节，而后是 CFGR1(S2)，...CFGR5(S2)，CFGR0(S2) 至 CFGR5(S2) 的 PEC
5. 发送器件 S1 的 CFGR0 字节，接着是 CFGR1(S1)，...CFGR5(S1)，CFGR0(S1) 至 CFGR5(S1) 的 PEC
6. 把 CSB 拉至高电平，数据在 CSB 的上升沿上被锁定至所有的器件中针对上述序列的串行接口时间计算：

菊链式堆栈中的 MT9805 数目 = n

序列中的字节数目 (B)：

命令：2 个 (命令字节) + 2 个 (命令 PEC) = 4 个

数据：每个 MT9805 6 个 (数据字节) + 2 个 (数据 PEC) = 每个器件 8 个字节

$$B = 4 + 8 \cdot n$$

每个位的串行端口频率 = F

$$\text{时间} = (1/F) \cdot B \cdot 8 \text{ 位/字节} = (1/F) \cdot [4 + 8 \cdot n] \cdot 8$$

对于上述 3 个 MT9805 堆栈实例的时间 (采用 1MHz 串行

$$\text{端口}) = (1/1e6) \cdot (4 + 8 \cdot 3) \cdot 8 = 224 \mu\text{s}$$

注：该时间对于所有的写命令和读命令保持相同。

读电池寄存器组 A

- 1.把 CSB 拉至低电平
- 2.发送 RDCVA 命令(0x00 0x04)及其 PEC(0x07 0xC2)
- 3.读取器件 S1 的 CVAR0 字节, 然后是 CVAR1(S1), ...CVAR5(S1), CVAR0(S1)至 CVAR5(S1)的 PEC
- 4.读取器件 S2 的 CVAR0 字节, 而后是 CVAR1(S2), ...CVAR5(S2), CVAR0(S2)至 CVAR5(S2)的 PEC
- 5.读取器件 S3 的 CVAR0 字节, 接着是 CVAR1(S3), ...CVAR5(S3), CVAR0(S3)至 CVAR5(S3)的 PEC
- 6.把 CSB 拉至高电平

启动电池电压 ADC 转换

(所有电池, 标准模式, 且允许放电)和轮询状态

- 1.把 CSB 拉至低电平
- 2.发送具有 MD[1:0]=10 和 DCP=1 的 ADCV 命令(即 0x03 0x70)及其 PEC(0xAF 0x42)把 CSB 拉至高电平

清除电池电压寄存器

- 1.把 CSB 拉至低电平
- 2.发送 CLRCELL 命令(0x07 0x11)及其 PEC(0xC9 0xC0)
- 3.把 CSB 拉至高电平

连接至 MT9805 的 I2C 受控器通信

通过连接至 GPIO4(SDA)和 GPIO5(SCL), MT9805 可支持 I2C 从属器件。该功能的一种有价值的用途是将生产校准常数或其他信息存储在一个小的串行 EEPROM 中。

该例采用独立 MT9805-1 将一个数据字节写至一个 I2CEEPROM。MT9805 将发送三个数据字节至 I2C 从属器件。发送的数据将为 B0=0xA0(EEPROM 地址), B1=0x01(写命令)和 B2=0xAA(即将存储在 EEPROM 中的数据)。

三个字节将以下列格式传输至 I2C 从属器件: START-B0-NACK-B1-NACK-B2-NACK-STOP

- 1.采用 WRCOMM 命令将数据写至 COMM 寄存器
 - a.把 CSB 拉至低电平
 - b.发送 WRCOMM 命令(0x07 0x21)及其 PEC(0x24 0xB2)
 - c.发送 COMM0=0x6A,

COMM1=0x08([START][B0][NACK]),

COMM2=0x00, COMM3=0x18([BLANK][B1][NACK]),

COMM4=0x0A, COMM5=0xA9([BLANK][B2]

