

# TCAN5102-Q1 汽车级 CAN FD 灯响应器（支持 SPI、UART 或 I<sup>2</sup>C 控制器和可编程 PWM）

## 1 特性

- AEC-Q100：符合汽车应用要求
- CAN FD Light 响应器符合 ISO 11898-1:202x 标准
- 使用 TCAN4572-Q1 CAN FD Light 命令器时，支持高达 5Mbps 的 CAN FD Light 数据速率
- 具备控制外部 CAN 收发器的能力，例如：TCAN1162x-Q1, TCAN1043A, TCAN1463A
- 可通过 CAN 总线进行编程
- 3.3V 至 5V 电源电压
- SPI 控制器支持最高 20MHz 时钟频率，最多可配置 8 个片选信号
- UART 控制器支持最高 2.5M 波特率
- I<sup>2</sup>C 控制器支持快速增强模式 (1MHz)
- 13 个 GPIO 引脚，可与其他功能复用
- 可编程 PWM 输出，支持用于电机控制的梯形斜坡曲线
- 20 引脚 VSSOP (DGQ) 封装

## 2 应用

- 车身电子装置和照明
- 混合动力、电动和动力总成系统
- 信息娱乐系统与仪表组
- 电器

## 3 说明

TCAN5102-Q1 是一款控制器局域网 (CAN) 灵活数据 (FD) 轻型响应器器件，符合 CiA 604-1 和 ISO 11898-1:2024 标准，当使用专用 CAN FD light 命令器时，能够支持高达 5Mbps 的数据速率。该器件设计用于支持命令器-响应器架构中的 CAN FD light 响应器节点应用，无需响应器节点处理器。对响应器节点的所有控制都通过来自命令器节点处理器的 CAN 总线进行，无需响应器节点处理器和软件。

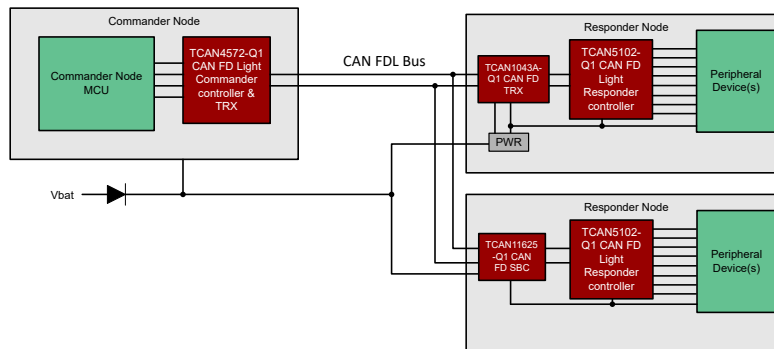
该器件从 CAN FD light 命令器节点接收数据和/或命令，并将其转换为串行外设接口 (SPI) 控制器、UART 控制器、I<sup>2</sup>C 控制器和/或 GPIO 引脚通信，以与 TCAN5102-Q1 所控制的器件和/或外设进行通信。PWM 输出通道在硬件上还支持梯形斜坡曲线，用于控制步进电机。可斜升占空比或频率。无需外部的晶体或时钟。该器件控制外部 TCAN1162x-Q1、TCAN1043A-Q1 或 TCAN1463A-Q1 CAN FD (SIC) 收发器，以实现系统级灵活性。该器件在 CAN FD 收发器/SBC 上进行重新连接，以控制节点电源，并通过将 CAN RXD (CRXD) 引脚锁定为低电平，将唤醒信号传送至 TCAN5102-Q1。

### 封装信息

器件型号	封装 <sup>(1)</sup>	封装尺寸 <sup>(2)</sup>
TCAN5102-Q1	20 引脚 ( VSSOP、DGQ )	5.1mm × 4.9mm

(1) 有关更多信息，请参阅 节 11。

(2) 封装尺寸 (长 × 宽) 为标称值，并包括引脚 (如适用)。



CAN FD 光命令器/响应器应用

## 内容

1 特性.....	1	7.3 特性说明.....	16
2 应用.....	1	7.4 器件功能模式.....	19
3 说明.....	1	7.5 编程.....	20
4 引脚配置和功能.....	3	7.6 寄存器映射.....	53
5 规格.....	4	8 应用和实施.....	221
5.1 绝对最大额定值.....	4	8.1 应用信息.....	221
5.2 ESD 等级.....	4	8.2 典型应用.....	221
5.3 建议运行条件.....	4	8.3 电源相关建议.....	222
5.4 热性能信息.....	4	8.4 布局.....	222
5.5 电源特性.....	5	9 器件和文档支持.....	224
5.6 电气特性.....	5	9.1 文档支持.....	224
5.7 时序要求.....	6	9.2 接收文档更新通知.....	224
5.8 开关特性.....	7	9.3 支持资源.....	224
5.9 I <sup>2</sup> C 总线时序要求.....	9	9.4 商标.....	224
6 参数测量信息.....	11	9.5 静电放电警告.....	224
7 详细说明.....	14	9.6 术语表.....	224
7.1 概述.....	14	10 修订历史记录.....	224
7.2 功能方框图.....	14	11 机械、封装和可订购信息.....	224

## 4 引脚配置和功能

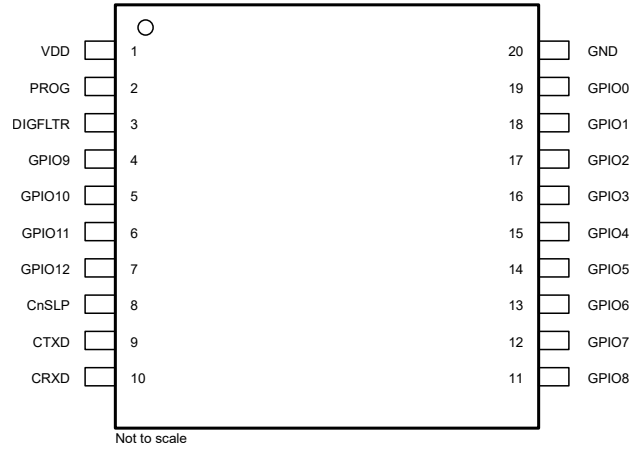


图 4-1. DGQ 20-PIN ( 顶视图 )

表 4-1. 引脚功能

引脚		类型 <sup>(1)</sup>	说明
名称	编号		
VDD	1	P	器件电源输入, 3.3V 或 5V
PROG	2	I	用于器件配置和调试的输入
DIGFLTR	3	P	数字电源滤波器引脚
GPIO9/CS4/SCL	4	I/O	通用输入/输出引脚、SPI 片选通道 4 输出或 I <sup>2</sup> C 时钟输出
GPIO10/CS5/SDA	5	I/O	通用输入/输出引脚、SPI 片选通道 5 输出或 I <sup>2</sup> C 数据引脚
GPIO11/CS6/PWM0	6	I/O	通用输入/输出引脚、SPI 片选通道 6 输出或 PWM0 输出
GPIO12/CS7/PWM1	7	I/O	通用输入/输出引脚、SPI 片选通道 6 输出或 PWM1 输出
CnSLP	8	O	CAN 收发器控制输出引脚 (nSLP)
CTXD	9	O	CAN 收发器 TXD 输出引脚 ( 连接到收发器的 TXD 输入引脚 )
CRXD	10	I	CAN 收发器 RXD 输入引脚 ( 连接到收发器的 RXD 输出引脚 )
GPIO8/URXD	11	I/O	通用输入/输出引脚或 UART RXD 输入
GPIO7/UTXD	12	I/O	通用输入/输出引脚或 UART TXD 输出
GPIO6/PICO	13	I/O	通用输入/输出引脚或 SPI 外设输入、控制器输出
GPIO5/POCI	14	I/O	通用输入/输出引脚或 SPI 外设输出、控制器输入
GPIO4/SCK	15	I/O	通用输入/输出引脚或 SPI 时钟输出
GPIO3/CS0	16	I/O	通用输入/输出引脚或 SPI 片选通道 0 输出
GPIO2/CS1	17	I/O	通用输入/输出引脚或 SPI 片选通道 1 输出
GPIO1/CS2	18	I/O	通用输入/输出引脚或 SPI 片选通道 2 输出
GPIO0/CS3	19	I/O	通用输入/输出引脚或 SPI 片选通道 3 输出
GND	20	P	接地引脚

(1) I = 输入, O = 输出, I/O = 输入或输出, G = 接地, P = 电源。

### 备注

对于具有多种功能的引脚, 请参见 IO\_CFG 寄存器来配置引脚, 并参阅节 7.3.4 了解有关引脚多路复用的更多信息。

## 5 规格

### 5.1 绝对最大额定值

在自然通风条件下的工作温度范围内测得（除非另有说明）<sup>(1)</sup>

		最小值	最大值	单位
VDD	电源输入	-0.3	6	V
DIGFLTR	数字内核滤波器引脚	-0.3	1.8	V
V <sub>LOGIC_IN</sub>	逻辑引脚输入电压范围	-0.3	6	V
V <sub>LOGIC_OUT</sub>	逻辑引脚输出电压范围	-0.5	6	V
I <sub>O(LOGIC)</sub>	逻辑引脚输出电流		12	mA
T <sub>J</sub>	结温	-40	150	°C
T <sub>stg</sub>	贮存温度	-65	150	°C

(1) 超出“绝对最大额定值”运行可能会对器件造成永久损坏。绝对最大额定值并不表示器件在这些条件下或在建议运行条件以外的任何其他条件下能够正常运行。如果超出建议运行条件但在绝对最大额定值范围内使用，器件可能不会完全正常运行，这可能影响器件的可靠性、功能和性能，并缩短器件寿命。

### 5.2 ESD 等级

			值	单位
V <sub>(ESD)</sub>	静电放电	人体放电模型 (HBM) 分类等级 3A，所有其他引脚，符合 AEC Q100-002 <sup>(1)</sup>	±2000	V
		充电器件模型 (CDM) 分级等级 C5，转角引脚	±750	
		符合 AEC Q100-011 标准 其他引脚	±750	

(1) AEC Q100-002 指示应当按照 ANSI/ESDA/JEDEC JS-001 规范执行 HBM 应力测试。

### 5.3 建议运行条件

在自然通风条件下的工作温度范围内测得（除非另有说明）

		最小值	标称值	最大值	单位
VDD	电源电压	3.0		5.5	V
I <sub>OH(DO)</sub>	数字输出高电平电流	-12			mA
I <sub>OL(DO)</sub>	数字输出低电平电流			12	mA
C <sub>(VDD)</sub>	VDD 电源电容	1			μF
C <sub>(DIGFLTR)</sub>	数字内核滤波器引脚电容	0.33	1		μF
ESR <sub>CO</sub>	电源和滤波器引脚 ESR 电容要求	0.001		1	Ω
T <sub>J</sub>	工作结温范围	-40		125	°C

### 5.4 热性能信息

热指标 <sup>(1)</sup>		TCAN5102-Q1	单位
		RGY (QFN)	
		24-PINS	
R <sub>θJA</sub>	结至环境热阻	31.8	°C/W
R <sub>θJB</sub>	结至电路板热阻	11.8	°C/W
R <sub>θJC(top)</sub>	结至外壳（顶部）热阻	21.4	°C/W
R <sub>θJC(bot)</sub>	结至外壳（底部）热阻	2.8	°C/W
Ψ <sub>JT</sub>	结至顶部特征参数	0.3	°C/W

## 5.4 热性能信息 (续)

热指标 <sup>(1)</sup>		TCAN5102-Q1	单位
		RGY (QFN)	
		24-PINS	
$\Psi_{JB}$	结至电路板特征参数	11.8	°C/W

(1) 有关新旧热指标的更多信息, 请参阅[半导体和 IC 封装热指标](#)应用报告。

## 5.5 电源特性

在自然通风条件下的工作温度范围内测得 (除非另有说明)

参数	测试条件	最小值	典型值	最大值	单位
VDD					
IDD	VDD 电源电流器件, 正常模式	正常模式; VDD = 5 V $I_I = 0$ , 0 V 或 VDD SPI、UART、I2C 禁用		1.4	1.8 mA
		正常模式; VDD = 3.3 V $I_I = 0$ , 0 V 或 VDD SPI、UART、I2C 禁用		1.4	1.8 mA
IDD	VDD 电源电流器件, 睡眠模式	睡眠模式; VDD = 5V $I_I = 0$ , 0V 或 VDD $-40^{\circ}\text{C} \leq T_J \leq 85^{\circ}\text{C}$		10	20 $\mu\text{A}$
		睡眠模式; VDD = 5V $I_I = 0$ , 0V 或 VDD $T_J > 85^{\circ}\text{C}$		20	60 $\mu\text{A}$
		睡眠模式; VDD = 3.3V $I_I = 0$ , 0V 或 VDD $-40^{\circ}\text{C} \leq T_J \leq 85^{\circ}\text{C}$		10	20 $\mu\text{A}$
		睡眠模式; VDD = 3.3V $I_I = 0$ , 0V 或 VDD $T_J > 85^{\circ}\text{C}$		20	60 $\mu\text{A}$
UV <sub>DDR</sub>	电源欠压上升阈值	VDD 上升		2.35	2.95 V
UV <sub>DDF</sub>	电源欠压下降阈值	VDD 下降		2.1	2.7 V
UV <sub>DDHYS</sub>	电源欠压检测迟滞			350	430 mV

## 5.6 电气特性

在自然通风条件下的工作温度范围内测得 (除非另有说明)

参数	测试条件	最小值	典型值	最大值	单位
GPIO (输入)					
V <sub>IH</sub>	高电平输入电压	GPIOx 引脚		0.7	VDD
		CRXD 引脚		0.7	
		PROG 引脚		0.7	
V <sub>IL</sub>	低电平输入电压	GPIOx 引脚		0.3	VDD
		CRXD 引脚		0.3	
		PROG 引脚		0.3	

## 5.6 电气特性 (续)

在自然通风条件下的工作温度范围内测得 (除非另有说明)

参数		测试条件	最小值	典型值	最大值	单位
I <sub>IH</sub>	高电平输入漏电流	GPIOx 引脚 输入 = VDD 禁用上拉/下拉电阻器	-1		1	μA
		CRXD 引脚 输入 = VDD	-1		1	
		PROG 引脚 输入 = VDD		90	140	
I <sub>IL</sub>	低电平输入漏电流	GPIOx 引脚 输入 = 0V 禁用上拉/下拉电阻器	-1		1	μA
		PROG 引脚 输入 = 0V	-1		1	
		CRXD 引脚 输入 = 0V	-140	-90		
C <sub>IN</sub>	输入电容	20MHz, GPIOx 引脚		4	10	pF
		1MHz, CRXD 引脚		4	10	
		1MHz, PROG 引脚		4	10	
I <sub>LKG(OFF)</sub>	未供电时的漏电流	GPIOx 引脚 输入 = 5.5V, VDD = 0V	-1	0	1	μA
R <sub>PD</sub>	下拉电阻	GPIOx ( 如果已启用 )	40	60	80	k Ω
		PROG	40	60	80	
R <sub>PU</sub>	上拉电阻	GPIOx ( 如果已启用 )	40	60	80	k Ω
		CRXD	40	60	80	
GPIO ( 输出 )						
V <sub>OH</sub>	高电平输出电压	GPIOx、CTXD、CnSLP 引脚 I <sub>OH</sub> = 10mA	0.7	0.85		VDD
		GPIOx、CTXD、CnSLP 引脚 I <sub>OH</sub> = 2mA	0.8	0.9		
V <sub>OL</sub>	低电平输出电压	GPIOx、CTXD、CnSLP 引脚 I <sub>OL</sub> = - 10mA		0.15	0.3	VDD
		GPIOx、CTXD、CnSLP 引脚 I <sub>OL</sub> = - 2mA		0.1	0.2	
I <sub>OH</sub>	高电平输出电流	GPIOx、CTXD、CnSLP 引脚 V <sub>O</sub> = 0.7 VDD	10	12		mA
I <sub>OL</sub>	低电平输出电流	GPIOx、CTXD、CnSLP 引脚 V <sub>O</sub> = 0.3 VDD		-12	-10	mA
I <sub>LKG(OFF)</sub>	未供电时的漏电流	CTXD、CnSLP 引脚 VPIN = 5.5V	-5		5	μA
热关断						
TSD <sub>F</sub>	热关断下降		155	171	180	°C
TSD <sub>R</sub>	热关断上升		160	176	190	°C
TSD <sub>HYS</sub>	热关断迟滞		2	5	10	°C

## 5.7 时序要求

	最小值	典型值	最大值	单位
电源				

## 5.7 时序要求 (续)

		最小值	典型值	最大值	单位
t <sub>PWRUP</sub>	VDD 超过 POR 之后器件上电所需的时间			2	ms
<b>模式更改</b>					
t <sub>MODE_NOM_SLP</sub>	从进入睡眠命令到器件关闭 CAN 收发器所用的时间		100	600	μs
<b>器件时序</b>					
t <sub>MODE_POR_NOM</sub>	从 POR 到正常模式，为接收 CAN 消息做好准备的时间		150	400	μs
<b>SPI 时序要求</b>					
t <sub>HD-DAT</sub>	有效 SCK 边沿后的 POCI 输入保持时间要求 请参阅 <a href="#">SPI 时序特性</a>	20			ns
t <sub>SU-DAT</sub>	有效 SCK 边沿之前的 POCI 输入设置时间要求 请参阅 <a href="#">SPI 时序特性</a>	5			ns
<b>UART</b>					
t <sub>W(RX)</sub>	脉冲宽度，接收开始、停止，数据位 <sup>(1)</sup>	0.97	1	1.04	U

(1) U = UART 波特时间 = 1/编程波特率

## 5.8 开关特性

在自然通风条件下的工作温度范围内测得 (除非另有说明)

参数		测试条件	最小值	典型值	最大值	单位
SPI 开关特性						
f <sub>SCK-MAX</sub>	SCK，最大 SPI 时钟频率 <sup>(1)</sup>	正常模式；C <sub>L</sub> = 20 pF	20			MHz
t <sub>SCK-MIN</sub>	SCK，最小 SPI 时钟周期 <sup>(1)</sup>	正常模式；C <sub>L</sub> = 20 pF	50			ns
t <sub>SCKR</sub>	SCK 上升时间 <sup>(1)</sup>	正常模式；C <sub>L</sub> = 20 pF	4      10			ns
t <sub>SCKF</sub>	SCK 下降时间 <sup>(1)</sup>	正常模式；C <sub>L</sub> = 20 pF	4      8			ns
t <sub>SCKH</sub>	SCK，SPI 时钟高电平 <sup>(1)</sup>	正常模式	0.47	0.49	0.5	S <sup>(3)</sup>
t <sub>SCKL</sub>	SCK，SPI 时钟低电平 <sup>(1)</sup>	正常模式	0.5	0.51	0.53	S <sup>(3)</sup>
t <sub>SCK-CS</sub>	从最后一个 SCK 边沿到 CS 无效的延迟时间 <sup>(1)</sup>	正常模式 请参阅 <a href="#">SPI 时序特性</a>	0.44	0.5	0.52	S <sup>(3)</sup>
t <sub>CS-SCK</sub>	从 CS 有效到第一个 SCK 边沿的延迟时间 <sup>(1)</sup>	正常模式 请参阅 <a href="#">SPI 时序特性</a>	0.49	0.5	0.57	S <sup>(3)</sup>
t <sub>HD-DAT</sub>	从捕获 SCK 边沿到 PICO 数据变化的数据保持时间 <sup>(1)</sup>	正常模式 请参阅 <a href="#">SPI 时序特性</a>	0.47	0.5	0.51	S <sup>(3)</sup>
t <sub>CS-DV</sub>	片选有效边沿到 PICO 数据有效的延迟时间 <sup>(1)</sup>	正常模式 SPI 模式 0、2 请参阅 <a href="#">SPI 时序特性</a>	0      5			ns
		正常模式 SPI 模式 1、3 请参阅 <a href="#">SPI 时序特性</a>	0.5      0.57			S <sup>(3)</sup>
I2C 开关特性						
t <sub>STUCKBUS_I2C</sub>	总线“卡在”低电平且直到器件认为总线卡住的时间。		25	40	65	ms
f <sub>SB_SCLOUT_I2C</sub>	尝试取消固定总线时所生成的 I2C 总线阻塞恢复模式时钟频率		5.5	8.5	14	kHz
UART						
t <sub>baud</sub>	最大 UART 波特率		2.5			MHz
t <sub>UARTTXD</sub>	从 CAN 命令完成到 UART 发送开始的时间		1      2			U

5.8 开关特性（续）

在自然通风条件下的工作温度范围内测得（除非另有说明）

参数		测试条件	最小值	典型值	最大值	单位
t <sub>UARTRXD</sub>	从 UART 接收到 CAN 读取就绪所用的时间 <sup>(1)</sup>	最后一个数据/奇偶校验位结束到设置 UART.RXN 位 C <sub>L</sub> = 20pF		0.5	2	U
t <sub>W(TX)</sub>	脉冲宽度，发送开始、停止，数据位 <sup>(1)</sup> <sup>(2)</sup>	C <sub>L</sub> = 20pF	0.98		1.02	U