[NACK+STOP])和 PEC=0x6D 0xFB(对于上述数据)

d.把 CSB 拉至高电平

2.采用 STCOMM 命令将 3 个数据字节发送至 I2C 从属器件

a.把 CSB 拉至低电平

b.发送 STCOMM 命令(0x07 0x23)及其 PEC(0xB9 0xE4)

c.在 SCK 上发送 72 个时钟周期

d.把 CSB 拉至高电平

3.在 STCOMM 命令期间传输至受控器的数据被存储在 COMM 寄存器中。采用 RDCOMM 命令以取回数据

a.把 CSB 拉至低电平

b.发送 RDCOMM 命令(0x07 0x22)及其 PEC(0x32 0xD6)

c.读取 COMM0-COMM5 和 6 个数据字节的 PEC 假设受控器确认了所有的 3 个数据字节, 则该例中的回读数据应该是:

COMM0=0x6A,COMM1=0x07, COMM2=0x70,

COMM3=0x17, COMM4=0x7A, COMM5=0xA1, PEC=0xD0 0xDE

d.把 CSB 拉至高电平。

上例中, 在 STCOMM 命令期间 I2C 主控器的 GPIO5(SCL)和 GPIO4(SDA)端口上 72 个时钟周期内的活动情况示于图 33。

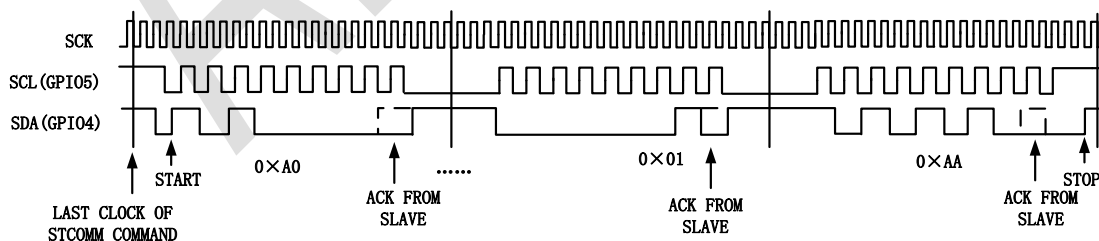


图 33 MT9805 I2C 通信实例

与一个连接至 MT9805 的 SPI 受控器通信

该例采用了单个 MT9805-1 器件，它具有一个通过 GPIO3(CSBM)、GPIO4(SDOM)和 GPIO5(SCKM)与其相连的 SPI 器件。在此实例中，MT9805 器件采用下面的格式将 3 个数据字节 B0=0x55、B1=0xAA 和 B2=0xCC 发送至 SPI 从属器件：CSB 为低电平–B0–B1–B2–CSB 为高电平

1.采用 WRCOMM 命令将数据写至 COMM 寄存器

a.把 CSB 拉至低电平

b.发送 WRCOMM 命令(0x07 0x21)及其 PEC(0x240xB2)

c.发送

COMM0=0x85,

COMM1=0x50([CSBM low][B0][CSBM low]),

COMM2=0x8A,

COMM3=0xA0([CSBM low][B1][CSBM low]),

COMM4=0x8C,

COMM5=0xC9([CSBM low][B2][CSBM high])和 PEC=0x89 0xA4(对于上述数据)。

d.把 CSB 拉至高电平。

2.采用 STCOMM 命令将 3 个数据字节发送至 SPI 从属器

件

a. 把 CSB 拉至低电平

b.发送 STCOMM 命令(0x07 0x23)及其 PEC(0xB9 0xE4)

c.在 SCK 上发送 72 个时钟周期

d.把 CSB 拉至高电平

3.在 STCOMM 命令期间传输至受控器的数据被存储在 COMM 寄存器中。采用 RDCOMM 命令以取回数据

a.把 CSB 拉至低电平

b.发送 RDCOMM 命令(0x07 0x22)及其 PEC(0x32 0xD6)

c.读取 COMM0-COMM5 和 6 个数据字节的 PEC。该例中的回读数据应该是：

COMM0=0x755F,

COMM1=0x7AAF,

COMM2=7CCF, PEC=0xF2BA

d.把 CSB 拉至高电平

上例中，在 STCOMM 命令期间 SPI 主控器的 GPIO3(CSBM)、GPIO5(SCKM)和 GPIO4(SDOM)端口上 72 个时钟周期内的活动情况示于图 34。