- (1) 根据设计确定
- (2) U = UART 波特时间 = 1/编程波特率
- (3) S = SPI 时钟周期 = 1/SPI 时钟频率



## 5.9 I<sup>2</sup>C 总线时序要求

在自然通风条件下的工作温度范围内测得（除非另有说明）

			最小值	最大值	单位
I <sup>2</sup> C 总线 - 标准模式					
f <sub>scl</sub>	I <sup>2</sup> C 时钟频率		0	100	kHz
t <sub>sch</sub>	I <sup>2</sup> C 时钟高电平时间 <sup>(1)</sup>		4		μs
t <sub>scl</sub>	I <sup>2</sup> C 时钟低电平时间 <sup>(1)</sup>		4.7		μs
t <sub>sp</sub>	I <sup>2</sup> C 尖峰时间			50	ns
t <sub>sds</sub>	I <sup>2</sup> C 串行数据设置时间		250		ns
t <sub>sdh</sub>	I <sup>2</sup> C 串行数据保持时间		0		ns
t <sub>icr</sub>	I <sup>2</sup> C 输入上升时间 <sup>(1)</sup>			1000	ns
t <sub>icf</sub>	I <sup>2</sup> C 输入下降时间			300	ns
t <sub>ocf</sub>	I <sup>2</sup> C 输出下降时间	10pF 至 400pF 总线		300	ns
t <sub>buf</sub>	停止和启动之间的 I <sup>2</sup> C 总线空闲时间		4.7		μs
t <sub>sts</sub>	I <sup>2</sup> C 启动或重复启动条件设置		4.7		μs
t <sub>sth</sub>	I <sup>2</sup> C 启动或重复启动条件保持		4		μs
t <sub>sps</sub>	I <sup>2</sup> C 停止条件设置		4		μs
t <sub>vd(data)</sub>	有效数据时间	SCL 低电平到 SDA 输出有效		3.45	μs
t <sub>vd(ack)</sub>	ACK 条件的有效数据时间	从 SCL 低电平到 SDA ( 输出 ) 低电平的 ACK 信号		3.45	μs
C <sub>b</sub>	I <sup>2</sup> C 总线容性负载			400	pF
I <sup>2</sup> C 总线 - 快速模式					
f <sub>scl</sub>	I <sup>2</sup> C 时钟频率		0	400	kHz
t <sub>sch</sub>	I <sup>2</sup> C 时钟高电平时间 <sup>(1)</sup>		0.6		μs
t <sub>scl</sub>	I <sup>2</sup> C 时钟低电平时间 <sup>(1)</sup>		1.3		μs
t <sub>sp</sub>	I <sup>2</sup> C 尖峰时间			50	ns
t <sub>sds</sub>	I <sup>2</sup> C 串行数据设置时间		100		ns
t <sub>sdh</sub>	I <sup>2</sup> C 串行数据保持时间		0		ns
t <sub>icr</sub>	I <sup>2</sup> C 输入上升时间 <sup>(1)</sup>		20	300	ns
t <sub>icf</sub>	I <sup>2</sup> C 输入下降时间		20 × (V <sub>CC</sub> /5.5V)	300	ns
t <sub>ocf</sub>	I <sup>2</sup> C 输出下降时间	10pF 至 400pF 总线	20 × (V <sub>CC</sub> /5.5V)	300	ns
t <sub>buf</sub>	停止和启动之间的 I <sup>2</sup> C 总线空闲时间		1.3		μs
t <sub>sts</sub>	I <sup>2</sup> C 启动或重复启动条件设置		0.6		μs
t <sub>sth</sub>	I <sup>2</sup> C 启动或重复启动条件保持		0.6		μs
t <sub>sps</sub>	I <sup>2</sup> C 停止条件设置		0.6		μs
t <sub>vd(data)</sub>	有效数据时间	SCL 低电平到 SDA 输出有效		0.9	μs
t <sub>vd(ack)</sub>	ACK 条件的有效数据时间	从 SCL 低电平到 SDA ( 输出 ) 低电平的 ACK 信号		0.9	μs
C <sub>b</sub>	I <sup>2</sup> C 总线容性负载			400	pF
I <sup>2</sup> C 总线 - 快速+ 模式					
f <sub>scl</sub>	I <sup>2</sup> C 时钟频率		0	1000	kHz

## 5.9 I<sup>2</sup>C 总线时序要求 (续)

在自然通风条件下的工作温度范围内测得 (除非另有说明)

			最小值	最大值	单位
t <sub>sch</sub>	I <sup>2</sup> C 时钟高电平时间 <sup>(1)</sup>		0.26		μs
t <sub>scl</sub>	I <sup>2</sup> C 时钟低电平时间 <sup>(1)</sup>		0.5		μs
t <sub>sp</sub>	I <sup>2</sup> C 尖峰时间			50	ns
t <sub>sds</sub>	I <sup>2</sup> C 串行数据设置时间		50		ns
t <sub>sdh</sub>	I <sup>2</sup> C 串行数据保持时间		0		ns
t <sub>icr</sub>	I <sup>2</sup> C 输入上升时间 <sup>(1)</sup>			120	ns
t <sub>icf</sub>	I <sup>2</sup> C 输入下降时间		20 × (V <sub>CC</sub> /5.5V)	120	ns
t <sub>ocf</sub>	I <sup>2</sup> C 输出下降时间	10pF 至 550pF 总线	20 × (V <sub>CC</sub> /5.5V)	120	ns
t <sub>buf</sub>	停止和启动之间的 I <sup>2</sup> C 总线空闲时间		0.5		μs
t <sub>sts</sub>	I <sup>2</sup> C 启动或重复启动条件设置		0.26		μs
t <sub>sth</sub>	I <sup>2</sup> C 启动或重复启动条件保持		0.26		μs
t <sub>sps</sub>	I <sup>2</sup> C 停止条件设置		0.26		μs
t <sub>vd(data)</sub>	有效数据时间	SCL 低电平到 SDA 输出有效		0.45	μs
t <sub>vd(ack)</sub>	ACK 条件的有效数据时间	从 SCL 低电平到 SDA (输出) 低电平的 ACK 信号		0.45	μs
C <sub>b</sub>	I <sup>2</sup> C 总线容性负载			550	pF

(1) 系统设计人员应当确保满足此参数要求。它会因为总线负载和上拉电阻而异。

## 6 参数测量信息

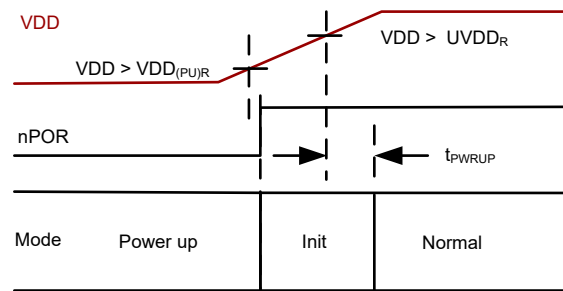


图 6-1. 上电时序

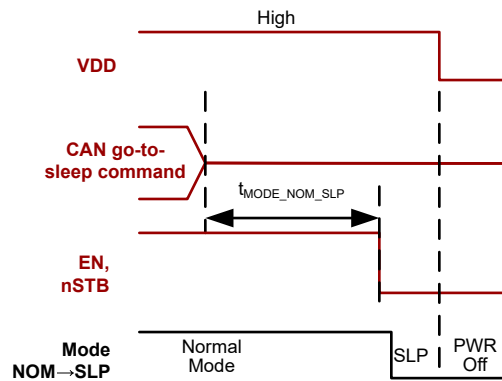


图 6-2. 正常模式到休眠模式时序

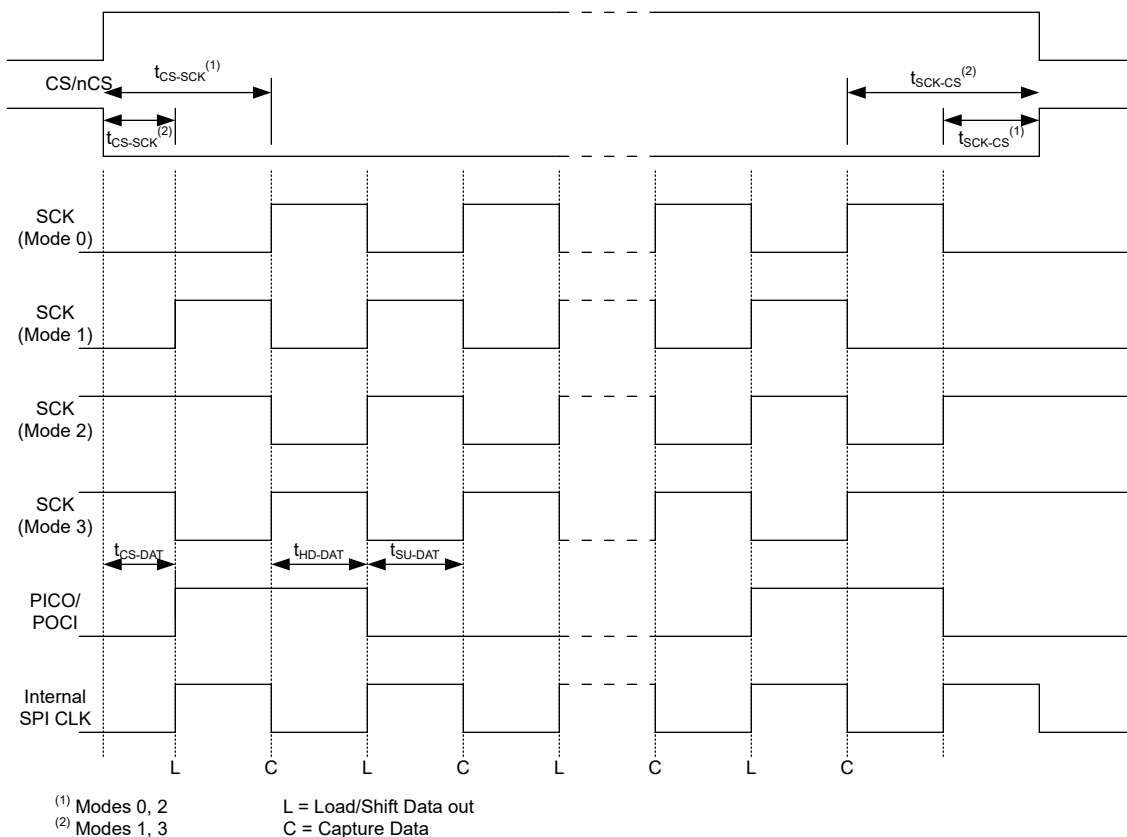


图 6-3. SPI 时序特性

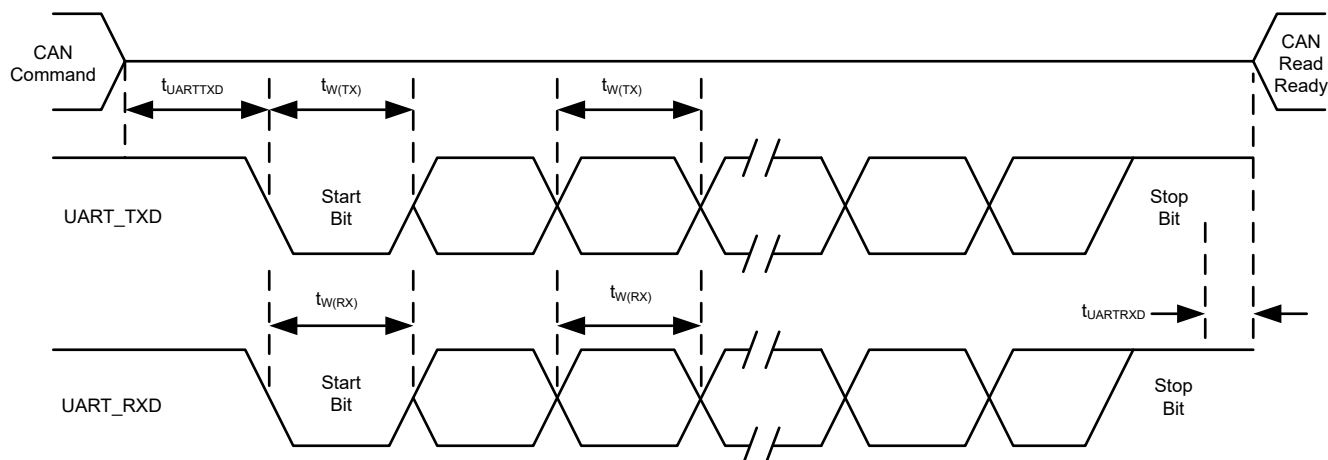
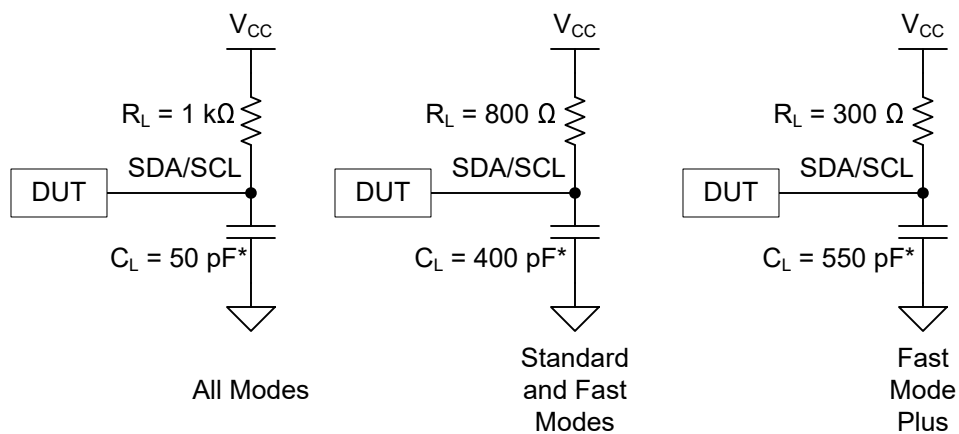


图 6-4. UART 时序特性



\* $C_L$  includes probe and jig capacitance

图 6-5. I2C 总线负载

## 7 详细说明

### 7.1 概述

TCAN5102-Q1 是一款控制器局域网 (CAN) 灵活数据 (FD) 轻型响应器器件, 符合 CiA 604-1 和 ISO 11898-1:2024 标准, 当使用专用 CAN FD light 命令器时, 能够支持高达 5Mbps 的数据速率。该器件设计用于支持命令器-响应器架构中的 CAN FD light 响应器节点应用, 无需响应器节点处理器。对响应器节点的所有控制都通过来自命令器节点处理器的 CAN 总线进行, 无需响应器节点处理器和软件。

该器件从 CAN FD light 命令器节点接收数据和/或命令, 并将其转换为串行外设接口 (SPI) 控制器、UART 控制器、I<sup>2</sup>C 控制器和/或 GPIO 引脚通信, 以与 TCAN5102-Q1 所控制的器件和/或外设进行通信。PWM 输出通道在硬件上还支持梯形斜坡曲线, 用于控制步进电机。可实现占空比或频率的斜坡控制。无需外部晶体或时钟。该器件控制外部 TCAN1162x-Q1、TCAN1043A-Q1 或 TCAN1463A-Q1 CAN FD (SIC) 收发器, 以实现系统级灵活性。该器件依靠 CAN FD 收发器/SBC 来控制节点电源并通过将 CAN RXD (CRXD) 引脚锁存为低电平, 来向 TCAN5102-Q1 发送唤醒信号。

### 7.2 功能方框图

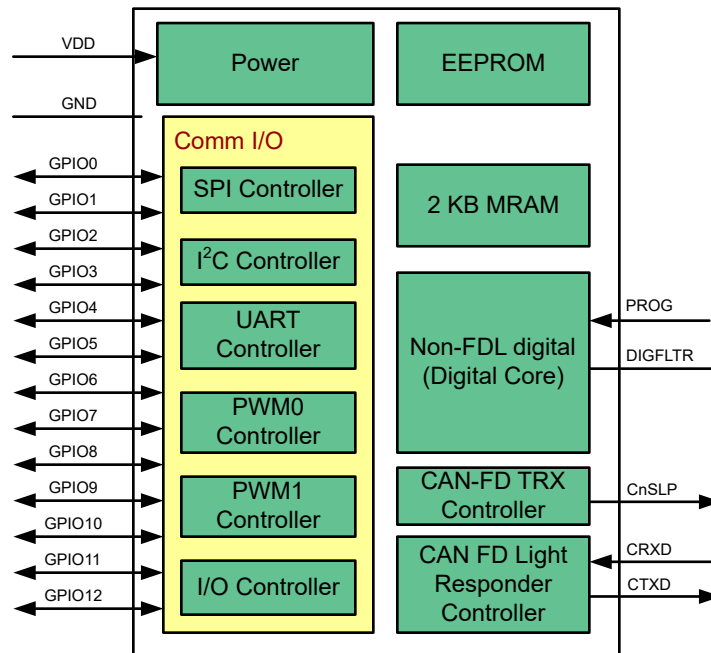


图 7-1. 高级方框图

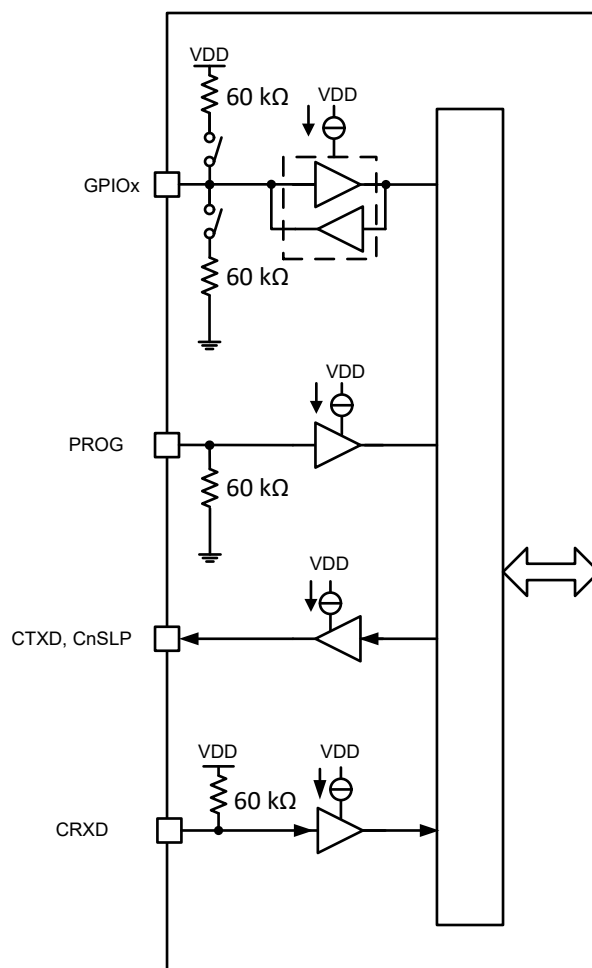


图 7-2. 输入和输出方框图

## 7.3 特性说明

### 7.3.1 VDD

VDD 是器件的电源输入，用于设置数字输入/输出电压电平 3.3V 和 5V。

### 7.3.2 PROG 引脚

在 ECU 开发阶段使用 PROG 引脚。上电时将此引脚设置为 GND，TCAN5102-Q1 可通过 CAN 总线进行下线编程。此过程中 CAN 总线编程数据速率设置为 1 Mbps。

### 7.3.3 DIGFLTR 引脚

此引脚用于为内部数字核心稳压器提供滤波。请参阅建议运行条件表，了解建议的滤波电容值。

### 7.3.4 GPIOx 和引脚多路复用特殊功能

TCAN5102-Q1 具有 12 个 GPIO 引脚，可配置为多种用途。作为通用 I/O，该引脚可配置为输入、推挽输出或漏极开路输出。这些引脚还提供用户可启用的集成上拉和下拉电阻器。

为了支持器件内不同的串行接口和 IP 块，多个引脚可以选择与 GPIO 多路复用。在寄存器配置中，这种额外功能被称为“特殊功能”，这意味着该引脚不再作为 GPIO 使用，而是连接到特定的 IP 模块。当配置为特殊功能时，该引脚仅使用上拉和下拉电阻器配置。当选择特殊功能后，系统会自动为您配置输出/输入、推挽/漏极开路等模式。例如，如果对应于 SPI POCI 的 GPIO 被选择为特殊功能，则该 GPIO 的设置会被 SPI IP 覆盖，以将该 GPIO 配置为输入。如果需要，可在该引脚上使能内部上拉或下拉电阻器，以确保引脚处于已知状态。

#### 备注

部分引脚被分配了 2 种特殊功能。若出现使能冲突，优先级较高的 IP 将获得优先权。例如，如果 SPI IP 块已使能 SPI 片选通道 4 (CS4) 并且 I<sup>2</sup>C IP 也已使能，则 GPIO9 会多路复用至 I<sup>2</sup>C SCL 引脚。所有引脚的默认未编程行为是输入和高阻抗。

GPIO 的默认上电状态可以存储在 EEPROM 中，以实现上电响应。引脚在未供电或低于 POR 阈值时处于高阻抗状态。器件上电后，配置信息会从 EEPROM 复制到 GPIO 寄存器。只有会影响输出行为的配置寄存器会存储在 EEPROM 中。这些被存储的寄存器包括

1. IO\_OE\_0 和 IO\_OE\_1
2. IO\_RE\_0 和 IO\_RE\_1
3. IO\_PU\_0 和 IO\_PU\_1
4. IO\_OUTPUT\_0 和 IO\_OUTPUT\_1

表 7-1. 用于特殊功能的引脚多路复用

引脚	名称	低优先级	高优先级
4	GPIO9	SPI CS4	I <sup>2</sup> C SCL
5	GPIO10	SPI CS5	I <sup>2</sup> C SDA
6	GPIO11	SPI CS6	PWM0
7	GPIO12	SPI CS7	PWM1
11	GPIO8	UART RXD	-
12	GPIO7	UART TXD	-
13	GPIO6	SPI PICO	-
14	GPIO5	SPI POCI	-
15	GPIO4	SPI SCK	-
16	GPIO3	SPI CS0	-
17	GPIO2	SPI CS1	-
18	GPIO1	SPI CS2	-



表 7-1. 用于特殊功能的引脚多路复用 (续)

引脚	名称	低优先级	高优先级
19	GPIO0	SPI CS3	-

#### 7.3.4.1 GPIO 同步

配置为输出的 GPIO 支持同时更新。此 GPIO 输出更新有 2 种模式

1. 默认值：当 IO\_OUTPUT\_x 寄存器更新时 (每个字节)，GPIO 会立即更新
2. 同步：仅在写入 IO\_OUTPUT\_1 后才会更新。若需同步所有 GPIO，需先向 IO\_OUTPUT\_0 写入数据；当向 IO\_OUTPUT\_1 的写入操作完成后，所有 GPIO 位将同时更新

通过将 MRAM\_IP\_CFG 寄存器中的 GPIO\_OUT\_SYNC 位设为 1，可启用该功能。

#### 7.3.5 EEPROM

TCAN5102-Q1 内置 EEPROM，用于存储上电配置信息，包括：

1. CAN 总线数据速率
2. CAN ID
3. 广播 CAN 标识符及掩码
4. 配置锁 (数据速率和 CAN ID)
5. 设置上电复位引脚状态所需的 GPIO 配置
  - a. IO\_OE\_0 和 IO\_OE\_1
  - b. IO\_RE\_0 和 IO\_RE\_1
  - c. IO\_PU\_0 和 IO\_PU\_1
  - d. IO\_OUTPUT\_0 和 IO\_OUTPUT\_1

#### 备注

EEPROM 包含 CRC 字段，若检测到 CRC 故障，器件会禁用 CAN 接口，并将 GPIO 置于高阻态

#### 7.3.6 SPI 控制器

TCAN5102-Q1 具有 SPI 控制器，其特点是有 8 个片选，且 SPI 通道 0 至 3 有不同的 SPI 设置。最后 4 个通道与 SPI 通道 3 共享配置。前 4 个通道中的每个通道都可以有独立的 SPI 速度和 SPI 模式。使用不同的片选可以通过共享的 SPI POCI、PICO 和 SCK 线进行通信。片选可以配置为高电平有效或低电平有效。消息存储在消息 RAM (MRAM) 中，并且 IP 必须在 MRAM 中分配空间才能运行。有关更多信息，请参阅节 7.5.3。

有关 SPI 控制器的配置和使用的更多信息，请参阅节 7.5.4。

#### 7.3.7 UART 控制器

TCAN5102-Q1 具有 UART 控制器。UART 是一种标准的面向字节的协议，用于许多不同的接口 (LIN、RS232、RS485、Flexwire 等)。该控制器具有小数波特率分频器，以支持多种接口使用的广泛波特率。

消息存储在消息 RAM (MRAM) 中，并且 IP 必须在 MRAM 中分配空间才能运行。有关更多信息，请参阅节 7.5.3。可以在不使用 MRAM 的情况下，以单字节模式运行 UART IP。UART 配置寄存器中有一个用于此模式的使能位。

有关 UART 控制器的配置和使用的更多信息，请参阅节 7.5.5。

#### 7.3.8 I2C 控制器

TCAN5102-Q1 具有 I2C 控制器，支持标准模式、快速模式和快速增强模式。除标准控制器外，该器件还内置了总线卡滞恢复功能，可检测 SDA 线何时卡滞为低电平，并切换 SCL 线以尝试解除总线卡滞。此外，只要时钟拉伸时间小于 t<sub>STUCKBUS\_I2C</sub> 时间，也支持时钟拉伸。

消息存储在消息 RAM ( MRAM ) 中，并且 IP 必须在 MRAM 中分配空间才能运行。有关更多信息，请参阅节 7.5.3。

有关 I<sup>2</sup>C 控制器的配置和使用的更多信息，请参阅节 7.5.6。

#### 7.3.8.1 I2C 总线卡滞恢复

I2C 控制器具有一项可选功能，称为总线卡滞恢复 (SBR)。当 SDA 或 SCL 线路“卡滞”在低电平时，总线卡滞恢复功能会尝试修复总线。若总线上存在噪声，导致 I2C 外设检测到错误的时钟数量，就可能出现这种情况。总线卡滞恢复功能最多能将 SCL 线路切换 16 次，以尝试让有问题的器件释放总线。有关更多信息，请参阅节 7.5.6.2。

#### 7.3.9 PWM 控制器

TCAN5102-Q1 具有集成斜坡控制功能的 PWM 控制器。PWM 控制器支持静态/典型 PWM 输出 ( 固定占空比和频率 )，还支持高级 PWM 功能，如占空比或频率斜坡控制，以支持基本的电机控制。该控制器在开发时就考虑了步进电机控制，允许用户编程加速和减速曲线。一旦告知控制器开始斜坡控制，它会进行计算并在 PWM 输出上执行指定的斜坡曲线。

PWM 控制器甚至支持 PWM 脉冲计数等附加功能，允许用户配置 PWM 输出以生成加速和减速曲线，然后在一定数量的脉冲/步进后自动停止。该序列被编程到寄存器中，除了被告知开始斜坡控制外，不需要上游控制器的干预。

对于附加功能，GPIO 输入可用于启动减速斜坡或立即关闭 PWM 输出。这对于限位开关或某些电机故障检测非常实用。

有关 PWM 控制器的配置和使用的更多信息，请参阅节 7.5.7。

#### 7.3.10 CAN 收发器控制引脚

TCAN5102-Q1 没有内置 CAN 收发器，需要外部收发器。这允许用户选择他们想要支持的任何 CAN 收发器。该器件具有 CAN TXD、RXD 和 nSLP 引脚。nSLP 引脚用于在进入睡眠模式时控制外部收发器的睡眠或待机模式。

#### 7.3.11 欠压锁定与未供电器件

TCAN5102-Q1 会监测器件的电源轨 (VDD)。监测 VDD 是否存在欠压情况。当检测到 UVDD 故障时，器件会发生相应的状态变化。

表 7-2. VDD 故障与器件模式

VDD	器件模式
> UVDD	正常
< UVDD	上电复位

## 7.4 器件功能模式

TCAN5102-Q1 具有四种工作模式：未上电、初始化、睡眠和正常模式。有关不同器件模块在各模式下的表现，请参阅表 7-3。有关各工作模式之间的整体交互，请参阅图 7-3。

表 7-3. 模式概述

块	Init	正常	睡眠
CAN 收发器控制	关闭 (高阻态)	开启	关闭
CAN nSLP	关闭 (高阻态)	高电平 (VDD)	低电平 (GND)
SPI	关闭 (高阻态)	如果已编程，则开启	关闭
I <sup>2</sup> C	关闭 (高阻态)	如果已编程，则开启	关闭
UART	关闭 (高阻态)	如果已编程，则开启	关闭
GPIOx	关闭 (高阻态)	按编程	按编程

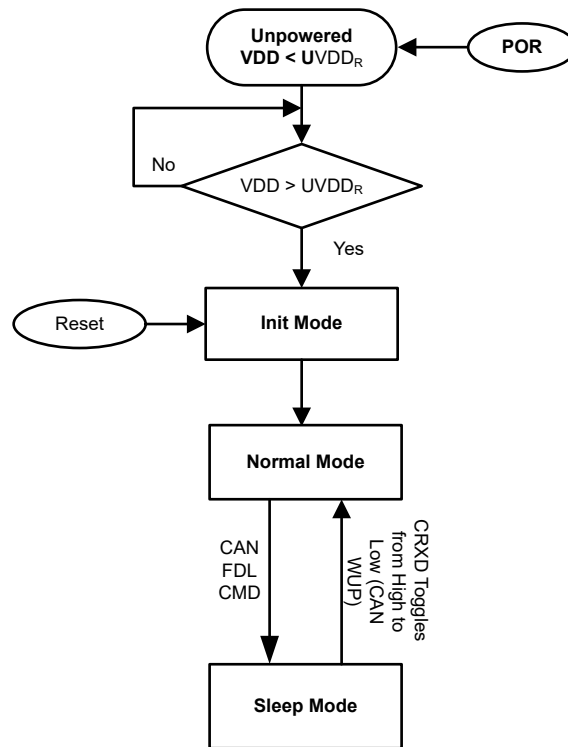


图 7-3. TCAN5102-Q1 的工作模式

### 7.4.1 初始化模式

这是器件上电后的初始操作模式。这是一种过渡模式：当 VDD 电压高于 UV<sub>DD</sub> 且已加载保存的器件配置后，器件进入该模式。配置加载完成后，器件将切换至正常模式。

在初始化模式下，器件会读取 EEPROM；EEPROM 读取完成后，配置数据将复制到寄存器中，器件随即切换至正常模式。有关保存哪些配置的更多信息，请参阅节 7.3.5。

### 7.4.2 睡眠模式

睡眠模式是器件的最低功耗状态。TCAN5102-Q1 从 CAN 总线接收进入睡眠模式命令，并执行以下操作：

- 执行任何内部维护操作
- 将外部 CAN 收发器置于睡眠模式

一旦 CAN 收发器进入睡眠模式，VDD 上预计不会通电。器件处于未通电状态。可以在睡眠模式下保持 VDD 通电。器件保持低功耗状态，直到从 CAN 收发器接收到 WUP 信号来唤醒器件。

### 7.4.3 正常模式

退出 INIT 模式后，器件将进入正常模式。正常模式下，收发器被激活，器件可准备接收或传输数据。

## 7.5 编程

该器件使用 CAN FD Light 接口进行配置。通过向器件上的寄存器写入数据来完成配置。存在一种特定的 CAN FD 消息结构，用于访问器件寄存器。该器件支持访问器件寄存器时使用多种帧格式。

### 7.5.1 通过 SPI 外设模式进行器件编程

配置编程有 2 种选项：

1. 通过 CAN FD Light 编程（使用 PROG 引脚将波特率强制为 1Mbps）
2. 使用 SPI（通过 PROG 和 RXD 引脚的切换序列启用本地 SPI 外设，允许编程器写入 EEPROM）

如果 CAN 接口不可用于编程 EEPROM，可使用 SPI 外设模式方法。此方法需要特定的事件序列才能启用本地 SPI 外设。当器件处于 SPI 外设模式时，CAN 接口被禁用，并保持禁用状态直至电源循环。

### 7.5.2 CAN FD Light 协议

TCAN5102-Q1 通过使用逻辑链路控制 (LLC) 和 FD 基本帧格式 (FBFF) 支持 CiA 604-1 和 CiA 604-3 通信标准，请参阅表 7-4。

**表 7-4. CiA 格式协议**

字段	尺寸	功能
标识符	11 位	基本标识符（目标或广播 ID）
DLC	4 位	数据长度码，如 ISO 11898-1:2015 表 5 中指定
LLC 数据	0 至 64 字节	数据帧的数据内容
由于所有发送和接收的帧均采用 FBFF 格式，且 ESI=0、BRS=0，因此格式字段的所有位均为固定值。因此，格式字段被省略。		

器件协议分为请求和响应两部分。器件请求是向该器件发出的对特定寄存器执行读取或写入操作的请求。请求始终至少有 3 个字段，这些字段是解码任何操作所必需的。这些字段始终位于 CAN 帧数据段的开头。

1. 操作码（Op 码）：这会告知器件是否正在对寄存器或 FIFO 进行读取或写入
2. 长度：指示正在传输的数据的总大小（以字节为单位）（在器件报头之后）
3. 寄存器地址：要写入或读取的寄存器或 FIFO 的 16 位地址

器件响应在请求之后发送，用于告知请求器件该请求已被接收且正在处理中或无效。响应至少包含 3 个字段

1. 操作码（Op 码）：回传接收到的 Op 码
2. 长度：指示报头之后返回的数据的总大小（以字节为单位）（如果请求是写操作，则该值与写入的字节数一致，但不发送数据。长度字段用于确认已写入的字节数）
3. 状态字节：指示请求的状态。这可能是正常/成功，或某种错误，如无效地址或长度。

器件使用 2 种类型的标识符。目标标识符是单个响应器唯一的特定地址。这使命令器能够对单个节点进行寻址/配置。同时，该器件还支持广播标识符功能，通过广播 ID 和掩码字段，可以匹配一个或多个 ID。来自广播 ID 的任

何请求都会被执行，但响应器不会返回响应，因为 CAN FD Light 不支持仲裁。如果命令器希望一次性配置多个节点或使多个节点进入休眠状态，广播会非常有用。但这样会丢失响应帧，因此无法得到响应器的确认。

#### 7.5.2.1 CAN 帧格式

标准 CAN 帧格式使用 CAN 有效载荷的 3 个字节作为报头，包含 3 个字段：操作码 ( op 码 )、数据长度和地址。

CAN ID		Byte 0	Byte 1	Byte 2	Byte 3	...
Device ID[10:0]	CTRL	OP[7:6]/ LEN[5:0]	ADDR[15:8]	ADDR[7:0]	Data	...

图 7-4. 标准 CAN 帧格式请求示例

表 7-5. 标准 CAN 帧格式请求报头字段

字节	位	字段	说明
0	7:6	操作码	器件的操作码 2'b00 = 写入 2'b01 = 读取 2'b10 = 保留 2'b11 = 保留
	5:0	LEN	在该报头之后传输的数据字节数 有效值 ( 写入 ) : 0 至 61 有效值 ( 读取 ) : 0 至 62
1	7:0	ADDR[15:8]	地址的高字节
2	7:0	ADDR[7:0]	地址的低字节

#### 备注

对于多字节读取，如果起始地址在某个串行通信块 ( I2C、SPI、UART ) 的 RX FIFO 之前，则实际上不会读取 FIFO。相反，对应于 RX FIFO 的字节返回 0，然后继续读取。这是为了防止突发读取在 RX FIFO 中被“卡住”。如果读取的起始地址从 RX FIFO 开始，则所有读取的字节都仅来自 RX FIFO。

CAN ID		Byte 0	Byte 1	Byte 2	...
Device ID[10:0]	CTRL	OP[7:6]/ LEN[5:0]	STATUS_BYTE	Data	...

图 7-5. 标准 CAN 帧格式响应示例

表 7-6. 标准 CAN 帧格式响应报头字段

字节	位	字段	说明
0	7:6	操作码	器件的操作码 2'b00 = 写入 2'b01 = 读取 2'b10 = 保留 2'b11 = 保留
	5:0	LEN	如果请求的数据量有效，则此字段将回显请求的、在报头后发送的数据字节数。  <b>备注</b> 若为针对写入请求的响应，则长度字段值等于请求的字节数，但报头后不附带任何数据。这是为了确认该响应帧与对应的写入请求相匹配
1	7:5	RSVD	保留

表 7-6. 标准 CAN 帧格式响应报头字段 (续)

字节	位	字段	说明
1	4	FE	上一个请求上出现组帧错误。请求被忽略。通常是由于 DLC 小于 3 字节所致，因为报头有最小长度要求
1	3	IO	无效操作。请求被忽略
1	2	IL	长度无效。请求被忽略
1	1	IA	地址无效。请求被忽略
1	0	OK	请求正常，请求已完成或正在进行中

### 7.5.3 消息 RAM (MRAM) 和 IP 使能

TCAN5102-Q1 具有 2kB 消息 RAM (MRAM)，用于存储 SPI、I2C 和 UART 通信的邮箱。该 RAM 可根据终端应用需求，按 25% 的增量比例，将内存灵活分配给任意组合的 2 个 IP 模块。RX 和 TX 之间的拆分比例不可调节，始终为 50/50%。RX 和 TX FIFO 的处理由 TCAN5102-Q1 完成，用户只需选择为每个 IP 模块分配多少 MRAM，无需其他设置。

SPI 和 I2C 的内存使用分配非常简单直接，若指定，分配内存的 50% 用于存储 TX 数据，另外 50% 就会用于存储 RX 数据。

对于 UART，需要额外字节存储每个接收字节的状态。这意味着 UART 每接收 1 字节数据，就需要额外 1 字节存储该字节的状态。因此，分配给 UART 的内存中，仅 2/3 用于存储实际数据字节。

表 7-7 显示了为每个 IP 模块的 TX 或 RX 缓冲区分配的字节数。所示大小对于 TX 和 RX 缓冲区相同，且并非两者共享的值。例如，如果 100% 的 MRAM 空间分配给 SPI，则 SPI TX 大小为 1024 字节，SPI RX 缓冲区为 1024 字节。TX 和 RX 缓冲区总和为 2048 字节，即全部内存。

在将 50% 的 MRAM 分配给 SPI 和 UART 的示例中，SPI RX 和 TX 大小各为 512 字节 (总共 1kB)。每个缓冲区的 UART IP 仅为 340 字节。由于 UART RX 需要额外字节存储每个接收字节的状态，因此在可用的 1024 字节中，仅 2/3 用于存储数据。在内部分配中，TX 和 RX 缓冲区可存储的字节数相同。因此，有 4 字节数据未被使用，因为剩余的 4 字节无法在 3 个功能 (TX、RX 和 RX 状态) 之间平均分配。

#### 备注

MRAM 分配设置用于启用 SPI 和 I2C 的 IP 模块。如果为该模块分配了内存，那么该模块会被使能，但模块所使用的 GPIO 仍需在 GPIO 配置寄存器中手动设置为特殊功能模式。

表 7-7. MRAM 分配

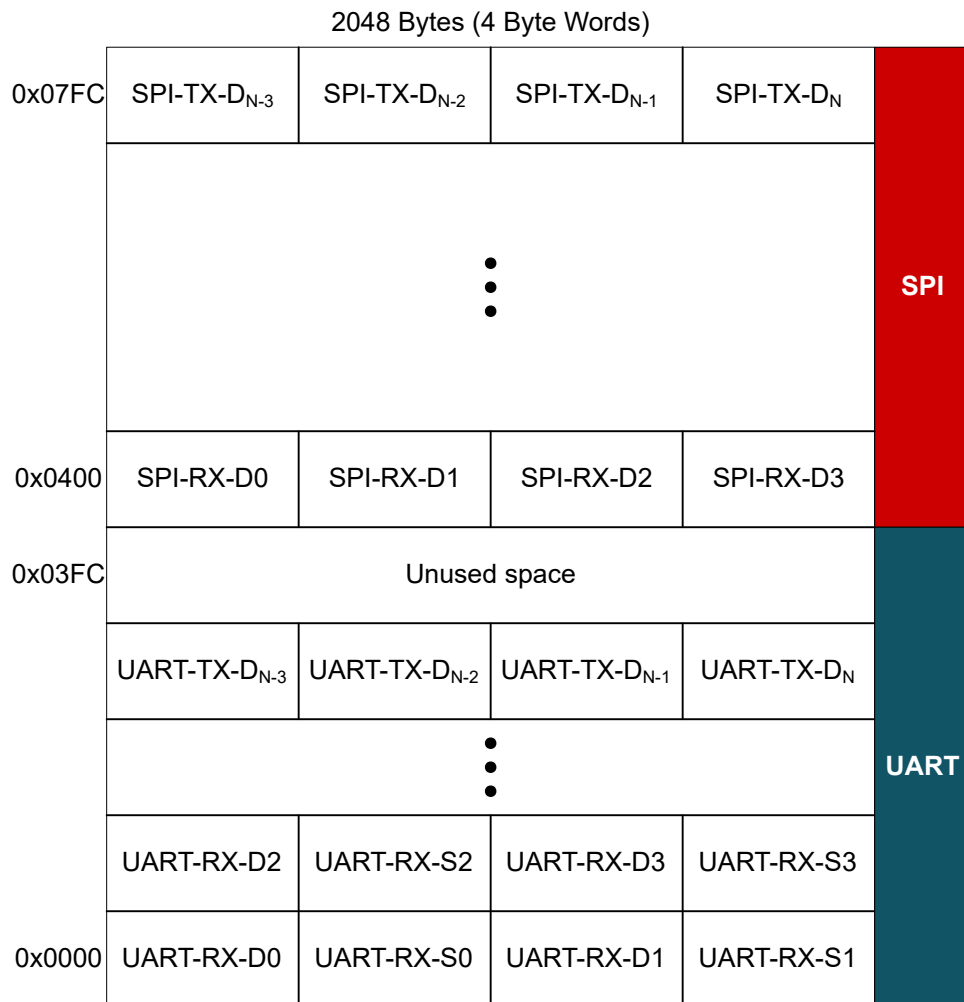
分配给 SPI 的百分比	SPI TX 或 RX 大小 (字节)	UART TX 或 RX 大小 (字节)	已使用的字节总数 (占总数的百分比)
100%	1024	0	2048 (100%)
75%	768	168	2040 (99.6%)
50%	512	340	2044 (99.8%)
25%	256	512	2048 (100%)
0%	0	680	2040 (99.6%)

表 7-8. MRAM 分配

MRAM_IP_EN (十六进制)	分配给 SPI 的百分比	分配给 UART 的百分比	分配给 I2C 的百分比	SPI TX 和 RX 大小 (字节)	UART TX 和 RX 大小 (字节)	I2C TX 和 RX 大小 (字节)	已使用的字节总数 (占总数的百分比)
0h	0%	0%	0%	0	0	0	0 (0%)
1h	0%	100%	0%	0	680	0	2040 (99.61%)
2h	25%	75%	0%	256	512	0	2048 (100%)
3h	50%	50%	0%	512	340	0	2044 (99.8%)

**表 7-8. MRAM 分配 ( 续 )**

MRAM_IP_EN ( 十六进制 )	分配给 SPI 的百分比	分配给 UART 的百分比	分配给 I2C 的百分比	SPI TX 和 RX 大小 ( 字节 )	UART TX 和 RX 大小 ( 字节 )	I2C TX 和 RX 大小 ( 字节 )	已使用的字节总数 ( 占总数的百分比 )
4h	75%	25%	0%	768	168	0	2040 (99.61%)
5h	100%	0%	0%	1024	0	0	2048 (100%)
6h	0%	0%	100%	0	0	1024	2048 (100%)
7h	0%	25%	75%	0	168	768	2040 (99.61%)
8h	0%	50%	50%	0	340	512	2044 (99.8%)
9h	0%	75%	25%	0	512	256	2048 (100%)
Ah	25%	0%	75%	256	0	768	2048 (100%)
Bh	50%	0%	50%	512	0	512	2048 (100%)
Ch	75%	0%	25%	768	0	256	2048 (100%)