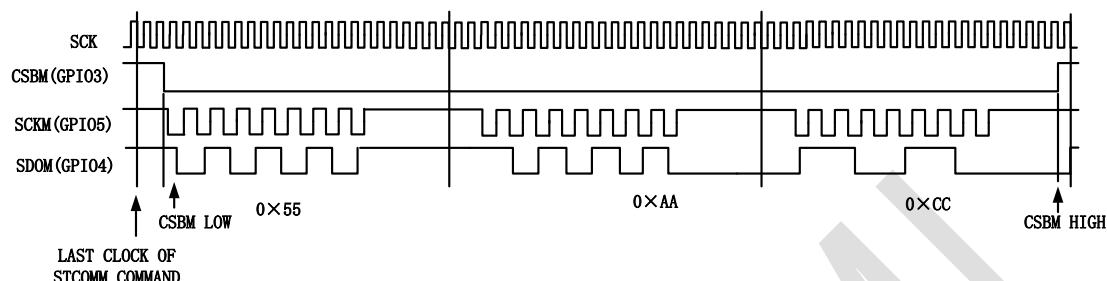


图 34 MT9805 SPI 通信实例

应用

简单的线性稳压器

MT9805 从 VREG 引脚吸取其所需的大部分功率。应给 VREG 施加 $5V \pm 0.5V$ 的电压。一个稳压的 DC/DC 转换器能够直接给 VREG 供电，或者，也可以采用 DRIVE 引脚和少量附加的外部组件形成一个分立式稳压器。当运行时，

DRIVE 输出引脚提供一个低电流 5.6V 输出，该输出可采用一个分立的 NPN 晶体管进行缓冲，如图 35 所示。NPN 的集电极电源可以取自任何比 V-高 6V 或更多的电位，包括被监视的电池或一个未经调节的转换器电源。集电极电源连接建议使用一个 $100\Omega/100nF$ RC 去耦网络，以保护 NPN 免遭瞬变的损坏。NPN 的发射极应采用一个 $1\mu F$ 电容器进行旁路。应避免采用较大的电容器数值，因为这会增加 MT9805 的唤醒时间。必需对 NPN 的热特性稍加关注，因为当采用一个高集电极电压时会产生大量的热量。图 35 示出的 CZT5551 是一款可提供良好设计裕量的 SOT-223 封装器件。

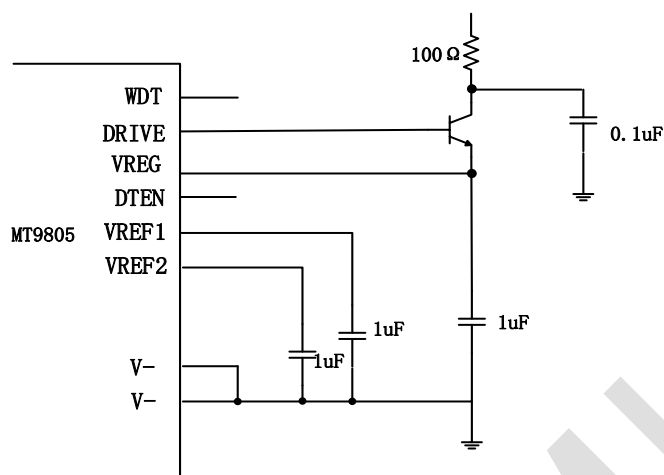


图 35 采用 NPN 传输晶体管的简单 VREG 电源

电池和 GPIO 输入的滤波

MT9805 采用了一个 $\Delta \Sigma$ ADC，该 ADC 具有 $\Delta \Sigma$ 调制器和一个位于其后的 SINC3 有限脉冲响应(FIR)数字滤波器。这极大地降低了输入滤波要求。此外，可编程过采样比还使用户能够在测量速度与滤波器截止频率之间确定最佳的折衷方案。即使采用这种高阶低通滤波器，快速瞬态噪声还是会在测量中引起某些残留噪声，特别是在较快的转换模式中。

这种噪声可以通过给每个 ADC 输入增设一个 RC 低通去耦滤波器来尽量降低，此方法还有助于抑制具潜在破坏性的高能量瞬变。给 ADC 输入增添约 100Ω 以上的电阻将开始在测量中引起系统误差，这可以通过提高滤波器电容或借助软件和一种校准程序进行数学补偿予以改善。对于需要最高电池电压纹波抑制水平的场合，建议采用接地电容器滤波。

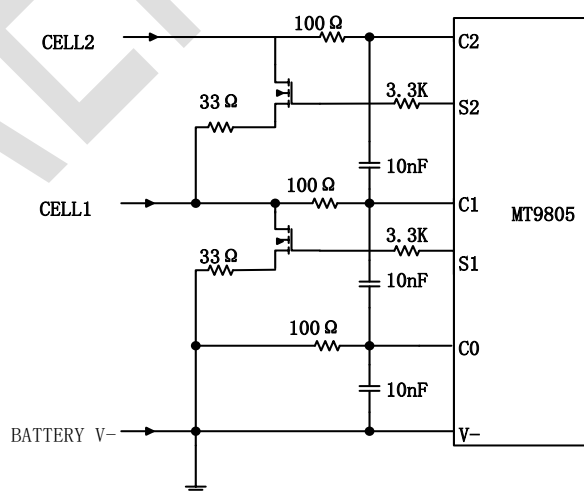


图 36 差分电容滤波器

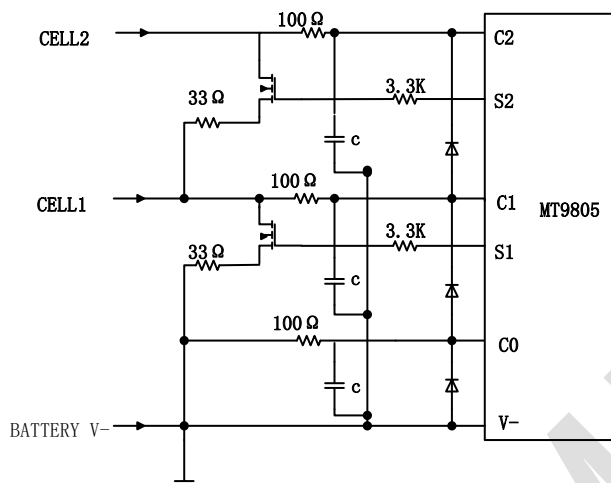


图 37 输入滤波器结构配置

该配置具有一个串联电阻和电容器，可将 HF 噪声去耦至 V-。在那些噪声不太具有周期性或者使用较高过采样速率的系统中，采用差分电容器滤波器结构就足够了。在该配置中，每个输入都有串联电阻器，但电容器则连接在相邻的 C 引脚之间。然而，差分电容器节段会相互影响。因此，滤波器响应不太一致并导致低于 RC 预测值的衰减(差了一个数量级)。请注意，电容器仅承受一个电池的施加电压(因而尺寸较小和成本较低)且往往在 IC 上均匀地分布瞬态能量(从而减少了内部保护结构上的应力事件)。基本的 ADC 准确度随 R、C 而改变，但在 $R=100\Omega$ 和 $C=10\text{nF}$ 时可实现误差的最小化。由于测量均参考于 V-，因此 GPIO 引脚将始终采用一种接地电容器配置。

利用内部 MOSFET 进行电池平衡

S1 至 S18 引脚用于平衡电池单元。如果一串电池中的某一节过度充电，则可采用一个 S 输出对这节电池进行放电。每个 S 输出具有一个用于放电的内部 N 沟道 MOSFET。该 NMOS 具有一个 10Ω 的最大导通电阻。应连接一个与 NMOS 相串联的外部电阻器以把热量散逸到 MT9805 封装的外部，如图 38 所示。仍然可以使用一个 RC 以给电池电压测量增加额外的滤波，但滤波器 R 必须保持很小(通常在 10Ω 左右)以减轻对于编程平衡电流的影响。当采用内部 MOSFET 给电池放电时，应监视芯片温度。见“功率耗散和热停机”部分。