**图 7-6. MRAM 布局示例 ( SPI 和 UART 启用 , 各占 50% )**



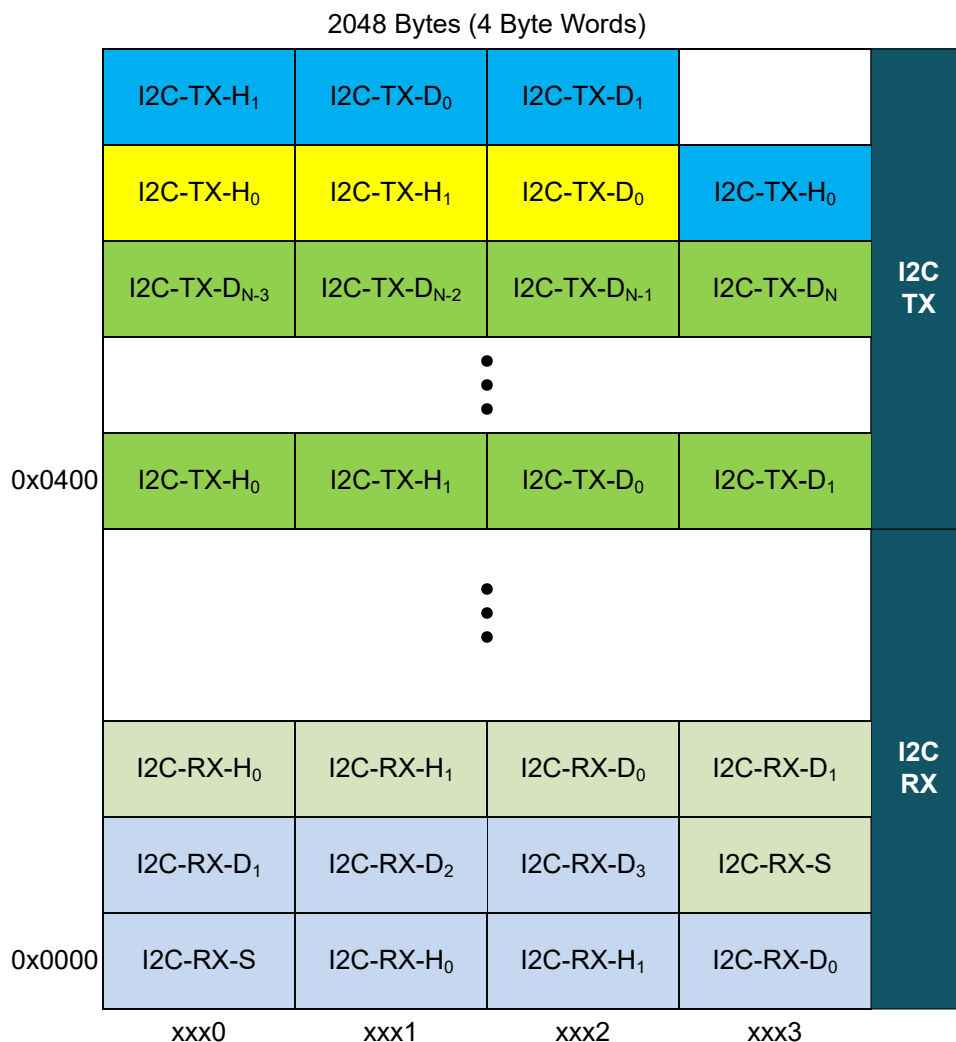


图 7-7. 带示例数据的 MRAM 布局示例 ( 100% MRAM 分配给 I2C )

上图展示了 MRAM 如何为 I2C 保存数据的示例。每个单独的 I2C 帧/消息以不同的颜色显示，以展示消息在每个 RX 或 TX FIFO 中的紧凑分组方式。用户无需了解数据在 FIFO 中的存储方式，只需知道 FIFO 的总大小以及报头/状态字节也存储在内存中这一事实即可。

#### 7.5.4 SPI 控制器

##### 备注

在使用该模块之前，必须将相应的 GPIO 设置为特殊功能（请参阅节 7.3.4 和节 7.6.1.18），并且 MRAM 必须通过分配内存启用模块（请参阅节 7.5.3 和节 7.6.1.17）。只有将 GPIO 设置为特殊功能，且通过选择分配内存的 MRAM 配置启用该模块后，方可使用该模块。相应的 SPI 配置寄存器可参阅节 7.6.2。

##### 7.5.4.1 SPI 引脚

SPI 通信通常使用 4 个引脚与另一 SPI 器件通信：时钟 (SCLK)、数据输入 (PICO)、数据输出 (POCI) 和片选 (CS 或 nCS)。



#### 7.5.4.1.1 SPI 时钟 (SCLK)

SPI 时钟输出引脚用于生成时钟，以同步移入或移出目标外设的数据。该引脚可配置为 4 种 SPI 模式中的任何一种（2 种时钟极性），以提供最大的可配置性和与下游外设的兼容性。

#### 7.5.4.1.2 外设输入/控制器输出 (PICO)

该输出引脚用于将数据传输到外设（下游器件）。当片选信号有效时（根据配置，可为高电平或低电平），PICO 将数据传输到目标/外设。此引脚将连接到外设/目标器件上的串行数据输入 (SDI) 引脚。

#### 7.5.4.1.3 外设输出/控制器输入 (POCI)

该输入引脚用于将数据从外设/下游器件移出并移入此器件/控制器。POCI 引脚通常连接到外设/目标器件的串行数据输出 (SDO) 引脚。

#### 7.5.4.1.4 片选 (CS 或 nCS)

片选输出引脚用于告知目标外设何时正在与其通信。该器件支持多达 4 个独立的片选通道，每个通道都有单独的极性选择。

#### 7.5.4.2 SPI 时钟发生器

TCAN5102-Q1 SPI 模块内置一个可编程时钟发生器，采用 8 位分频值。SPI\_DR 寄存器被解释为比 SPI\_DR 寄存器中的值大 1，用于半周期。分频值的计算公式如下：

$$\text{Divisor} = (\text{Clock Frequency} / (\text{Desired SPI clock frequency} \times 2)) - 1 \quad (1)$$

其中时钟频率为 40MHz。由该公式可知，支持的最大 SPI 时钟频率为 20MHz，最小 SPI 时钟频率为 78,125Hz。

**表 7-9. 常见 SPI 时钟**

期望 SPI 时钟频率	用于生成时钟的分频值	SPI_DR 值 (十六进制)
80kHz	249	0xF9
100kHz	199	0xC7
200kHz	99	0x63
250kHz	79	0x4F
500kHz	39	0x27
1MHz	19	0x13
2MHz	9	0x09
4MHz	4	0x04
5MHz	3	0x03
6.67MHz	2	0x02
10MHz	1	0x01
20MHz	0	0x00

#### 7.5.4.3 SPI 控制协议

向 SPI TX 或 RX FIFO 读取或写入时，有一个额外的 2 字节报头，在向目标外设读取或写入时提供更高的稳健性。有 2 种格式，它们相似但略有不同，具体取决于对 SPI TX 或 RX FIFO 执行的是读取还是写入操作。

读取 RX FIFO 时，数据字段的开头总有一个 2 字节报头，包含剩余字节数以及该读取是否是前一次读取的延续（对于大于单个 CAN 帧所能容纳的 SPI 消息而言是必需的）。

对 TX FIFO 进行写入时，只有消息的第一次写入才包含 2 字节报头。

有关报头格式的更多信息，请参阅 [SPI 接收 FIFO \(地址 = h1010\)](#) 和 [SPI 发送 FIFO \(地址 = h1010\)](#)。

#### 7.5.4.3.1 SPI 写入示例 1

这是一个通过 SPI 发送可容纳于单个 CAN 帧的字节流的基础示例。一旦器件接收到 SPI 帧的所有字节 ( 由 SPI 报头中指定的字节数确定 ) 后, 该器件将开始发送该 SPI 帧。

##### 备注

在使用该模块之前, 必须将相应的 GPIO 设置为特殊功能 ( 请参阅节 7.3.4 和节 7.6.1.18 ), 并且 MRAM 必须通过分配内存启用模块 ( 请参阅节 7.5.3 和节 7.6.1.17 )。只有将 GPIO 设置为特殊功能, 且通过选择分配内存的 MRAM 配置启用该模块后, 方可使用该模块。相应的 SPI 配置寄存器可参阅节 7.6.2。

表 7-10. 设计参数和假设

参数	值
CAN 帧格式	标准帧格式
要传输的字节	10'B
SPI 通道	通道 1
将接收到的数据保存到 RX FIFO	否
SPI 字节流	0x11、0x22、0x33、0x44、0x55、0x66、0x77、0x88、0x99、0xAA

下表显示了在 CAN 总线上传输的步骤和数据。总线的中央网关/控制器称为 ECU, TCAN 器件称为“响应器”。

- 标记为**粗体**的十六进制值表示 CAN 帧头的字节 ( 本例中为标准帧格式 )
- 标记为**斜体**的十六进制值表示 SPI 报头字节
- 无格式的十六进制值为 SPI 数据字节

表 7-11. CAN 序列示例 ( 单个 CAN 帧 )

步骤	发送器	数据	说明
1	ECU (DLC = 16B)	<b>0x0C</b> , <b>0x10</b> , <b>0x10</b> , 0x01, 0x0A, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA, 0x00	ECU 请求将 10 字节写入寄存器 0x1010 (SPI TX FIFO), 其中 SPI 报头的存储 = 0、SPI 通道 = 1 以及 SPI 帧长度为 10 字节。随后将 10 字节数据移入设备。由于 CAN DLC 必须设为 16 字节来容纳这 15 字节数据, 因此在末尾填充 1 字节 0x00。
2	响应器	<b>0x0C</b> , <b>0x01</b>	响应器发回一个 OK 以确认已收到请求。
3	SPI	PICO : 0x11、0x22、0x33、0x44、0x55、0x66、0x77、0x88、0x99、0xAA	此时, 响应器在 SPI 通道 1 上执行 10 字节的 SPI 传输。

表 7-12. 示例 CAN 序列 ( 多个 CAN 帧到传输有效载荷 )

步骤	发送器	数据	说明
1	ECU (DLC = 8B)	<b>0x05</b> , <b>0x10</b> , <b>0x10</b> , 0x01, 0x0A, 0x11, 0x22, 0x33	ECU 请求将 5 字节写入寄存器 0x1010 (SPI TX FIFO), 其中 SPI 报头的存储 = 0、SPI 通道 = 1 以及 SPI 帧长度为 10 字节。仅传输 SPI 帧的前 3 个字节数据。
2	响应器	<b>0x05</b> , <b>0x01</b>	响应器发回一个 OK 以确认已收到请求。
3	ECU (DLC = 3B)	<b>0x42</b> , <b>0x10</b> , <b>0x12</b>	ECU 请求读取 2 字节数据, 以读取 TX FIFO 状态寄存器和 TX 元素状态寄存器。此步骤为可选操作, 但能展示一些关于部分写入的有用信息。
4	响应器 (DLC = 4B)	<b>0x42</b> , <b>0x01</b> , 0x0C, 0x87	SPI_TXFS 显示 TX FIFO 中有 12 个空闲空间 ( 字节 ), 因此无溢出风险。SPI_TXES 如预期所示设置了 TXEIP ( 发送元素进行中 ) 标志, 表示当前 SPI 帧在内存中不完整, 还需要 7 字节数据 ( TXEBP 字段 )

表 7-12. 示例 CAN 序列 ( 多个 CAN 帧到传输有效载荷 ) ( 续 )

步骤	发送器	数据	说明
5	ECU (DLC = 12B)	<b>0x07, 0x10, 0x10</b> , 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA, 0x00, 0x00	ECU 继续向同一地址发起 7 字节写入请求。由于能容纳该帧的 CAN DLC 为 12 字节, 因此在末尾填充 2 字节。填充字节的值无实际意义, 会被忽略。
6	响应器	<b>0x07, 0x01</b>	响应器发回一个 OK 以确认已收到请求。
7	SPI	PICO : 0x11、0x22、0x33、0x44、0x55、0x66、0x77、0x88、0x99、0xAA	此时, 响应器在 SPI 通道 1 上执行 10 字节的 SPI 传输。POCI 被忽略。

#### 7.5.4.3.2 SPI 读取示例 1

这是一个通过 SPI 发送和接收字节流的基础示例。一旦器件接收到 SPI 帧的所有字节 ( 由 SPI 报头中指定的字节数确定 ) 后, 该器件将开始发送该 SPI 帧。本示例会存储接收到的 POCI 数据, 以供后续回读。

#### 备注

在使用该模块之前, 必须将相应的 GPIO 设置为特殊功能 ( 请参阅节 7.3.4 和节 7.6.1.18 ), 并且 MRAM 必须通过分配内存启用模块 ( 请参阅节 7.5.3 和节 7.6.1.17 )。只有将 GPIO 设置为特殊功能, 且通过选择分配内存的 MRAM 配置启用该模块后, 方可使用该模块。相应的 SPI 配置寄存器可参阅节 7.6.2。

表 7-13. 设计参数和假设

参数	值
CAN 帧格式	标准帧格式
要传输的字节	10B
SPI 通道	通道 2
将接收到的数据保存到 RX FIFO	是
SPI 字节流	0x11、0x22、0x33、0x44、0x55、0x66、0x77、0x88、0x99、0xAA

下表显示了在 CAN 总线上传输的步骤和数据。总线的中央网关/控制器称为 ECU, TCAN 器件称为“响应器”。

- 标记为**粗体**的十六进制值表示 CAN 帧头的字节 ( 本例中为标准帧格式 )
- 标记为**斜体**的十六进制值表示 SPI 报头字节
- 无格式的十六进制值为 SPI 数据字节

表 7-14. CAN 序列示例 ( 单个 CAN 帧 )

步骤	发送器	数据	说明
1	ECU (DLC = 16B)	<b>0x0C, 0x10, 0x10</b> , 0x82, 0x0A, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA, 0x00	ECU 请求将 10 字节写入寄存器 0x1010 (SPI TX FIFO), 其中 SPI 报头的存储 = 1、SPI 通道 = 2 以及 SPI 帧长度为 10 字节。随后将 10 字节数据移入设备。由于 CAN DLC 必须设为 16 字节来容纳这 15 字节数据, 因此在末尾填充 1 字节 0x00。
2	响应器	<b>0x0C, 0x01</b>	响应器发回一个 OK 以确认已收到请求。
3	SPI	PICO : 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA POCI : 0x01、0x02、0x03、0x04、0x05、0x06、0x07、0x08、0x09、0x0A	此时, 响应器在 SPI 通道 2 上执行 10 字节的 SPI 传输。POCI 数据存储在 RX FIFO 中
4	ECU	<b>0x43, 0x10, 0x0F</b>	ECU 请求从寄存器 0x100F-0x1011 ( SPI_STATUS 和 SPI_RXFS ) 读取 3 字节数据。由于突发读取并非从 0x1010 (SPI RX FIFO) 开始, 因此对应该寄存器的字节将填充为 0。对于未从 FIFO 开始的突发读取, 其对应 FIFO 的字节将返回“0”。

表 7-14. CAN 序列示例 ( 单个 CAN 帧 ) ( 续 )

步骤	发送器	数据	说明
5	响应器	<b>0x43, 0x01</b> , 0xCA, 0x00, 0x01	响应器返回寄存器 0x100F 的数据、寄存器 0x1010 的填充值 0 以及寄存器 0x1011 的数据。返回数据表明：TX FIFO 为空，RX FIFO 包含 1 个 SPI 帧，且 RX FIFO 的下一个元素包含 10 字节数据。ECU 根据该信息，将请求从器件读取 10 + 2 ( SPI 报头 ) 字节数据。
6	ECU	<b>0x4C</b> 、 <b>0x10</b> 、 <b>0x10</b>	ECU 请求从 RX FIFO 读取 12 字节 ( 10 字节数据 + 2 字节报头 )
7	响应器 (DLC = 16B)	<b>0x4C, 0x01</b> , 0x02, 0x0A, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x00, 0x00	响应器返回请求的 12 字节数据。SPI 报头字节表明：此为帧读取的起始，使用的是 SPI 通道 2，待读取的 SPI 数据为 10 字节 ( 可容纳于整个 CAN 帧中 )。由于能容纳该数据的最近 CAN DLC 为 16 字节，因此在末尾填充 2 字节数据。

表 7-15. 示例 CAN 序列 ( 多个 CAN 帧到传输有效载荷 )

步骤	发送器	数据	说明
1	ECU (DLC = 8B)	<b>0x05, 0x10, 0x10</b> , 0x02, 0x0A, 0x11, 0x22, 0x33	ECU 请求将 5 字节写入寄存器 0x1010 (SPI TX FIFO)，其中 SPI 报头的存储 = 1、SPI 通道 = 2 以及 SPI 帧长度为 10 字节。仅传输 SPI 帧的前 3 个字节数据。
2	响应器 (DLC = 2B)	<b>0x05, 0x01</b>	响应器发回一个 OK 以确认已收到请求。
3	ECU (DLC = 12B)	<b>0x07, 0x10, 0x10</b> , 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA, 0x00, 0x00	ECU 继续向同一地址发起 7 字节写入请求。由于能容纳该帧的 CAN DLC 为 12 字节，因此在末尾填充 2 字节。填充字节的值无实际意义，会被忽略。
4	响应器 (DLC = 2B)	<b>0x07, 0x01</b>	响应器发回一个 OK 以确认已收到请求。
3	SPI	PICO : 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77, 0x88, 0x99, 0xAA POCI : 0x01、0x02、0x03、0x04、0x05、0x06、0x07、0x08、0x09、0x0A	此时，响应器在 SPI 通道 2 上执行 10 字节的 SPI 传输。POCI 数据存储在 RX FIFO 中
4	ECU (DLC = 3B)	<b>0x43, 0x10, 0x0F</b>	ECU 请求从寄存器 0x100F-0x1011 ( SPI_STATUS 和 SPI_RXFS ) 读取 3 字节数据。由于突发读取并非从 0x1010 (SPI RX FIFO) 开始，因此对应该寄存器的字节将填充为 0。对于未从 FIFO 开始的突发读取，其对应 FIFO 的字节将返回 “0”。
5	响应器 (DLC = 5B)	<b>0x43, 0x01</b> , 0xCA, 0x00, 0x01	响应器返回寄存器 0x100F 的数据、寄存器 0x1010 的填充值 0 以及寄存器 0x1011 的数据。返回数据表明：TX FIFO 为空，RX FIFO 包含 1 个 SPI 帧，且 RX FIFO 的下一个元素包含 10 字节数据。ECU 根据该信息，将请求从器件读取 10 + 2 ( SPI 报头 ) 字节数据。
6	ECU (DLC = 3B)	<b>0x48, 0x10, 0x10</b>	ECU 请求从 RX FIFO 读取 8 字节数据 ( 含 2 字节报头 + 6 字节数据 )
7	响应器 (DLC = 12B)	<b>0x48, 0x01</b> , 0x02, 0x0A, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x00, 0x00	响应器返回请求的 8 字节数据 ( 尽管 DLC 为 12，但我们知道，CAN 响应报头后的 8 字节才是有效数据，其余的 0 均为填充字节 )。SPI 报头字节表明：此为帧读取的起始，使用的是 SPI 通道 2，尚有 10 字节 SPI 数据待读取。由于能容纳该数据的最近 CAN DLC 为 12 字节，因此在末尾填充 2 字节数据。ECU 已获知有 10 字节 SPI 数据待读取，但仅请求读取 6 字节。这意味着尚有 4 字节 SPI 数据待读取。
8	ECU (DLC = 3B)	<b>0x46, 0x10, 0x10</b>	ECU 请求从 RX FIFO 读取 6 字节数据 ( 4 字节数据 + 2 字节报头 )，以完成读取操作。
9	响应器 (DLC = 8B)	<b>0x46, 0x01</b> , 0x82, 0x04, 0x07, 0x08, 0x09, 0x0A	响应器返回请求的 6 字节数据。SPI 报头字节表明：此为帧读取的延续，使用的是 SPI 通道 2，尚有 4 字节 SPI 数据待读取 ( 可容纳于本帧中 )。由于待发送数据与 CAN 帧大小匹配，因此无需填充字节。

## 7.5.5 UART 控制器

### 备注

在使用该模块之前，必须将相应的 GPIO 设置为特殊功能（请参阅节 7.3.4 和节 7.6.1.18），并且 MRAM 必须通过分配内存启用模块（请参阅节 7.5.3 和节 7.6.1.17）。只有将 GPIO 设置为特殊功能，且通过选择分配内存的 MRAM 配置启用该模块后，方可使用该模块。相应的 UART 配置寄存器可参阅节 7.6.4。

### 7.5.5.1 UART 波特率生成和分数分频值

TCAN5102-Q1 的 UART 模块配备了一个可编程波特率发生器，其核心功能是将基准时钟按 1 至  $(2^{16} - 1)$  范围内的分频值进行分频，且分频精度可达 1/64 的小数分辨率。波特率发生器的输出频率为波特率的 16 倍。波特率发生器的输入时钟固定为 40MHz。分频值的计算公式如下：

$$\text{Divisor} = (40\text{E6} / (\text{Desired baud rate} \times 16)) \quad (2)$$

若要计算需写入寄存器的具体数值，则需借助以下函数：

- TRUNC(X)：对 X 取整，仅返回实数的整数部分。例如：TRUNC(3.14) = 3
- ROUND(X)：将 X 舍入到最接近的整数。例如：ROUND(3.1) = 3 和 ROUND(3.6) = 4
- >>：右移位操作。例如：0x1000 >> 8 = 0x0010。或者 0b0001 0000 0000 0000 >> 8 = 0b0000 0000 0001 0000
- &：按位与函数，用于屏蔽位。例如：0x1234 & 0x00FF = 0x0034 以及 0x8765 & 0xFF00 = 0x8700

一旦找到所需的分频值，就可以通过以下方式计算寄存器值：

$$\text{UART\_BR\_MSB} = \text{TRUNC}(\text{Divisor}) \gg 8 \quad (3)$$

$$\text{UART\_BR\_LSB} = \text{TRUNC}(\text{Divisor}) \& 0xFF \quad (4)$$

$$\text{UART\_BR\_FRAC} = \text{ROUND}([\text{Divisor} - \text{TRUNC}(\text{Divisor})] \times 64) \quad (5)$$

表 7-16. 使用 16 倍波特率分频值的波特率 (40MHz 时钟)

所需波特率	用于生成 16 倍时钟的分频值	可获得的最接近的分频值	UART_BR_MSB 值 (十六进制)	UART_BR_LSB 值 (十六进制)	UART_BR_FRAC 值 (十六进制)	期望值与实际值之间的误差 差异百分比 (%)
400	6250.0000	6250	0x18	0x6A	0x00	0
2400	1041.6667	1041 43/64	0x04	0x11	0x2B	0
4800	520.8333	520 53/64	0x02	0x08	0x35	0.001
9600	260.4167	260 27/64	0x01	0x04	0x1B	0.002
10000	250.0000	250	0x00	0xFA	0x00	0
19200	130.2083	130 13/64	0x00	0x82	0x0D	0.004
25000	100.0000	100	0x00	0x64	0x00	0
28800	86.8056	86 52/64	0x00	0x56	0x34	0.008
38400	65.1042	65 7/64	0x00	0x41	0x07	0.008
50000	50.0000	50	0x00	0x32	0x00	0
57600	43.4028	43 26/64	0x00	0x2B	0x1A	0.008
75000	33.3333	33 21/64	0x00	0x21	0x15	0.016
100000	25.0000	25	0x00	0x19	0x00	0
115200	21.7014	21 45/64	0x00	0x15	0x2D	0.008
128000	19.5313	19 34/64	0x00	0x13	0x22	0
153600	16.2760	16 18/64	0x00	0x10	0x12	0.032

表 7-16. 使用 16 倍波特率分频值的波特率 ( 40MHz 时钟 ) ( 续 )

所需波特率	用于生成 16 倍时钟的分频值	可获得的最接近的分频值	UART_BR_MSB 值 ( 十六进制 )	UART_BR_LSB 值 ( 十六进制 )	UART_BR_FRAC 值 ( 十六进制 )	期望值与实际值之间的误差 差异百分比 (%)
200000	12.5000	12 32/64	0x00	0x0C	0x20	0
225000	11.1111	11 7/64	0x00	0x0B	0x07	0.016
230400	10.8507	10 54/64	0x00	0x0A	0x36	0.064
250000	10.0000	10	0x00	0x0A	0x00	0
256000	9.7656	9 49/64	0x00	0x09	0x31	0
300000	8.3333	8 21/64	0x00	0x08	0x15	0.063
400000	6.2500	6 16/64	0x00	0x06	0x10	0
460800	5.4253	5 27/64	0x00	0x05	0x1B	0.064
500000	5.0000	5	0x00	0x05	0x00	0
750000	3.3333	3 21/64	0x00	0x03	0x15	0.156
921600	2.7127	2 46/64	0x00	0x02	0x2E	0.224
1000000	2.5000	2 32/64	0x00	0x02	0x20	0

### 7.5.5.2 UART 控制协议

由于 UART 是基于字节的传输，没有更高级别的协议，因此不存在帧的概念。UART 模块移入和移出字节时，会对每个字节进行错误检查。因此，每个接收到的字节都有一个相关联的状态字节，用于向 MCU 提示任何错误。由于错误并非经常发生，因此除非发生错误，否则无需读取每个字节的状态字节。为节省吞吐量，每次读取 UART RX FIFO 的末尾都会附加一个全局状态字节。此全局状态字节是已读出字节的所有状态字节的逻辑“或”结果。这使 MCU 能够确定所传输的字节块是否有任何错误报告。如果有，可以请求读取。读取 RX 错误状态寄存器以确定哪个字节出错以及所接收的错误。

发送或接收数据时，没有使用额外的 UART 特定报头，仅在读取 RX FIFO 时有全局状态字节。

有关寄存器和数据格式的更多信息，请参阅 [UART 发送 FIFO \( 地址 = h2010 \)](#)、[UART 发送 FIFO \( 地址 = h2010 \)](#) 和 [UART 接收错误状态 \( 地址 = h2011 \)](#)。

#### 7.5.5.2.1 UART 写入示例 1

这是一个通过 UART 发送字节流的基础示例。一旦器件在 TX FIFO 中接收到任何字节后，该器件将开始发送 UART 字节。

#### 备注

在使用该模块之前，必须将相应的 GPIO 设置为特殊功能 ( 请参阅 [节 7.3.4](#) 和 [节 7.6.1.18](#) )，并且 MRAM 必须通过分配内存启用模块 ( 请参阅 [节 7.5.3](#) 和 [节 7.6.1.17](#) )。只有将 GPIO 设置为特殊功能，且通过选择分配内存的 MRAM 配置启用该模块后，方可使用该模块。相应的 UART 配置寄存器可参阅 [节 7.6.4](#)。

表 7-17. 设计参数和假设

参数	值
CAN 帧格式	标准帧格式
要传输的字节	6B
UART 的 TX 字节流	0x11、0x22、0x33、0x44、0x55、0x66

下表显示了在 CAN 总线上传输的步骤和数据。总线的中央网关/控制器称为 ECU，TCAN 器件称为“响应器”。

- 标记为**粗体**的十六进制值表示 CAN 帧头的字节 ( 本例中为标准帧格式 )
- 无格式的十六进制值为 UART 数据字节



**表 7-18. CAN 序列示例 ( 单个 CAN 帧 )**

步骤	发送器	数据	说明
1	ECU (DLC = 12B)	<b>0x06, 0x20, 0x10</b> , 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x00, 0x00, 0x00	ECU 请求向寄存器 0x2010 (UART TX FIFO) 写入 6 字节, 由于 CAN DLC 必须为 12 字节才能容纳这 6 字节数据, 因此在末尾填充 0x00。
2	响应器	<b>0x06, 0x01</b>	响应器发回一个 OK 以确认已收到请求。
3	UART	TXD : 0x11、0x22、0x33、0x44、0x55、0x66	此时, 响应器进行 6 字节的 UART 传输。

**表 7-19. 示例 CAN 序列 ( 多个 CAN 帧到传输有效载荷 )**

步骤	发送器	数据	说明
1	ECU (DLC = 6B)	<b>0x03, 0x20, 0x10</b> , 0x11, 0x22, 0x33	ECU 请求向寄存器 0x2010 (UART TX FIFO) 写入 3 字节。
2	响应器	<b>0x03, 0x01</b>	响应器发回一个 OK 以确认已收到请求。响应器也在此处开始通过 UART 传输数据, 但为简化起见, 此表显示所有字节稍后传输。
3	ECU (DLC = 6B)	<b>0x03, 0x20, 0x10</b> , 0x44, 0x55, 0x66	ECU 请求向寄存器 0x2010 (UART TX FIFO) 写入 3 字节以完成剩余数据
4	响应器	<b>0x03, 0x01</b>	响应器发回一个 OK 以确认已收到请求。
5	UART	TXD : 0x11、0x22、0x33、0x44、0x55、0x66	此时, 响应器进行 6 字节的 UART 传输。

#### 7.5.5.2.2 UART 读取示例 1

本示例为通过 UART 发送和接收字节流的基础示例。一旦器件在 TX FIFO 中接收到任何字节后, 将开始发送 UART 字节。

#### 备注

在使用该模块之前, 必须将相应的 GPIO 设置为特殊功能 ( 请参阅节 7.3.4 和节 7.6.1.18 ), 并且 MRAM 必须通过分配内存启用模块 ( 请参阅节 7.5.3 和节 7.6.1.17 )。只有将 GPIO 设置为特殊功能, 且通过选择分配内存的 MRAM 配置启用该模块后, 方可使用该模块。相应的 UART 配置寄存器可参阅节 7.6.4。

**表 7-20. 设计参数和假设**

参数	值
CAN 帧格式	标准帧格式
要传输的字节	6B
UART 的 RX 字节流	0x11、0x22、0x33、0x44、0x55、0x66
错误	字节 0x33 存在奇偶校验错误, 字节 0x44 存在帧错误

下表列出了操作步骤及 CAN 总线上传输的数据。总线的中央网关/控制器称为 ECU, TCAN 器件称为“响应器”。

- 标记为**粗体的**十六进制值表示 CAN 帧头的字节 ( 本例中为标准帧格式 )
- 无格式的十六进制值为 UART 数据字节

**表 7-21. CAN 序列示例 ( 单个 CAN 帧 )**

步骤	发送器	数据	说明
1	UART	RXD : 0x11、0x22、0x33、0x44、0x55、0x66	UART RXD 将接收 6 字节
2	ECU (DLC = 16B)	<b>0x41, 0x20, 0x0B</b>	ECU 请求读取 0x200B ( UART RX FIFO 状态 )

表 7-21. CAN 序列示例 ( 单个 CAN 帧 ) ( 续 )

步骤	发送器	数据	说明
3	响应器	<b>0x41, 0x01</b> , 0x06	响应器发回 OK 以确认已接收请求并发送寄存器数据, 表明当前 RX FIFO 中有 6 字节数据
4	ECU	<b>0x47, 0x20, 0x10</b>	ECU 请求读取 7 字节 ( 6 字节数据 + 1 字节全局状态 ) 至 0x2010 (UART RX FIFO)
5	响应器	<b>0x47, 0x01</b> , 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x0C, 0x00, 0x00, 0x00	响应器回复 7 字节数据 ( 6 字节 UART 数据 + 1 字节全局状态 )。全局状态字节表明, 在刚读取的 6 字节数据中, 至少有 1 字节同时存在奇偶校验错误和帧错误。
6	ECU	<b>0x46, 0x20, 0x11</b>	由于 ECU 检测到非正常的全局状态字节, 因此请求读取 0x2011 ( UART RX 错误状态 ) 的 6 字节数据 ( 无全局状态字节 )。
7	响应器	<b>0x46, 0x01</b> , 0x01, 0x01, 0x04, 0x08, 0x01, 0x01	响应器发送 6 字节状态数据, 表明 0x33 存在奇偶校验错误, 0x44 存在帧错误。ECU 可以决定如何处理该错误。

表 7-22. 示例 CAN 序列 ( 多个 CAN 帧到传输有效载荷 )

步骤	发送器	数据	说明
1	UART	RXD : 0x11, 0x22, 0x33, 0x44, 0x55, 0x66	UART RXD 将接收 6 字节
2	ECU	<b>0x41, 0x20, 0x0B</b>	ECU 请求读取 0x200B ( UART RX FIFO 状态 )
3	响应器	<b>0x41, 0x01</b> , 0x06	响应器发回 OK 以确认已接收请求并发送寄存器数据, 表明当前 RX FIFO 中有 6 字节数据
4	ECU	<b>0x43, 0x20, 0x10</b>	ECU 请求读取 3 字节 ( 2 字节数据 + 1 字节全局状态 ) 至 0x2010 (UART RX FIFO)
5	响应器	<b>0x43, 0x01</b> , 0x11, 0x22, 0x01	响应器回复 3 字节数据 ( 2 字节 UART 数据 + 1 字节全局状态 )。全局状态字节表示没有报告错误 ( 正常状态 )。
6	ECU	<b>0x41, 0x20, 0x0B</b>	ECU 请求读取 0x200B ( UART RX FIFO 状态 )
7	响应器	<b>0x41, 0x01</b> , 0x04	响应器发回 OK 以确认已接收请求并发送寄存器数据, 表明当前 RX FIFO 中有 4 字节数据
8	ECU	<b>0x45, 0x20, 0x10</b>	ECU 请求读取 5 字节 ( 4 字节数据 + 1 字节全局状态 ) 至 0x2010 (UART RX FIFO)
9	响应器	<b>0x45, 0x01</b> , 0x33, 0x44, 0x55, 0x66, 0x0C	响应器回复 3 字节数据 ( 2 字节 UART 数据 + 1 字节全局状态 )。全局状态字节表明, 在刚读取的 4 字节数据中, 至少有 1 字节同时存在奇偶校验错误和帧错误。
10	ECU	<b>0x44, 0x20, 0x11</b>	由于 ECU 检测到非正常的全局状态字节, 因此请求读取 0x2011 ( UART RX 错误状态 ) 的 4 字节数据 ( 无全局状态字节 )。
11	响应器	<b>0x44, 0x01</b> , 0x04, 0x08, 0x01, 0x01	响应器发送 4 字节状态数据, 表明 0x33 存在奇偶校验错误, 0x44 存在帧错误。ECU 可以决定如何处理该错误。



## 7.5.6 I2C 控制器

### 备注

在使用该模块之前，必须将相应的 GPIO 设置为特殊功能（请参阅节 7.3.4 和节 7.6.1.18），并且 MRAM 必须通过分配内存启用模块（请参阅节 7.5.3 和节 7.6.1.17）。只有将 GPIO 设置为特殊功能，且通过选择分配内存的 MRAM 配置启用该模块后，方可使用该模块。相应的 I2C 配置寄存器可参阅节 7.6.6。

### 7.5.6.1 I2C 波特率生成

TCAN5102-Q1 I2C 模块包含一个可编程波特率发生器，可将参考时钟除以 1 至 255 范围内的分频值。分频值的输入时钟可以是 10MHz 或 2.5MHz，由 I2C\_CTRL.LSM（低速模式）位设置。分频值的计算公式如下：

$$\text{Divisor} = (\text{Input Clock} / \text{Desired I2C Speed}) - 1 \quad (6)$$

表 7-23. 使用高速时钟的 I2C 速度（10MHz 时钟，I2C\_CTRL.LSM = 0）

期望的 I2C 速度 (kHz)	可获得的最接近的分频值	I2C_BR 值 (十六进制)	实际 I2C 速度 (kHz)	期望值与实际值之间的误差差异百分比 (%)
39	255	0xFF	39	0.16
50	199	0xC7	50	0
60	166	0xA6	60	0.2
70	142	0x8E	70	0.1
80	124	0x7C	80	0
100	99	0x63	100	0
200	49	0x31	200	0
250	39	0x27	250	0
400	24	0x18	400	0
500	19	0x13	500	0
600	16	0x10	588	2
800	12	0x0C	769	4
1000	9	0x09	1000	0

表 7-24. 使用低速时钟的 I2C 速度（2.5MHz 时钟，I2C\_CTRL.LSM = 1）

期望的 I2C 速度 (kHz)	可获得的最接近的分频值	I2C_BR 值 (十六进制)	实际 I2C 速度 (kHz)	期望值与实际值之间的误差差异百分比 (%)
10	249	0xF9	10	0
50	49	0x31	50	0
60	41	0x29	60	0.8
70	35	0x23	69	0.8
80	30	0x1E	81	0.8
100	24	0x18	100	0
200	12	0x0C	192	4
250	9	0x09	250	0
400	5	0x05	417	4
500	4	0x04	500	0
600	3	0x03	625	4
800	2	0x02	833	4

### 7.5.6.2 I2C 总线卡滞恢复

I2C 控制器具有一项可选功能，称为总线卡滞恢复 (SBR)。如果使能，一旦检测到总线卡滞条件，IP 会自动尝试解除总线卡滞状态。总线卡滞的检测条件是 SCL 或 SDA 线保持低电平的时间达到  $t_{\text{STUCKBUS\_I2C}}$ 。一旦检测到卡滞条件且已启动自动 SBR，I2C 控制器会尝试生成最多 16 个时钟脉冲。在生成时钟脉冲时，控制器会监控卡滞条件是否已解决。一旦该情况解决，时钟生成会立即停止，并生成一个 STOP 条件以复位总线。总线卡滞恢复序列完成后，会置位 SBRC (总线卡滞恢复完成) 中断。可通过 I2C\_STATUS 寄存器轮询 I2C 总线的状态。通过向 I2C\_CTRL 寄存器的 SBR\_START 位写入，可手动请求总线卡滞恢复序列。

#### 备注

一旦 I2C IP 使能并连接到 GPIO，总线卡滞检测计时器便会启动。请注意，在使能 I2C IP 之前，确保 I2C 总线上有上拉电阻，否则可能会出现虚假的总线卡滞条件

### 7.5.6.3 I2C 控制协议

向 I2C 发送 (TX) FIFO 写入时，需要一个 2 字节的报头，包含 I2C 地址和数据字节数。从 I2C 接收 (RX) FIFO 读取时，有 3 个额外的报头字节，可在从目标外设读取时提供更高的稳健性。有 2 种报头格式，它们相似但略有不同，具体取决于对 TX 或 RX FIFO 执行的是读取还是写入操作。

向 TX FIFO 写入时，2 字节报头是写入 FIFO 的前 2 个字节。

读取 RX FIFO 时，数据字段的开头总有一个 2 字节报头，包含剩余字节数以及该读取是否是前一次读取的延续 (对于大于单个 CAN 帧所能容纳的 SPI 消息而言是必需的)。

如果多个 I2C 帧在 TX FIFO 中排队，IP 将使用重复起始位，而非停止位加起始位。

有关报头格式的更多信息，请参阅节 7.6.7.1 和节 7.6.7.2。

#### 7.5.6.3.1 I2C 写入示例 1

本示例为通过 I2C 发送字节流的基础示例。

#### 备注

在使用该模块之前，必须将相应的 GPIO 设置为特殊功能 (请参阅节 7.3.4 和节 7.6.1.18)，并且 MRAM 必须通过分配内存启用模块 (请参阅节 7.5.3 和节 7.6.1.17)。只有将 GPIO 设置为特殊功能，且通过选择分配内存的 MRAM 配置启用该模块后，方可使用该模块。相应的 I2C 配置寄存器可参阅节 7.6.6。

表 7-25. 设计参数和假设

参数	值
CAN 帧格式	标准帧格式
I2C 目标地址 (7 位)	0x12
I2C 帧	写入
要传输的字节	8B
要写入的 I2C 数据	0x01、0x02、0x03、0x04、0x05、0x06、0x07、0x08

下表显示了在 CAN 总线上传输的步骤和数据。总线的中央网关/控制器称为 ECU，TCAN 器件称为“响应器”。

- 标记为**粗体**的十六进制值表示 CAN 帧头的字节 (本例中为标准帧格式)
- 标记为**斜体**的十六进制值表示 I2C 报头字节
- 无格式的十六进制值为 I2C 数据字节

**表 7-26. CAN 序列示例 ( 单个 CAN 帧 )**

步骤	发送器	数据	说明
1	ECU (DLC = 16B)	<b>0x0A, 0x30, 0x10, 0x25, 0x88, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x00, 0x00, 0x00</b>	ECU 请求向寄存器 0x3010 (I2C TX FIFO) 写入 10 字节数据, 对应的 I2C 报头参数为: 存储 = 1、I2C 目标地址 = 0x12、写入 = 1、I2C 数据大小 = 8 字节。随后将 10 字节数据移入设备。由于需容纳这 13 字节数据, CAN DLC 必须设为 16 字节, 因此在末尾填充 0x00 字节。
2	响应器	<b>0x0A, 0x01</b>	响应器发回一个 OK 以确认已收到请求。
3	I2C	I2C 帧: 0x25、0x88、0x01、0x02、0x03、0x04、0x05、0x06、0x07、0x08	此时, 响应器开始执行 10 字节的 I2C 传输。

**表 7-27. 示例 CAN 序列 ( 多个 CAN 帧到传输有效载荷 )**

步骤	发送器	数据	说明
1	ECU (DLC = 7B)	<b>0x04, 0x30, 0x10, 0x25, 0x88, 0x01, 0x02</b>	ECU 请求向寄存器 0x3010 (I2C TX FIFO) 写入 4 字节数据, 对应的 I2C 报头参数为: 存储 = 1、I2C 目标地址 = 0x12、写入 = 1、I2C 数据大小 = 8 字节。随后将 10 字节数据移入设备。
2	响应器	<b>0x04, 0x01</b>	响应器发回一个 OK 以确认已收到请求。
3	ECU (DLC = 12B)	<b>0x06, 0x30, 0x10, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x00, 0x00, 0x00</b>	ECU 继续向同一地址发起 6 字节写入请求。由于能容纳该帧的 CAN DLC 为 12 字节, 因此在末尾填充 3 字节。填充字节的值无实际意义, 会被忽略。
4	响应器	<b>0x06, 0x01</b>	响应器发回一个 OK 以确认已收到请求。
5	I2C	I2C 帧: 0x25、0x88、0x01、0x02、0x03、0x04、0x05、0x06、0x07、0x08	此时, 响应器开始执行 10 字节的 I2C 传输。

#### 7.5.6.3.2 I2C 读取示例 1

这是一个通过 I2C 发送和接收字节流的基础示例。一旦器件在 TX FIFO 中接收到任何字节后, 该器件将开始发送 I2C 字节。

#### 备注

在使用该模块之前, 必须将相应的 GPIO 设置为特殊功能 ( 请参阅节 7.3.4 和节 7.6.1.18 ), 并且 MRAM 必须通过分配内存启用模块 ( 请参阅节 7.5.3 和节 7.6.1.17 )。只有将 GPIO 设置为特殊功能, 且通过选择分配内存的 MRAM 配置启用该模块后, 方可使用该模块。相应的 I2C 配置寄存器可参阅节 7.6.6。

**表 7-28. 设计参数和假设**

参数	值
CAN 帧格式	标准帧格式
I2C 目标地址 ( 7 位 )	0x12
I2C 帧	读取
I2C 目标寄存器地址	0x10
要传输的字节	8B

下表显示了在 CAN 总线上传输的步骤和数据。总线的中央网关/控制器称为 ECU, TCAN 器件称为“响应器”。

- 标记为**粗体**的十六进制值表示 CAN 帧头的字节 ( 本例中为标准帧格式 )
- 标记为**斜体**的十六进制值表示 I2C 报头字节
- 无格式的十六进制值为 I2C 数据字节

执行 I2C 读取时，常规操作是先写入寄存器地址，再执行读取。这可使目标器件知道要读取哪个寄存器。这意味着 I2C 读取通常包含两个独立的 I2C 帧：先执行 I2C 写入，再执行 I2C 读取。此示例如下所示。

表 7-29. CAN 序列示例 ( 单个 CAN 帧 )