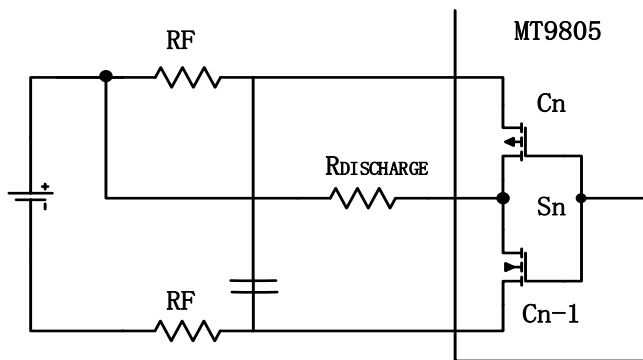


图 38 内部放电电路

利用外部 MOSFET 进行电池平衡

S 输出包括一个内部上拉 PMOS 晶体管。S 引脚能够充当一个适合于驱动外部 MOSFET 的栅极的数字输出。对于那些需要高电池放电电流的应用，将一个分立型 PMOS 开关器件和合适的放电电阻器连接至电池，并把栅极端子连接至 S 输出引脚，如图 39 示出了包括 RC 滤波的外部 MOSFET 电路。

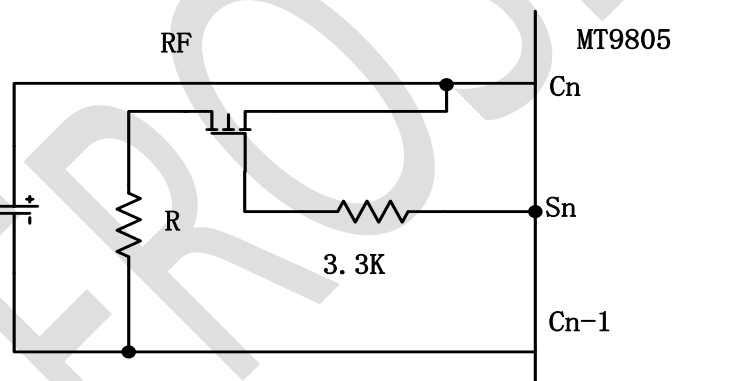


图 39 外部放电电路

电池测量期间的放电控制

如果放电允许(DCP)命令位在电池测量命令中为高电平，则 S 引脚放电状态在电池测量期间不改变。然而，倘若 DCP 位为低电平，那么任何处于接通状态的放电都将在对应的电池或相邻的电池处于被测量状态时关断。表 52 说明了在一个具有 DCP=0 的 ADCV 命令期间的此类情况。在该表中，OFF 表示在该周期中放电被强制关断，即使配置寄存器中的对应 DCC[x]位为高电平也不例外。ON 意味着假如放电接通，则其将在该周期中保持导通状态。

功率耗散和过热关闭

连接至引脚 S1 至 S18 的内部 MOSFET 可用于对电池进行放电。应采用一个外部电阻器以限制 MOSFET 消耗

的功率。MOSFET 中的最大功率耗散受限于 MT9805 所能容许的热量。过高的热量将导致芯片温度的升高。在芯片温度高达 125°C 的情况下，几乎或完全没有观察到测量准确度的下降。当温度高于 150°C 时有可能出现器件受损，因此建议的最大芯片温度为 125°C。为了保护 MT9805 免遭过热的损坏，内置了一个热停机电路。当在电池放电开关中消耗大量的功率时会出现器件过热的情况。每当器件不处于睡眠状态时，热停机电路即被使能(见“操作模式”)。如果器件上的检测温度变至高于约 150°C，则配置寄存器将被复位至默认状态，从而切断所有的放电开关。当发生了一个热停机时，状态寄存器组 B 中的 THSD 位将走高。这个位将在状态寄存器组 B 的一个读操作之后清零。另外，这个位也可以采用 CLRSTAT 命令来设定。由于热停机会中断正常的操作，因此应采用内部温度监视器以确定器件温度逐渐接近不可接受之水平的时刻。