步骤	发送器	数据	说明
1	ECU (DLC = 6B)	<b>0x03, 0x30, 0x10</b> , 0x25, 0x81, 0x10	ECU 请求向寄存器 0x3010 (I2C TX FIFO) 写入 3 字节数据，对应的 I2C 报头参数为：存储 = 1、I2C 目标地址 = 0x12、写入 = 1、I2C 数据大小 = 1 字节 ( 寄存器地址 0x10 )
2	响应器	<b>0x03, 0x01</b>	响应器发回一个 OK 以确认已收到请求。
3	I2C	I2C 帧：0x25, 0x10	此时，响应器开始执行 1 字节数据的 I2C 传输。
4	ECU (DLC = 3B)	<b>0x43, 0x30, 0x10</b>	ECU 请求读取 RX FIFO，以检查用于设置目标地址的初始 I2C 写入状态。这样做是为了验证目标是否已应答。
5	响应器	<b>0x43, 0x01</b> , 0x25, 0x01, 0x01	响应器返回 I2C 状态标志 0x01，表示“成功”状态，即目标器件已应答地址和数据 ( 寄存器地址 )。
6	ECU (DLC = 5B)	<b>0x02, 0x30, 0x10</b> , 0x24, 0x08	ECU 现请求向寄存器 0x3010 (I2C TX FIFO) 写入 2 字节数据，对应的 I2C 报头参数为：存储 = 1、I2C 目标地址 = 0x12、写入 = 1、I2C 读取大小 = 8 字节
7	响应器	<b>0x02, 0x01</b>	响应器发回一个 OK 以确认已收到请求。
8	I2C	I2C 帧：0x24、0x01、0x02、0x03、0x04、0x05、0x06、0x07、0x08	此时，响应器开始执行 8 字节的 I2C 读取
9	ECU (DLC = 3B)	<b>0x4B, 0x30, 0x10</b>	ECU 请求从 RX FIFO 读取 11 字节数据 ( 2 字节报头 + 8 字节数据 + 1 字节状态 )，以获取从 I2C 目标读取的数据
10	响应器	<b>0x4B, 0x01</b> , 0x24, 0x09, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x01, 0x00, 0x00, 0x00	响应器发回 I2C 报头信息及 8 字节数据。数据后附带 I2C 状态标志 0x01，表示“成功”状态，即目标已应答地址。由于能容纳该数据的最近 CAN FD 有效载荷大小为 16 字节，因此在末尾填充 0x00 字节。

表 7-30. 示例 CAN 序列 ( 多个 CAN 帧到传输有效载荷 )

步骤	发送器	数据	说明
1	ECU (DLC = 6B)	<b>0x03, 0x30, 0x10</b> , 0x25, 0x81, 0x10	ECU 请求向寄存器 0x3010 (I2C TX FIFO) 写入 3 字节数据，对应的 I2C 报头参数为：存储 = 1、I2C 目标地址 = 0x12、写入 = 1、I2C 数据大小 = 1 字节 ( 寄存器地址 0x10 )
2	响应器	<b>0x0A, 0x01</b>	响应器发回一个 OK 以确认已收到请求。
3	I2C	I2C 帧：0x25, 0x10	此时，响应器开始执行 1 字节数据的 I2C 传输。
4	ECU (DLC = 3B)	<b>0x43, 0x30, 0x10</b>	ECU 请求读取 RX FIFO，以检查用于设置目标地址的初始 I2C 写入状态。这样做是为了验证目标是否已应答。
5	响应器	<b>0x43, 0x01</b> , 0x25, 0x01, 0x01	响应器返回 I2C 状态标志 0x01，表示“成功”状态，即目标器件已应答地址和数据 ( 寄存器地址 )。
6	ECU (DLC = 5B)	<b>0x02, 0x30, 0x10</b> , 0x24, 0x08	ECU 现请求向寄存器 0x3010 (I2C TX FIFO) 写入 2 字节数据，对应的 I2C 报头参数为：存储 = 1、I2C 目标地址 = 0x12、写入 = 1、I2C 读取大小 = 8 字节
7	响应器	<b>0x02, 0x01</b>	响应器发回一个 OK 以确认已收到请求。
8	I2C	I2C 帧：0x24、0x01、0x02、0x03、0x04、0x05、0x06、0x07、0x08	此时，响应器开始执行 8 字节的 I2C 读取
9	ECU (DLC = 3B)	<b>0x46, 0x30, 0x10</b>	ECU 请求从 RX FIFO 读取 6 字节数据，以获取从 I2C 目标读取的数据
10	响应器	<b>0x46, 0x01</b> , 0x24, 0x09, 0x01, 0x02, 0x03, 0x04	响应器发回 I2C 报头信息及 8 字节数据。
11	ECU (DLC = 3B)	<b>0x47, 0x30, 0x10</b>	ECU 请求从 RX FIFO 读取 7 字节数据 ( 2 字节报头 + 8 字节数据 + 1 字节状态 - 已读取的 4 字节 )，以获取从 I2C 目标读取的数据

表 7-30. 示例 CAN 序列 ( 多个 CAN 帧到传输有效载荷 ) ( 续 )

步骤	发送器	数据	说明
12	响应器	0x47, 0x01, 0x24, 0x85, 0x05, 0x06, 0x07, 0x08, 0x01, 0x00, 0x00, 0x00	响应器发回 I2C 报头信息及 8 字节数据。第二个 I2C 报头字节设置了 CONT 标志, 表示此为部分读取的 I2C 帧的延续。数据后附带 I2C 状态标志 0x01, 表示“成功”状态, 即目标已应答地址。由于能容纳该数据的最近 CAN FD 有效载荷大小为 12 字节, 因此在末尾填充 0x00 字节

### 7.5.7 PWM 和梯形 PWM 斜坡曲线

TCAN5102-Q1 具有 2 个高度可配置的 PWM 输出 ( PWM0 和 PWM1 ), 支持 8 位或 10 位输出分辨率, 频率范围宽达 20Hz 至 100KHz。主要有 2 种工作模式: 静态 PWM 输出和梯形 PWM 斜坡控制。斜坡控制允许对两个参数进行斜坡调节: PWM 输出的占空比或频率。

#### 备注

必须通过在 IO\_CFG\_1 中将相应 GPIO 设置为特殊功能模式, 将 PWM 模块多路复用到 GPIO。如果未进行此操作, PWM 信号不会多路复用到引脚。

当 PWM 模式配置为静态输出时, 由于不会发生斜坡控制, 因此只需要部分寄存器。

- 频率由 PWM\_END\_VAL 寄存器 ( 0x4019-0x401B 和 0x4119-0x411B ) 控制
- 占空比由 PWM\_CONST 寄存器 ( 0x400B-0x400C 和 0x410B-0x410C ) 控制

这里有 2 个可配置的斜坡: 一个导通斜坡和一个关断斜坡。通过在请求另一个导通斜坡之前更新斜坡值, 导通斜坡可用于创建不同的斜坡曲线。关断斜坡用于提供可预测且受控的斜坡下降/关断过渡。硬关断可立即禁用 PWM 输出。导通斜坡完成后, PWM 输出保持在结束值, 直到收到更改指令。关断斜坡斜升至指定的结束点, 然后禁用 PWM 输出。

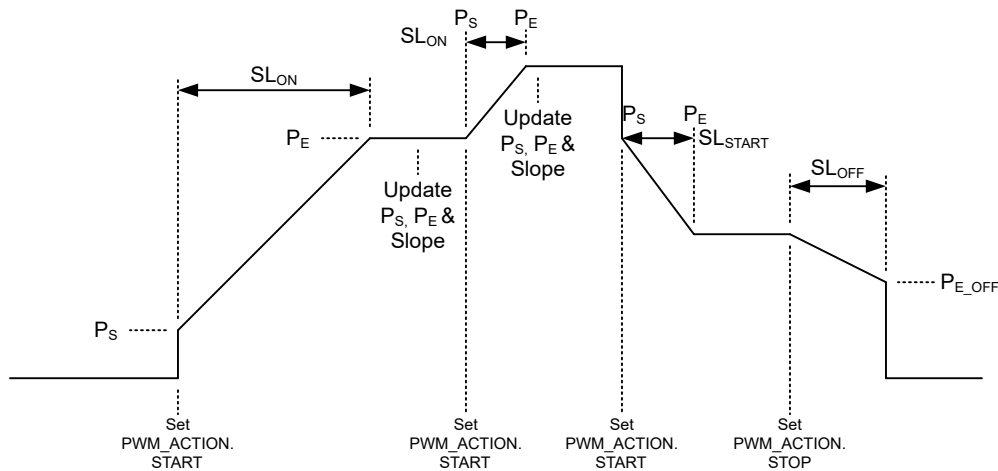


图 7-8. PWM 斜坡控制示例

其中各点定义如下:

- $P_S$ : 起始点 ( 对于导通斜坡 ) 描述了导通斜坡的起始点。
- $P_E$ : 结束点 ( 对于导通斜坡 ) 描述了导通斜坡的结束点。
- $SL_{ON}$ : 斜坡斜率 ( 对于导通斜坡 ) 描述了导通斜坡期间的变化率。
- $P_{E\_OFF}$ : 结束点 ( 对于关断斜坡 ) 描述了关断斜坡的结束点。
- $SL_{OFF}$ : 斜坡斜率 ( 对于关断斜坡 ) 描述了关断斜坡期间的变化率。

构成 PWM 斜坡的 3 个值为: 起始点、结束点和斜坡斜率。导通斜坡的起始点可以是当前 PWM 值或指定的起始点。这使得可以灵活地创建阶跃函数类型的行为 ( 如果需要 ), 或允许对现有斜坡进行平滑更改。斜坡斜率用于描述变化参数随时间的变化率。例如, 斜坡斜率可用于告知设备以 5%/ 秒的速率改变输出占空比。

由于有两个 PWM 参数 ( PWM 频率和占空比 ) 可进行调节, 因此未调节的参数是用户可配置的静态值。例如, 如果用户在斜坡调节中改变 PWM 频率, 占空比可以设置为任何值 ( 如 30% ), 但在斜坡调节期间不能改变。

#### 备注

在相应的斜坡调节期间更新导通斜坡或关断降斜坡曲线设置的值不会实时改变斜坡值。斜坡设置在设置动作命令 ( 如启动或停止动作 ) 时加载到缓冲区中。

#### 7.5.7.1 梯形斜坡控制

TCAN5102-Q1 支持硬件控制的梯形斜坡。斜坡控制支持 2 项参数：

- 占空比斜坡控制
- 开关频率斜坡控制

对于变化参数 ( 占空比或开关频率 ) 的斜坡控制, 有一对起始点值和结束点值。这对值描述了 PWM 输出的占空比或开关频率的起始点和结束点。未变化的参数 ( 非变化参数 ) 只有一个可设置的值, 且该值在斜坡期间不变化。

关断斜坡是接到关闭命令时的一种规定行为。这包含一个已设定的目标停止值 ( 斜坡将渐变至该值 ), 以及特定的变化速率。关断斜坡完成后, PWM 输出将被禁用, 直至重新使能。

##### 7.5.7.1.1 占空比斜坡控制

TCAN5102-Q1 支持通过改变占空比实现斜坡控制。当占空比处于斜坡控制时, 开关频率为固定值。

在斜坡控制期间, 占空比在每个周期开始时更新。换言之, 输出占空比值以开关频率速率更新。

在图 7-9 中, 如红色箭头所示, 占空比值在每个周期开始时更新。

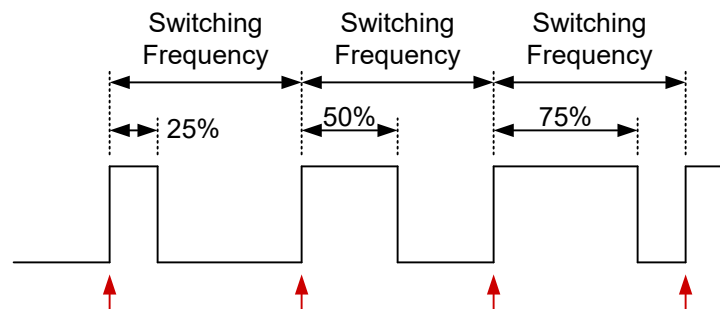


图 7-9. 占空比斜坡控制示例

#### 备注

如果斜坡相对较慢, 则在加载不同值之前可能会出现多个周期。这是由于 8 位或 10 位输出的最小步长导致的。

##### 7.5.7.1.2 开关频率斜坡控制

TCAN5102-Q1 支持通过改变开关频率实现斜坡控制。当开关频率处于斜坡状态时, 占空比固定。

在斜坡期间, 每 10 $\mu$ s 计算一次开关频率值, 但输出开关频率仅在周期开始时更新。这是为了确保在某个频率下至少有一个完整周期。

如图 7-10 所示, 斜坡的频率值每 10 $\mu$ s 计算一次, 但使用的是新周期开始时生成的最后一个值。对于 100 kHz 以上的速度, 在值更新前会出现多个周期。



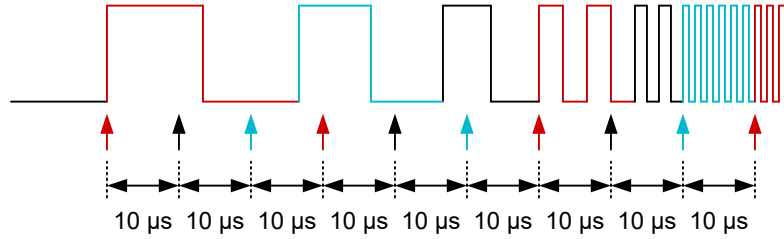


图 7-10. 开关频率斜坡更新

#### 备注

如果斜坡较慢，在加载不同的开关频率前可能会出现多个周期。这是由最小步长导致的。

#### 7.5.7.1.3 关断斜坡

TCAN5102-Q1 具有一组单独的关断斜坡值，可实现指定的关断行为。当通过向 PWM\_ACTION 寄存器的 STOP 位写入值来请求关断斜坡时，将使用这些数值。请求关断斜坡的时机无关紧要，既可在现有导通斜坡过程中进行，也可在 PWM 输出当前未处于斜坡控制状态时进行。关断斜坡始终遵循关断斜坡寄存器中指定的关断斜坡斜率和结束点。如果结束点不为零，则在可变参数达到指定的结束点后，PWM 输出会关闭。这一功能可用于生成阶跃函数，在电机驱动场景中十分实用。关断斜坡的起始点始终为可变参数的当前值。

图 7-11 中展示了 2 个关断斜坡示例。红线表示在导通斜坡过程中设置 STOP 位时的输出情况。黑线表示在导通斜坡完成且 PWM 输出处于空闲状态后设置 STOP 位时的输出情况。

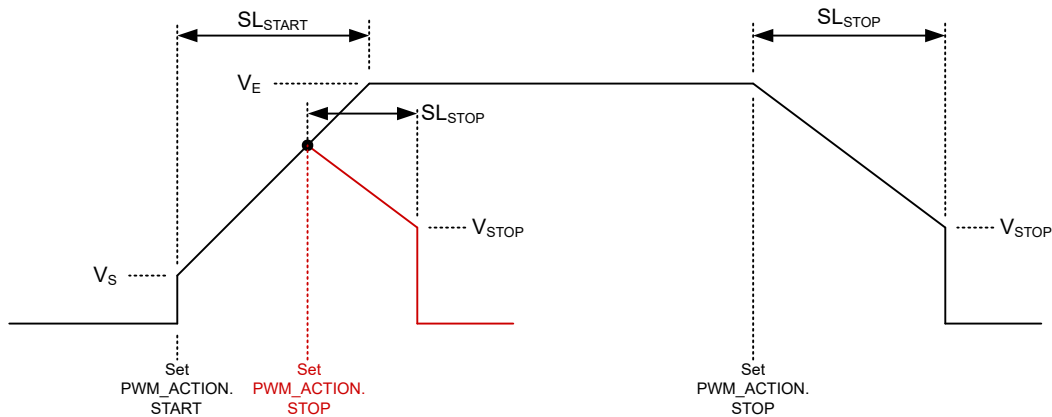


图 7-11. PWM 关断斜坡示例

#### 7.5.7.2 PWM 时钟发生器

TCAN5102-Q1 具有支持小数值的 PWM 时钟发生器。频率值始终为 11 位整数分频值加 7 位小数分频值。PWM 时钟发生器的输入时钟为 40MHz，可设置生成 8 位 (256 级) 或 10 位 (1024 级) 输出 (占空比分辨率)。可能的分频值范围为 1 至  $(2^{11} - 1)$ ，小数分辨率为 1/128。

计算分频值的公式为： $\text{divisor} = (40\text{E6} / (\text{Desired Switching Frequency} \times 2^{\text{RES}}))$

其中，“RES”为 PWM 输出的分辨率的位数 (8 或 10)。

要计算所需的寄存器值，需要以下函数：

- TRUNC(x)：对 x 取整数部分。仅返回该数的整数部分。(例如：TRUNC(3.14) = 3)
- ROUND(x)：将 x 舍入到最接近的整数。(例如：ROUND(3.1) = 3 和 ROUND(3.6) = 4)
- >>：右移位操作。例如：0x1000 >> 8 = 0x0010。或者 0b0001 0000 0000 0000 >> 8 = 0b0000 0000 0001 0000
- &：按位与函数，用于屏蔽位。例如：0x1234 & 0x00FF = 0x0034 以及 0x8765 & 0xFF00 = 0x8700

计算出分频值后，可通过以下 3 个公式计算寄存器值：

$$\text{FREQ MSB} = \text{TRUNC}(\text{Divisor}) \gg 8 \quad (7)$$

$$\text{FREQ LSB} = \text{TRUNC}(\text{Divisor}) \& 0x00FF \quad (8)$$

$$\text{FREQ FRAC} = \text{ROUND}([\text{Divisor} - \text{TRUNC}(\text{Divisor})] \times 128) \quad (9)$$

表 7-31. 开关频率 ( 8 位输出 )

期望开关频率 (Hz)	用于生成频率的分频值	可获得的最接近的分 频值	频率高位值 (十六进 制)	频率低位值 (十六进 制)	频率小数位 (十六进 制)	期望值与实际值之间的误差 差异百分比 (%)
100	1562.5000	1562 64/128	0x06	0x1A	0x40	0
250	625.0000	625	0x02	0x71	0x00	0
500	312.5000	312 64/128	0x01	0x38	0x40	0
1,000	156.2500	156 32/128	0x00	0x9C	0x20	0
2,500	62.5000	62 64/128	0x00	0x3E	0x40	0
5,000	31.2500	31 32/128	0x00	0x1F	0x20	0
7,500	20.8333	20 107/128	0x00	0x14	0x6B	0.013
10,000	15.6250	15 80/128	0x00	0x0F	0x50	0
25,000	6.2500	6 32/128	0x00	0x06	0x20	0
50,000	3.1250	3 16/128	0x00	0x03	0x10	0
75,000	2.0833	2 11/128	0x00	0x02	0x0B	0.125
100,000	1.5625	1 72/128	0x00	0x01	0x48	0

表 7-32. 开关频率 ( 10 位输出 )

期望开关频率 (Hz)	用于生成频率的分频值	可获得的最接近的分 频值	频率高位值 (十六进 制)	频率低位值 (十六进 制)	频率小数位 (十六进 制)	期望值与实际值之间的误差 差异百分比 (%)
50	781.2500	781 32/128	0x03	0x0D	0x20	0
100	390.6250	390 80/128	0x01	0x86	0x50	0
250	156.2500	156 32/128	0x00	0x9C	0x20	0
500	78.1250	78 16/128	0x00	0x4E	0x10	0
1,000	39.0625	39 8/128	0x00	0x27	0x08	0
2,500	15.6250	15 80/128	0x00	0x0F	0x50	0
5,000	7.8125	7 104/128	0x00	0x07	0x68	0
7,500	5.2083	5 27/128	0x00	0x05	0x1B	0.05
10,000	3.9063	3 116/128	0x00	0x03	0x74	0
25,000	1.5625	1 72/128	0x00	0x01	0x48	0

#### 备注

配置为 8 位输出模式时，可生成大于 100kHz 的值。运行频率斜坡时有一些特别注意事项。有关更多信息，请参阅[开关频率斜坡控制](#)。

用于写入值的相应寄存器根据要斜升的参数而异。例如，如果斜坡是开关频率斜坡，则寄存器值写入 PWM\_START、PWM\_END 和 PWM\_OFF 寄存器组。如果斜坡是占空比斜坡，则频率寄存器值写入 PWM\_CONST 和 PWM\_END\_VAL\_CONST\_FRAC\_F。

#### 7.5.7.3 PWM 占空比

TCAN5102-Q1 具有 PWM 模块，可设置为生成 8 位 ( 256 级 ) 或 10 位 ( 1024 级 ) 输出 ( 占空比分辨率 )。



占空比计数值的计算很简单，且根据 PWM 占空比是配置为 8 位模式还是 10 位模式而变化。

要计算所需的寄存器值，需要以下函数：

- **ROUND(x)**：将 x 舍入到最接近的整数。（例如：ROUND(3.1) = 3 和 ROUND(3.6) = 4）

$DC_{TARGET}$  表示所需的占空比百分比，为浮点数（例如：40% = 0.40，25% = 0.25）

$$DC\_COUNT_{8-BIT} = ROUND(DC_{TARGET} \times 256) \quad (10)$$

$$DC\_COUNT_{10-BIT} = ROUND(DC_{TARGET} \times 1024) \quad (11)$$

**表 7-33. 占空比**

所需占空比 (%)	PWM_CTRL.DC_8B = 1 8 位				PWM_CTRL.DC_8B = 0 10 位			
	占空比计数	可获取的最接近计数	占空比计数 (十六进制)	实际占空比 (%)	占空比计数	可获取的最接近计数	占空比计数 (十六进制)	实际占空比 (%)
5	12.800	13	0x0D	5.08%	51.200	51	0x033	4.98%
10	25.600	26	0x1A	10.16%	102.400	102	0x066	9.96%
15	38.400	38	0x26	14.84%	153.600	154	0x099	15.04%
20	51.200	51	0x33	19.92%	204.800	205	0x0CC	20.02%
25	64.000	64	0x40	25.00%	256.000	256	0x100	25.00%
30	76.800	77	0x4D	30.08%	307.200	307	0x133	29.98%
35	89.600	90	0x5A	35.16%	358.400	358	0x166	34.96%
40	102.400	102	0x66	39.84%	409.600	410	0x199	40.04%
45	115.200	115	0x73	44.92%	460.800	461	0x1CC	45.02%
50	128.000	128	0x80	50.00%	512.000	512	0x200	50.00%
55	140.800	141	0x8D	55.08%	563.200	563	0x233	54.98%
60	153.600	154	0x9A	60.16%	614.400	614	0x266	59.96%
65	166.400	166	0xA6	64.84%	665.600	666	0x299	65.04%
70	179.200	179	0xB3	69.92%	716.800	717	0x2CC	70.02%
75	192.000	192	0xC0	75.00%	768.000	768	0x300	75.00%
80	204.800	205	0xCD	80.08%	819.200	819	0x333	79.98%
85	217.600	218	0xDA	85.16%	870.400	870	0x366	84.96%
90	230.400	230	0xE6	89.84%	921.600	922	0x399	90.04%
95	243.200	243	0xF3	94.92%	972.800	973	0x3CC	95.02%

#### 7.5.7.4 斜坡斜率和标度因子

斜率控制是梯形 PWM 斜坡的一部分，用于使变化参数（如占空比或开关频率）随时间线性变化。本器件规定的斜率单位为 Hz/10μs（用于开关频率斜坡）或 %/周期（用于占空比斜坡）。

斜率值由 21 位组成，这些位分为整数部分和小数部分。整数与小数的划分由 PWM\_CTRL.SLOPE\_SCALE 位决定。可调节性使用户能够在慢速斜坡和快速斜坡之间进行选择。请参见下面的相关部分，确定所需的范围。

#### 备注

对于相同的斜率标度，整数位和小数位的划分因斜坡的是占空比还是频率而异。具体划分请参见下面各部分。

##### 7.5.7.4.1 占空比斜坡斜率

对于占空比斜坡控制，斜率调节以“%/周期”为单位进行。换言之，即每个开关周期（开关频率）内的占空比变化量。

由于占空比值在每个周期开始时更新，因此斜坡的最慢和最快速度取决于多个因素：

- 变化百分比。从 0% 至 100% 所需时间比从 0% 至 50% 长。
- 输出分辨率：对于相同百分比变化，10 位分辨率比 8 位慢，因为需要更多步骤。
- 开关频率：这决定了占空比更新的频率，因为更新发生在新周期开始时。

斜率的目标值可通过以下公式，由随时间的目标百分比变化量计算得出：

$$\text{SLOPE\_CHANGE\_STEPS} = \text{ABS}(\text{SLOPE\_CHANGE\_DEC}) \times (2^{\text{NUM\_DC\_BITS}}) \quad (12)$$

$$\text{SLOPE\_VAL} = \text{SLOPE\_CHANGE\_STEPS} / (t \times f) \quad (13)$$

$$\text{SLOPE\_INT} = \text{TRUNCATE}(\text{SLOPE\_VAL}) \quad (14)$$

$$\text{SLOPE\_FRAC} = \text{ROUND}([\text{SLOPE\_VAL} - \text{TRUNCATE}(\text{SLOPE\_VAL})] \times 2^{\text{NUM\_FRAC\_BITS}}) \quad (15)$$

其中，“SLOPE\_CHANGE\_DECIMAL”为目标百分比变化量（例如：从 100% 开始，至 40% 结束，变化量为 60%，即小数形式的 0.6），但此公式仅用于求解 SLOPE\_CHANGE\_STEPS。SLOPE\_CHANGE\_STEPS 用于表示总变化步数。例如，占空比从 102/1024（约 10%）开始到 768/1024（约 75%），总变化为 768-102=666 步。

“NUM\_DC\_BITS”为占空比分辨率的位数（8 位或 10 位）。“t”为变化发生的时间量（以秒为单位）。“f”为开关频率（以 Hz 或脉冲/秒为单位）。

简单示例：若用户希望在 2 秒内从 25% 斜坡变化至 75%，开关频率为 5KHz，分辨率为 10 位模式，则斜率变化 = 75%-25%=50%（即 0.5），t=2，f=5000，NUM\_DC\_BITS=10（10 位占空比模式）。由此得出 SLOPE\_VAL 为  $(0.50 \times 2^{10}) / (2 \times 5000) = 0.2048$ 。由于该斜率值较低，而 SLOPE\_SCALE = 3b000 (1.999999)，为获得斜坡的最高分辨率，最佳选择是将 SLOPE\_SCALE 设为 3b000，即 1 位整数位和 20 位小数位。SLOPE\_INT = 0 且 SLOPE\_FRAC =  $0.2048 \times 2^{20} = 214,748.3648$ 。四舍五入为十进制 214,748，或十六进制 0x346DC。

斜率标度设置决定了如何将 21 位数据在整数字段和小数字段之间进行分配。较慢的斜坡需要更大的小数字段。

**表 7-34. 斜率标度选项 (占空比)**

SLOPE_SCALE	NUM_INT_BITS	NUM_FRAC_BITS	最小斜率值	最大斜率值
3b000	1 - [20]	20 - [19:0]	1/1,048,576	1.999999
3b001	2 - [20:19]	19 - [18:0]	1/524,288	3.999998
3b010	3 - [20:18]	18 - [17:0]	1/262,144	7.999996
3b011	4 - [20:17]	17 - [16:0]	1/131,072	15.999992
3b100	5 - [20:16]	16 - [15:0]	1/65536	31.999985

**表 7-35. 占空比斜坡速度限制 (0% 至 100%，SLOPE\_SCALE = 3b000)**

开关频率 (Hz)	8-BIT 占空比		10-BIT 占空比	
	最慢斜坡 (秒)	最快斜坡 (秒)	最慢斜坡 (秒)	最快斜坡 (秒)
19 (仅限 10 位模式)	14,128,182	6.7E+0	56,512,728	26.9E+0
50 (仅限 10 位模式)	5,368,709	2.6E+0	21,474,836	10.2E+0
76	3,532,045	1.7E+0	14,128,182	6.7E+0
100	2,684,355	1.3E+0	10,737,418	5.1E+0
250	1,073,742	512.0E-3	4,294,967	2.0E+0
500	536,871	256.0E-3	2,147,484	1.0E+0
1,000	268,435	128.0E-3	1,073,742	512.0E-3
2,500	107,374	51.2E-3	429,497	204.8E-3
5,000	53,687	25.6E-3	214,748	102.4E-3
7,500	35,791	17.1E-3	143,166	68.3E-3

表 7-35. 占空比斜坡速度限制 ( 0% 至 100% , SLOPE\_SCALE = 3b000 ) ( 续 )

开关频率 (Hz)	8-BIT 占空比		10-BIT 占空比	
	最慢斜坡 ( 秒 )	最快斜坡 ( 秒 )	最慢斜坡 ( 秒 )	最快斜坡 ( 秒 )
10,000	26,844	12.8E-3	107,374	51.2E-3
25,000	10,737	5.1E-3	42,950	20.5E-3
39,000	6,883	3.3E-3	27,532	13.1E-3
50,000 ( 仅限 8 位模式 )	5,369	2.6E-3	21,475	10.2E-3
75,000 ( 仅限 8 位模式 )	3,579	1.7E-3	14,317	6.8E-3
100,000 ( 仅限 8 位模式 )	2,684	1.3E-3	10,737	5.1E-3

表 7-36. 占空比斜坡速度限制 ( 0% 至 100% , SLOPE\_SCALE = 3b001 )

开关频率 (Hz)	8-BIT 占空比		10-BIT 占空比	
	最慢斜坡 ( 秒 )	最快斜坡 ( 秒 )	最慢斜坡 ( 秒 )	最快斜坡 ( 秒 )
19 ( 仅限 10 位模式 )	7,064,091	3.4E+0	28,256,364	13.5E+0
50 ( 仅限 10 位模式 )	2,684,355	1.3E+0	10,737,418	5.1E+0
76	1,766,023	842.1E-3	7,064,091	3.4E+0
100	1,342,177	640.0E-3	5,368,709	2.6E+0
250	536,871	256.0E-3	2,147,484	1.0E+0
500	268,435	128.0E-3	1,073,742	512.0E-3
1,000	134,218	64.0E-3	536,871	256.0E-3
2,500	53,687	25.6E-3	214,748	102.4E-3
5,000	26,844	12.8E-3	107,374	51.2E-3
7,500	17,896	8.5E-3	71,583	34.1E-3
10,000	13,422	6.4E-3	53,687	25.6E-3
25,000	5,369	2.6E-3	21,475	10.2E-3
39,000	3,441	1.6E-3	13,766	6.6E-3
50,000 ( 仅限 8 位模式 )	2,684	1.3E-3	10,737	5.1E-3
75,000 ( 仅限 8 位模式 )	1,790	853.3E-6	7,158	3.4E-3
100,000 ( 仅限 8 位模式 )	1,342	640.0E-6	5,369	2.6E-3

表 7-37. 占空比斜坡速度限制 ( 0% 至 100% , SLOPE\_SCALE = 3b010 )

开关频率 (Hz)	8-BIT 占空比		10-BIT 占空比	
	最慢斜坡 ( 秒 )	最快斜坡 ( 秒 )	最慢斜坡 ( 秒 )	最快斜坡 ( 秒 )
19 ( 仅限 10 位模式 )	3,532,045	1.7E+0	14,128,182	6.7E+0
50 ( 仅限 10 位模式 )	1,342,177	640.0E-3	5,368,709	2.6E+0
76	883,011	421.1E-3	3,532,045	1.7E+0
100	671,089	320.0E-3	2,684,355	1.3E+0
250	268,435	128.0E-3	1,073,742	512.0E-3
500	134,218	64.0E-3	536,871	256.0E-3
1,000	67,109	32.0E-3	268,435	128.0E-3
2,500	26,844	12.8E-3	107,374	51.2E-3
5,000	13,422	6.4E-3	53,687	25.6E-3
7,500	8,948	4.3E-3	35,791	17.1E-3
10,000	6,711	3.2E-3	26,844	12.8E-3
25,000	2,684	1.3E-3	10,737	5.1E-3

表 7-37. 占空比斜坡速度限制 ( 0% 至 100% , SLOPE\_SCALE = 3b010 ) ( 续 )

开关频率 (Hz)	8-BIT 占空比		10-BIT 占空比	
	最慢斜坡 ( 秒 )	最快斜坡 ( 秒 )	最慢斜坡 ( 秒 )	最快斜坡 ( 秒 )
39,000	1,721	820.5E-6	6,883	3.3E-3
50,000 ( 仅限 8 位模式 )	1,342	640.0E-6	5,369	2.6E-3
75,000 ( 仅限 8 位模式 )	895	426.7E-6	3,579	1.7E-3
100,000 ( 仅限 8 位模式 )	671	320.0E-6	2,684	1.3E-3

表 7-38. 占空比斜坡速度限制 ( 0% 至 100% , SLOPE\_SCALE = 3b011 )

开关频率 (Hz)	8-BIT 占空比		10-BIT 占空比	
	最慢斜坡 ( 秒 )	最快斜坡 ( 秒 )	最慢斜坡 ( 秒 )	最快斜坡 ( 秒 )
19 ( 仅限 10 位模式 )	1,766,023	842.1E-3	7,064,091	3.4E+0
50 ( 仅限 10 位模式 )	671,089	320.0E-3	2,684,355	1.3E+0
76	441,506	210.5E-3	1,766,023	842.1E-3
100	335,544	160.0E-3	1,342,177	640.0E-3
250	134,218	64.0E-3	536,871	256.0E-3
500	67,109	32.0E-3	268,435	128.0E-3
1,000	33,554	16.0E-3	134,218	64.0E-3
2,500	13,422	6.4E-3	53,687	25.6E-3
5,000	6,711	3.2E-3	26,844	12.8E-3
7,500	4,474	2.1E-3	17,896	8.5E-3
10,000	3,355	1.6E-3	13,422	6.4E-3
25,000	1,342	640.0E-6	5,369	2.6E-3
39,000	860	410.3E-6	3,441	1.6E-3
50,000 ( 仅限 8 位模式 )	671	320.0E-6	2,684	1.3E-3
75,000 ( 仅限 8 位模式 )	447	213.3E-6	1,790	853.3E-6
100,000 ( 仅限 8 位模式 )	336	160.0E-6	1,342	640.0E-6

表 7-39. 占空比斜坡速度限制 ( 0% 至 100% , SLOPE\_SCALE = 3b100 )

开关频率 (Hz)	8-BIT 占空比		10-BIT 占空比	
	最慢斜坡 ( 秒 )	最快斜坡 ( 秒 )	最慢斜坡 ( 秒 )	最快斜坡 ( 秒 )
19 ( 仅限 10 位模式 )	883,011	421.1E-3	3,532,045	1.7E+0
50 ( 仅限 10 位模式 )	335,544	160.0E-3	1,342,177	640.0E-3
76	220,753	105.3E-3	883,011	421.1E-3
100	167,772	80.0E-3	671,089	320.0E-3
250	67,109	32.0E-3	268,435	128.0E-3
500	33,554	16.0E-3	134,218	64.0E-3
1,000	16,777	8.0E-3	67,109	32.0E-3
2,500	6,711	3.2E-3	26,844	12.8E-3
5,000	3,355	1.6E-3	13,422	6.4E-3
7,500	2,237	1.1E-3	8,948	4.3E-3
10,000	1,678	800.0E-6	6,711	3.2E-3
25,000	671	320.0E-6	2,684	1.3E-3
39,000	430	205.1E-6	1,721	820.5E-6

**表 7-39. 占空比斜坡速度限制 ( 0% 至 100% , SLOPE\_SCALE = 3b100 ) ( 续 )**

开关频率 (Hz)	8-BIT 占空比		10-BIT 占空比	
	最慢斜坡 ( 秒 )	最快斜坡 ( 秒 )	最慢斜坡 ( 秒 )	最快斜坡 ( 秒 )
50,000 ( 仅限 8 位模式 )	336	160.0E-6	1,342	640.0E-6
75,000 ( 仅限 8 位模式 )	224	106.7E-6	895	426.7E-6
100,000 ( 仅限 8 位模式 )	168	80.0E-6	671	320.0E-6

#### 7.5.7.4.2 开关频率斜坡斜率

对于开关频率斜坡控制，斜率调节以 “Hz/10μs” 为单位进行。

由于更新发生的周期固定为 10μs，因此斜坡的最小和最大时间仅取决于频率变化的大小。

斜率的期望值可以根据随时间变化的期望 Hz/sec，通过以下公式计算：

$$\text{SLOPE\_VAL} = ( \text{ABS}(\text{Freq Change}) \times 10\text{E-6} ) / t$$

$$\text{SLOPE\_INT} = \text{TRUNC}(\text{SLOPE\_VAL})$$

$$\text{SLOPE\_FRAC} = \text{ROUND}( [ \text{SLOPE\_VAL} - \text{TRUNC}(\text{SLOPE\_VAL}) ] \times 2^{\text{NUM\_FRAC\_BITS}} )$$

其中，“Freq Change” 是频率的变化量 ( 例如：从 200Hz 开始，到 2kHz 结束 = 1,800kHz )。“t” 是变化发生的时间量，单位为秒。

举个简单的例子，若用户希望在 4 秒内从 250Hz 斜坡到 2kHz，且运行时 SLOPE\_SCALE = 0b000，则频率变化量 = 2000 - 250 = 1750，t = 4，小数位数 = 16。这样可得出 SLOPE\_VAL 为 (1750 × 10E-6) / 4 = 0.07。SLOPE\_INT = 0 且 SLOPE\_FRAC = ROUND(0.07 × 2<sup>16</sup>) = 4588。

**表 7-40. 斜率标度选项 ( 开关频率 )**

SLOPE_SCALE	NUM_INT_BITS	NUM_FRAC_BITS	最小斜率值 (Hz/ 10μs)	最大斜率值 (Hz/ 10μs)	最小斜率值 (Hz/s)	最大斜率值 (Hz/s)
3b000	5 - [20:16]	16 - [15:0]	1/65536	31.99998	1.526	3.2E+6
3b001	9 - [20:12]	12 - [11:0]	1/4096	511.99976	24.414	51.2E+6
3b010	13- [20:8]	8 - [7:0]	1/256	8,191.99609	390.625	819.2E+6
3b011	17 - [20:4]	4 - [3:0]	1/16	131,071.93750	6250	13.1E+9
3b100	21 - [20:0]	0 - 无	1/1	2,097,151.00000	100000	209.7E+9

#### 7.5.7.5 自动减速和停止条件

除了支持上升或下降斜坡外，TCAN5102-Q1 还能够仅使用来自上游控制器的“启动”信号就能完成整个加速和减速斜坡。这是通过设置一个计数脉冲数量的计数器来实现的，用于确定何时开始“停止/关断”斜坡。还有一个计数器用于设置允许的最大脉冲数。当 PWM 斜坡期间生成的脉冲总数达到此值时，它将作为硬截止条件来停止 PWM 输出。所有 PWM 模式 ( 占空比、频率或静态 ) 都支持停止功能。

用户可选的 GPIO 可用于触发停止斜坡或立即关闭条件。这对于限位开关或某种故障需要立即禁用 PWM 输出的情况很有用。

##### 7.5.7.5.1 具有脉冲计数功能的自动减速关断斜坡

TCAN5102-Q1 可以使用脉冲计数器来计时何时开始关断斜坡。PWM\_PULSE\_STOP\_RAMP 寄存器用于设置 32 位计数器的比较值。启动新斜坡时，内部脉冲计数器会复位 ( 可通过读取 PWM\_CUR\_PULSE 查看 )，并在每个脉冲开始时向上计数。如果设置 START 位时 PWM.AUTO\_STOP 位已设为 1，则一旦 PWM\_CUR\_PULSE 与 PWM\_PULSE\_STOP\_RAMP 中的值匹配，PWM\_IP 会向 PWM\_ACTION.STOP 位写入 1，以自动开始 OFF\_RAMP。

然后，使用 PWM\_PULSE\_MAX 寄存器来确定何时关闭 PWM 输出。与 PWM\_PULSE\_STOP\_RAMP 寄存器类似，该值会与 PWM\_CUR\_PULSE 寄存器进行比较。当 PWM\_PULSE\_MAX 与 PWM\_CUR\_PULSE 寄存器匹配时，PWM 输出将关闭。注意，PWM\_PULSE\_MAX 寄存器的优先级高于 PWM\_PULSE\_STOP\_RAMP，因此若 PWM\_PULSE\_MAX 的值小于 PWM\_PULSE\_STOP\_RAMP 的值，则不会发生关断斜坡。

在图 7-12 中，显示了一个典型的自动关断斜坡。一旦达到  $P_{E\_OFF}$  值，输出将继续，直到达到 PWM\_PULSE\_MAX 值后关闭输出。在图 7-13 中，PWM\_PULSE\_MAX 值与 PWM\_PULSE\_STOP\_RAMP 过于接近，无法为关断斜坡提供足够的脉冲以使其完成，导致输出在关断斜坡中途被切断。

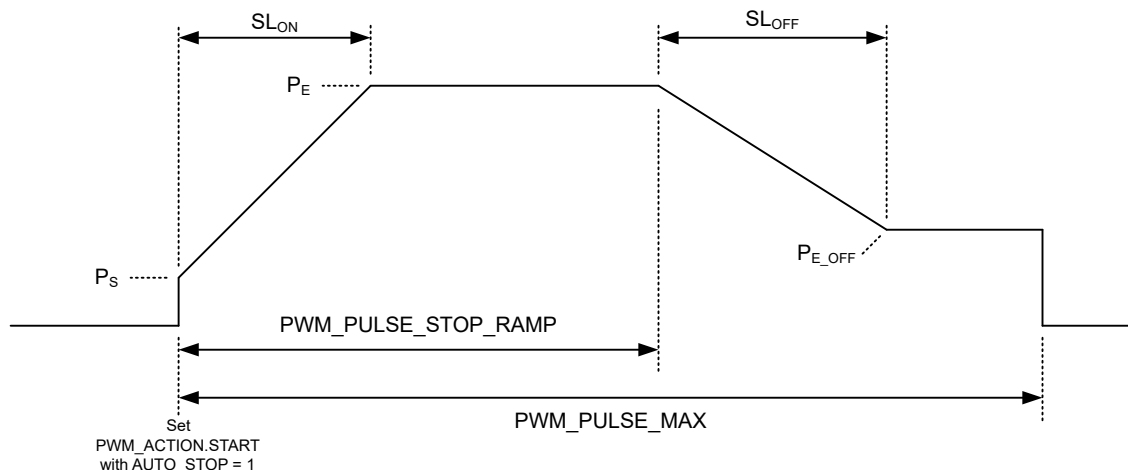


图 7-12. 自动关闭示例 1

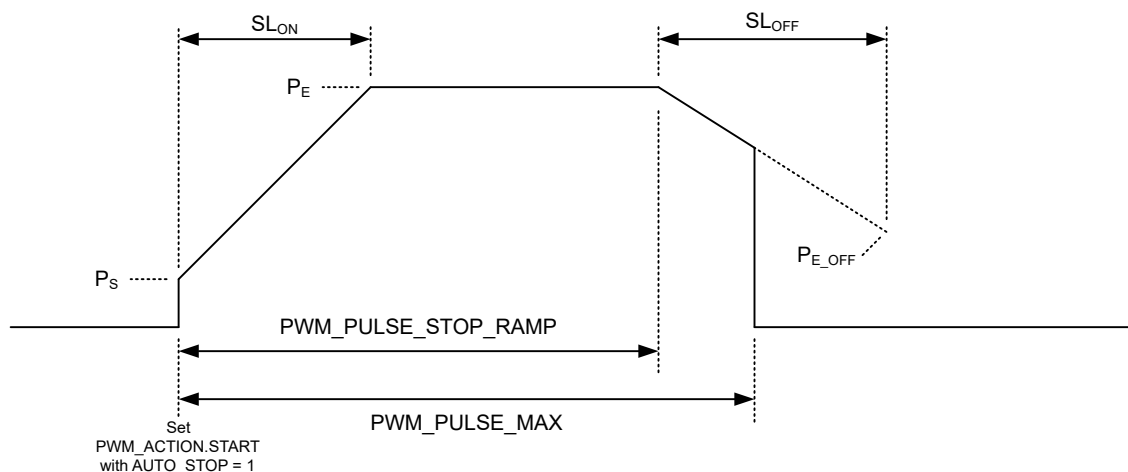


图 7-13. 自动关闭示例 2

#### 7.5.7.5.2 通过 GPIO 输入实现自动减速/关断斜坡

TCAN5102-Q1 可以使用 GPIO 输入触发关断斜坡。此功能的配置位于 PWM\_IAS\_CTRL 寄存器中，允许选择用作触发的 GPIO 及其极性。一旦 GPIO 切换到所选极性，行为取决于 PWM\_IAS\_CTRL.STOP\_MODE 的设置。如果 STOP\_MODE 设为 0，PWM\_ACTION.STOP 位将被置位，启动关断斜坡。如果 STOP\_MODE 设为 1，一旦 GPIO 切换到有效极性，PWM 输出立即停止。