表 52 DCP=0 的 ADCV 命令期间的放电控制

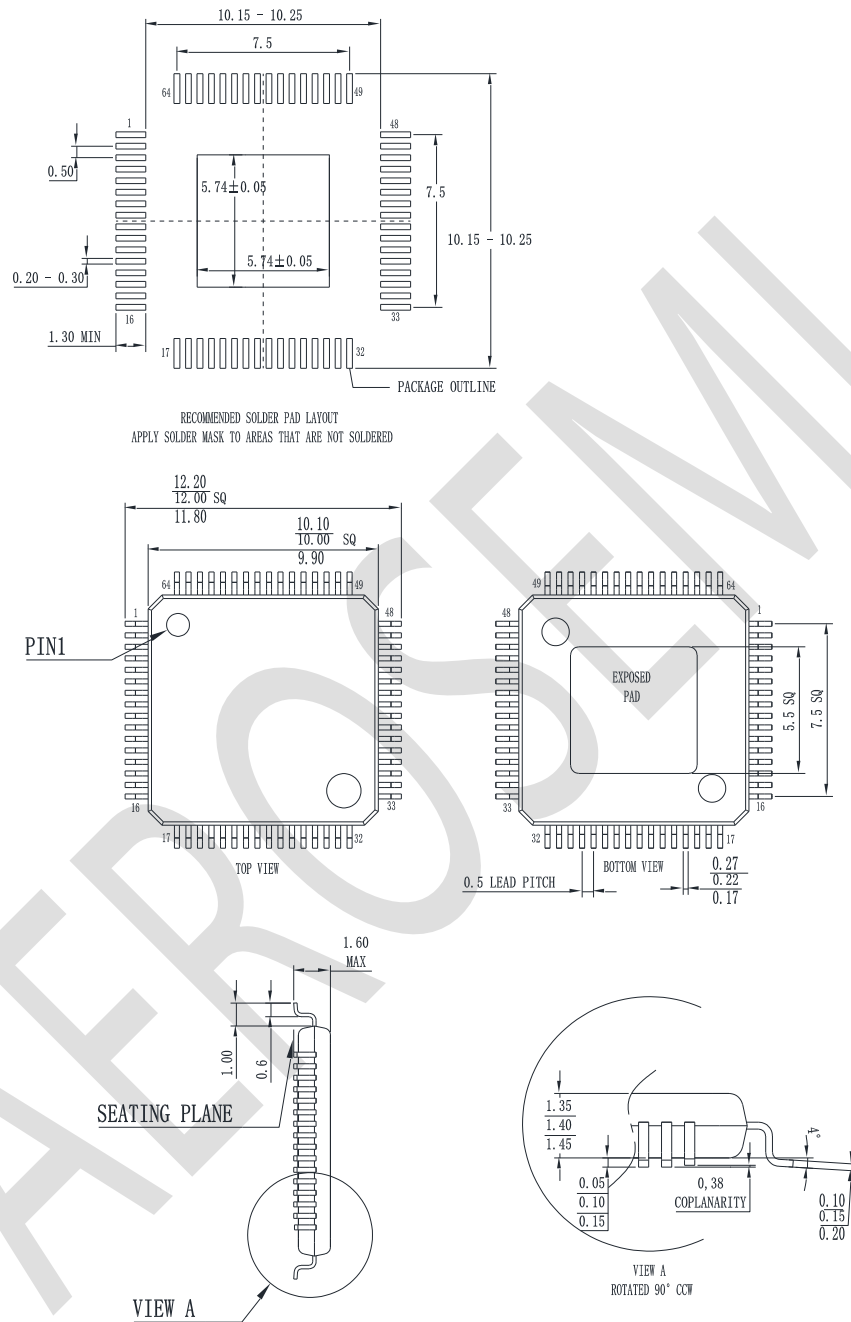
电池电压测量周期						
放电引脚	CELL1/CELL7	CELL2/CELL8	CELL3/CELL9	CELL4/CELL10	CELL5/CELL11	CELL6/CELL12
S1	OFF	OFF	ON	ON	ON	OFF
S2	OFF	OFF	OFF	ON	ON	ON
S3	ON	OFF	OFF	OFF	ON	ON
S4	ON	ON	OFF	OFF	OFF	OFF
S5	ON	ON	ON	OFF	OFF	OFF
S6	OFF	ON	ON	ON	OFF	OFF
S7	OFF	OFF	ON	ON	ON	OFF
S8	OFF	OFF	OFF	ON	ON	ON
S9	ON	OFF	OFF	OFF	ON	ON
S10	ON	ON	OFF	OFF	OFF	OFF
S11	ON	ON	ON	OFF	OFF	OFF
S12	OFF	ON	ON	ON	OFF	OFF
S13	OFF	OFF	ON	ON	ON	OFF
S14	OFF	OFF	OFF	ON	ON	ON
S15	ON	OFF	OFF	OFF	ON	ON
S16	ON	ON	OFF	OFF	OFF	OFF
S17	ON	ON	ON	OFF	OFF	OFF
S18	OFF	ON	ON	ON	OFF	OFF
电池电压校准周期						
放电引脚	CELL1/CELL7	CELL2/CELL8	CELL3/CELL9	CELL4/CELL10	CELL5/CELL11	CELL6/CELL12
S1	OFF	OFF	ON	ON	ON	OFF
S2	OFF	OFF	OFF	ON	ON	ON
S3	ON	OFF	OFF	OFF	ON	ON
S4	ON	ON	OFF	OFF	OFF	OFF
S5	ON	ON	ON	OFF	OFF	OFF
S6	OFF	ON	ON	ON	OFF	OFF
S7	OFF	OFF	ON	ON	ON	OFF
S8	OFF	OFF	OFF	ON	ON	ON

S9	ON	OFF	OFF	OFF	ON	ON
S10	ON	ON	OFF	OFF	OFF	OFF
S11	ON	ON	ON	OFF	OFF	OFF
S12	OFF	ON	ON	ON	OFF	OFF
S13	OFF	OFF	ON	ON	ON	OFF
S14	OFF	OFF	OFF	ON	ON	ON
S15	ON	OFF	OFF	OFF	ON	ON
S16	ON	ON	OFF	OFF	OFF	OFF
S17	ON	ON	ON	OFF	OFF	OFF
S18	OFF	ON	ON	ON	OFF	OFF

采用 MT9805 监视电池数目小于 18 的电池组

如果 MT9805 由电池组供电，则 MT9805 所能监视的最小电池数目受控于 MT9805 的电源电压要求。电池电压的总和应至少为 11V 以正确地给 MT9805 施加偏置。MT9805 监视的 18 个电池分为三组(各含 6 个电池)，并采用三个内部多路复用器和三个 ADC 进行测量。为了优化电池数目少于 18 之应用中的测量同步，未用的 C 引脚应在第三个多路复用器的顶端(C18)、第二个多路复用器的顶端(C12)和第一个多路复用器的顶端(C6)之间均匀分布。如果使用的电池数目为奇数，则顶端多路复用器应连接较少的电池。未用的电池通道应连接至同一个多路复用器上的其他未用通道，然后通过一个 100Ω 电阻器连接至电池组。对于那些电池通道，未用的输入将产生一个 0V 的读数。另外，采用常规序列进行连接(将所有的未用电池输入置于顶端)也是可以接受的。

封装形式



Unit:mm

HLQFP(64Pin)

版本号	修订内容	修订时间
1.3	格式修订	2023-08-25
1.4	图 3，图 4，图 5 修正；“电特性参数”中的 Ivp 单位更正； “测量 18 节电池”时间更正；“ADC 时钟频率 fs”参数更正； ADOW 指令描述更正；表 17 ICOMn 和 FCOMn 描述更正；表 34 内容更正；表 51 DCC 描述更正；增加 PS 和 DTMEN 描述	2023-09-19
1.5	指令位描述 ST 的内容；ISOMD 端口描述；图 37 电容接地线 更正；表 32 的 ADC 采集时间；“电特性参数表”中 V+电源 输入名称；Page39 的看门狗更正为看门狗；	2023-12-25
1.6	表 51 中 SOC 的计算与 SOC 文字描述保持一致；表 31 的 RDPWM (0x22) 更正为 (0x12)； 封装尺寸图及命名修改；	2024-02-22

重要声明

西安航天民芯科技有限公司保留对其半导体产品和服务进行更正、增强、改进和其他变更的权利。

西安航天民芯科技有限公司对此类文件的变更不承担任何责任。第三方的信息可能会受到额外的限制。西安航天民芯科技有限公司不承担因任何特定目的使用其产品的责任，也不承担因应用或使用其任意产品或电路而产生的任何责任。

版权所有：©西安航天民芯科技有限公司 2011

<http://www.aerosemi.com>

AEROSEMI