图 7-14 中展示了 GPIO 输入切换到高电平有效时的典型自动关断斜坡。一旦达到  $P_{E\_OFF}$  值，输出就会关闭。图 7-15 中 STOP\_MODE 设为 1，导致输出立即关闭，无关断斜坡。

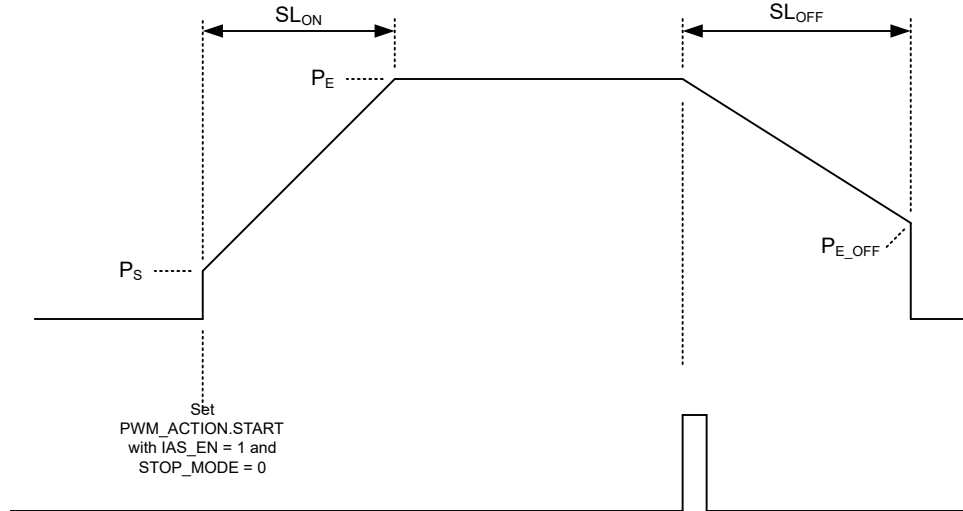


图 7-14. GPIO 触发关断示例 1

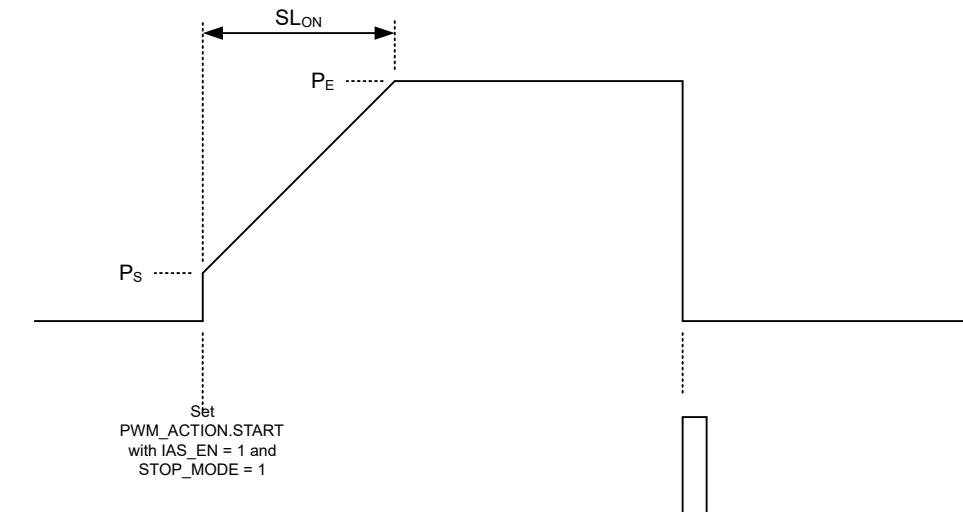


图 7-15. GPIO 触发关断示例 2

#### 7.5.7.6 占空比斜坡示例

本示例为配置具有以下参数的占空比斜坡的基础示例。



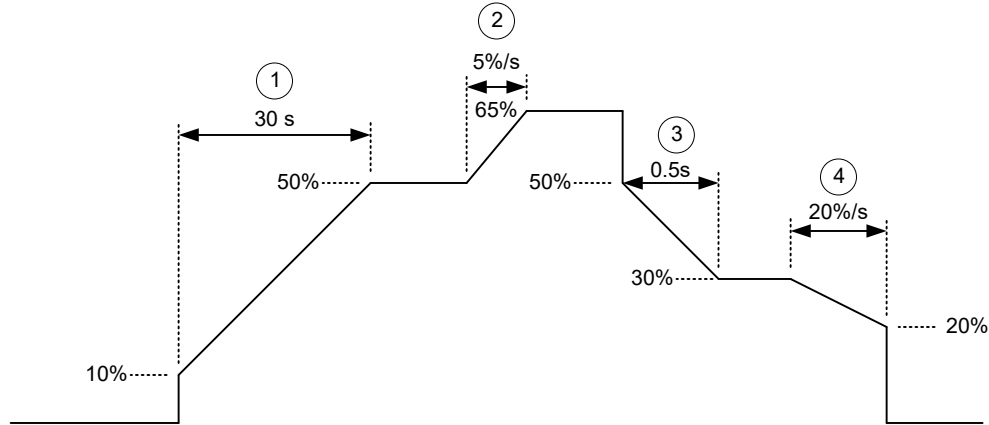


图 7-16. PWM 占空比斜坡示例

表 7-41. 设计参数和假设

参数	值
开关频率	20kHz
输出分辨率	10 位
斜率标度	0b000 (1/1,048,576)

上图所示的斜坡曲线可分解为 4 个独立的梯形斜坡。

1. 导通斜坡：30 秒内从 10% 至 50%
2. 导通斜坡：以 5%/秒的速率从 50% 至 65%
3. 导通斜坡：在 500ms 内从 50% 阶跃至 30%
4. 关断斜坡：以 20%/ 秒的速率斜坡下降至 20%，然后关闭输出。

改变斜率值的目的是说明如何从绝对时间或变化率目标计算斜率值。由于本示例由 4 个独立斜坡组成，下面对每个斜坡进行详细说明。这 4 个示例显示了获得相同结果的几种不同方法（不同的输入单元/合并一些步骤）。

表 7-42. 斜坡 1：30 秒内从 10 % 至 50 %

步骤	参数	示例	说明	值
1	起始值	$10\% \times 1024 = 102.4 \Rightarrow 102$	将起始占空比转换为 10 位值	d102 或 0x066
2	停止值	$50\% \times 1024 = 512$	将结束占空比转换为 10 位值	d512 或 0x200
3	斜率计算	$(512-102) = 410$	计算结束值与起始值之间的差值	
		$410 / 30s = 13.6666667$	将差值除以斜坡时间	
		$13.6666667 / 20kHz = 0.00068333$ 计数/周期	计算每个占空比周期内占空比计数值（10 位值）的变化量 (SLOPE_VAL)	
		$0.00068333 \times 1,048,576 = 716.52 \Rightarrow 717$	根据当前斜率标度因子计算小数部分值。舍入到最接近的整数	d717 或 0x2CD

表 7-43. 斜坡 2：以 5%/s 的速率从 50% 至 65%

步骤	参数	示例	说明	值
1	起始值	-	无需计算，因为我们将设置“使用当前 PWM 值”标志	-
2	停止值	$65\% \times 1024 = 665.6 \Rightarrow 666$	将结束占空比转换为 10 位值	d666 或 0x29A



**表 7-43. 斜坡 2 : 以 5%/s 的速率从 50% 至 65% (续)**

步骤	参数	示例	说明	值
3	斜率计算	$(5\% \times 1024) / 1 \text{ 秒} = 51.2 \text{ 计数/秒}$	将 %/s 斜率转换为计数/秒	
		$51.2 / 20\text{kHz} = 0.00256 \text{ 计数/周期}$	除以开关频率, 得到每个周期的计数变化量	
		$0.00256 \times 1,048,576 = 2684.35 \Rightarrow 2684$	根据当前斜率标度因子计算小数部分值。舍入到最接近的整数值	d2684 或 0xA7C

**表 7-44. 斜坡 3 : 0.5 秒内从 50 % 至 30 %**

步骤	参数	示例	说明	值
1	起始值	$50\% \times 1024 = 512$	将起始占空比转换为 10 位值	d512 或 0x200
2	停止值	$30\% \times 1024 = 307.2 \Rightarrow 307$	将结束占空比转换为 10 位值	d307 或 0x133
3	斜率计算	$512 - 307 = 205$	计算结束值与起始值之间的差值	
		$205 / 0.5\text{s} = 410 \text{ 计数/秒}$	将差值除以斜坡时间	
		$410 / 20\text{kHz} = 0.0205 \text{ 计数/周期}$	将计数/秒转换为计数/周期	
		$0.0205 \times 1,048,576 = 21498.8 \Rightarrow 21499$	根据当前斜率标度因子计算小数部分值。舍入到最接近的整数值	d21499 或 0x53FB

**表 7-45. 斜坡 4 : 以 20%/s 的速率从 x% 至 20%**

步骤	参数	示例	说明	值
1	起始值	-	不用于停止斜坡。始终使用当前值。	-
2	停止值	$20\% \times 1024 = 204.8$	将结束占空比转换为 10 位值	d205 或 0x0CD
3	斜率计算	$20\% / \text{s} \times 1024 = 204.8 \text{ 计数/秒}$	将斜率转换为 10 位值	
		$204.8 / 20\text{kHz} = 0.01024 \text{ 计数/周期}$	将斜率转换为每周期计数	
		$0.01024 \times 1,048,576 = 10737.4 \Rightarrow 10737$	根据当前斜率标度因子计算小数部分值。舍入到最接近的整数值	d10737 或 0x29F1

### 7.5.7.7 频率斜坡示例

本示例为配置具有以下参数的开关频率斜坡的基础示例。

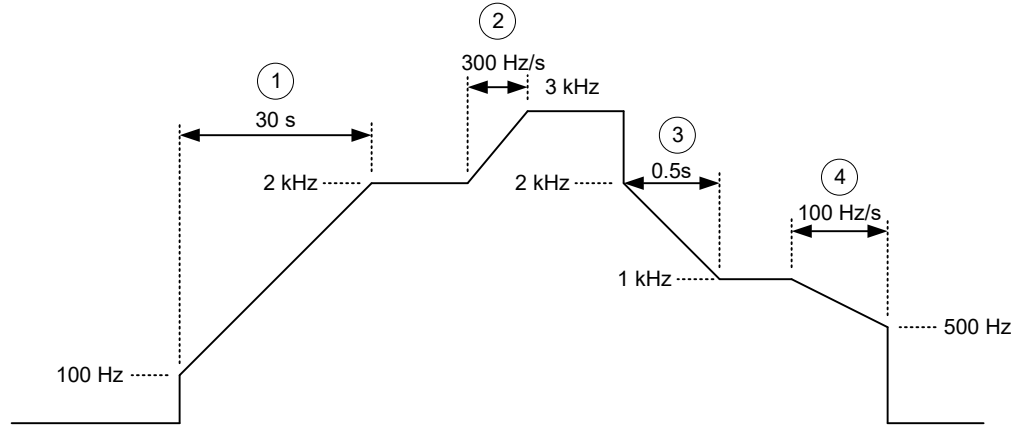


图 7-17. PWM 开关频率斜坡示例

表 7-46. 设计参数和假设

参数	值
开关频率	20kHz
输出分辨率	8 位
斜率标度	0b000 (1/65536)

上图所示的斜坡曲线可分解为 4 个独立的梯形斜坡。

1. 导通斜坡：在 30 秒内从 100Hz 达到 2kHz
2. 导通斜坡：以 300Hz/s 的速率斜升至 3kHz
3. 导通斜坡：0.5 秒内从 2kHz 阶跃至 1kHz
4. 关断斜坡：以 100Hz/s 的速率斜升至 500Hz

改变斜率值的目的是说明如何从绝对时间或变化率目标计算斜率值。由于本示例由 4 个独立斜坡组成，下面对每个斜坡进行详细说明。

表 7-47. 斜坡 1：在 30 秒内从 100Hz 达到 2kHz

步骤	参数	示例	说明	值
1	起始值	$40\text{MHz} / (100\text{Hz} \times 2^8) = 1562.5$	计算起始频率分频值	
		$\text{TRUNC}(1562.5) = 1562$	计算分频值的整数部分	d1562 或 0x61A
		$\text{ROUND}(0.5 \times 128) = 64$	计算分频值的小数部分	d64 或 0x40
2	停止值	$40\text{MHz} / (2\text{kHz} \times 2^8) = 78.125$	计算停止频率分频值	
		$\text{TRUNC}(78.125) = 78$	计算分频值的整数部分	d78 或 0x4E
		$\text{ROUND}(0.125 \times 128) = 16$	计算分频值的小数部分	d16 或 0x10
3	斜率计算	$(2\text{kHz} - 100\text{Hz}) / 30\text{ s} = 63.33\text{Hz/s}$	计算斜率，以 Hz/s 为单位	
		$63.33 \times 10\mu\text{s} = 0.06333\text{Hz}/10\mu\text{s}$	将斜率转换为 Hz/10μs	
		$\text{TRUNC}(0.06333) = 0$	计算斜率的整数值	d0 或 0x00
		$\text{ROUND}(0.06333 \times 65536) = 42$	根据斜率比例因子计算小数部分值。	d42 或 0x02A

**表 7-48. 斜坡 2：以 300Hz/s 的速率斜升至 3kHz**

步骤	参数	示例	说明	值
1	起始值	-	无需计算，因为我们将设置“使用当前 PWM 值”标志	-
2	停止值	$40\text{MHz} / (3\text{kHz} \times 2^8) = 52.083$	计算停止频率分频值	
		$\text{TRUNC}(52.083) = 52$	计算分频值的整数部分	d52 或 0x34
		$\text{ROUND}(0.083 \times 128) = 12$	计算分频值的小数部分	d12 或 0x0C
3	斜率计算	300Hz/s ( 已提供 )	计算斜率，以 Hz/s 为单位	
		$300 \times 10\mu\text{s} = 0.003\text{Hz}/10\mu\text{s}$	将斜率转换为 Hz/10μs	
		$\text{TRUNC}(0.003) = 0$	计算斜率的整数值	d0 或 0x00
		$\text{ROUND}(0.003 \times 65536) = 197$	根据斜率比例因子计算小数部分值。	d197 或 0x0C5

**表 7-49. 斜坡 3：在 0.5 秒内从 2kHz 斜升至 1kHz**

步骤	参数	示例	说明	值
1	起始值	$40\text{MHz} / (2\text{kHz} \times 2^8) = 78.125$	计算起始频率分频值	
		$\text{TRUNC}(78.125) = 78$	计算分频值的整数部分	d78 或 0x4E
		$\text{ROUND}(0.125 \times 128) = 16$	计算分频值的小数部分	d16 或 0x10
2	停止值	$40\text{MHz} / (1\text{kHz} \times 2^8) = 156.25$	计算停止频率分频值	
		$\text{TRUNC}(156.25) = 156$	计算分频值的整数部分	d156 或 0x9C
		$\text{ROUND}(0.25 \times 128) = 32$	计算分频值的小数部分	d32 或 0x20
3	斜率计算	$(2\text{kHz} - 1\text{kHz}) / 0.5\text{s} = 2000\text{Hz/s}$	计算斜率，以 Hz/s 为单位	
		$2000 \times 10\mu\text{s} = 0.02\text{Hz}/10\mu\text{s}$	将斜率转换为 Hz/10μs	
		$\text{TRUNC}(0.02) = 0$	计算斜率的整数值	d0 或 0x00
		$\text{ROUND}(0.02 \times 65536) = 1311$	根据斜率比例因子计算小数部分值。	d1311 或 0x51F

**表 7-50. 斜坡 4：以 100Hz/s 的速率斜升至 500Hz**

步骤	参数	示例	说明	值
1	起始值	-	不用于停止斜坡。始终使用当前值。	-
2	停止值	$40\text{MHz} / (500\text{Hz} \times 2^8) = 312.5$	计算停止频率分频值	
		$\text{TRUNC}(312.5) = 312$	计算分频值的整数部分	d312 或 0x138
		$\text{ROUND}(0.5 \times 128) = 64$	计算分频值的小数部分	d64 或 0x40
3	斜率计算	100Hz/s ( 已提供 )	计算斜率，以 Hz/s 为单位	
		$100 \times 10\mu\text{s} = 0.001\text{Hz}/10\mu\text{s}$	将斜率转换为 Hz/10μs	
		$\text{TRUNC}(0.001) = 0$	计算斜率的整数值	d0 或 0x00
		$\text{ROUND}(0.001 \times 65536) = 66$	根据斜率比例因子计算小数部分值。	d66 或 0x042

#### 7.5.7.8 静态开启示例

本示例是配置简单静态开启 PWM 波形的基础示例。

**表 7-51. 设计参数和假设**

参数	值
开关频率	10kHz
输出分辨率	8 位
占空比	20%

对于简单的静态 PWM 波形，仅需配置少量寄存器。所需的一般过程和寄存器为：

1. 配置寄存器必须将模式设置为静态开启，并将分辨率设置为 8 位模式
2. 对于频率，寄存器为 PWM\_END\_VAL\_CONST\_FRAC\_F、PWM\_END\_VAL\_MSB 和 PWM\_END\_VAL\_LSB
3. 对于占空比，寄存器为 PWM\_CONST\_MSB 和 PWM\_CONST\_LSB
4. 写入 PWM\_ACTION 寄存器中的启动位

表 7-52. 静态 PWM 计算

步骤	参数	示例	说明	值
1	开关频率	$40\text{MHz} / (10\text{kHz} \times 2^8) = 15.625$	计算开关频率分频值	
		$\text{TRUNC}(15.625) = 15$	计算分频值的整数部分	d15 或 0x00F
		$\text{ROUND}(0.625 \times 128) = 80$	计算分频值的小数部分	d80 或 0x50
2	占空比	$\text{ROUND}(20\% \times 256) = 51.2$	确定并获取最接近的整数占空比计数值。	d51 或 0x33

参数计算完成后，将参数写入相应寄存器，如下例所示。

表 7-53. 寄存器写入

步骤	寄存器	数据 (十六进制)	说明
1	PWM_CTRL	0x0F	设置 INIT，将 PWM 模式设置为静态开启，并使能 8 位占空比分辨率
2	PWM_END_VAL_CONST_FRAC_F	0x50	写入开关频率的小数分频值
3	PWM_END_VAL_MSB	0x00	写入开关频率分频值的 MSB
4	PWM_END_VAL_LSB	0x0F	写入开关频率分频值的 LSB
5	PWM_CONST_MSB	0x00	写入占空比计数的 MSB。由于本示例为 8 位模式，该值实际无关紧要
6	PWM_CONST_LSB	0x40	写入占空比计数的 LSB
7	PWM_CTRL	0x0E	禁用 INIT 位以使能 IP
8	PWM_ACTION	0x02	开启 PWM 输出

**备注**

PWM 模块输出必须多路复用至 IO\_CFG\_0 寄存器中的 GPIO，否则 PWM 输出不会出现在引脚上。

## 7.6 寄存器映射

TCAN5102-Q1 具有一个具有 16 位寻址功能的综合寄存器组。寄存器分为以下几个部分：

- 器件寄存器 ( 节 7.6.1 )
- SPI 寄存器 ( 节 7.6.2 和节 7.6.3 )
- UART 寄存器 ( 节 7.6.4 和节 7.6.5 )
- I2C 寄存器 ( 节 7.6.6 和节 7.6.7 )
- PWM0 寄存器 ( 节 7.6.8 )
- PWM1 寄存器 ( 节 7.6.9 )

### 7.6.1 器件寄存器

节 7.6.1 列出了器件寄存器的存储器映射寄存器。节 7.6.1 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

**表 7-54. 器件寄存器**

偏移	首字母缩写词	寄存器名称	部分
0h + 公式	DEV_ID[y]	器件 ID	节 7.6.1.1
8h	DEV_ID_REV	器件修订版本	节 7.6.1.2
9h	DEV_CMD	设备命令	节 7.6.1.3
Ah	DEV_CFG_EN	器件配置使能	节 7.6.1.4
Bh	DEV_CFG_BR	器件 CAN 波特率	节 7.6.1.5
Ch	DEV_CFG_ID_0	器件 CAN ID	节 7.6.1.6
Dh	DEV_CFG_ID_1	器件 CAN ID	节 7.6.1.7
Eh	DEV_CFG_ID_BCST_0	器件 CAN ID ( 广播 )	节 7.6.1.8
Fh	DEV_CFG_ID_BCST_1	器件 CAN ID ( 广播 )	节 7.6.1.9
10h	DEV_CFG_ID_BCST_MASK_0	器件 CAN ID ( 广播 ) 屏蔽	节 7.6.1.10
11h	DEV_CFG_ID_BCST_MASK_1	器件 CAN ID ( 广播 ) 屏蔽	节 7.6.1.11
12h	DEV_CFG_NVM_PROG_CODE	器件 NVM 编程密钥	节 7.6.1.12
13h	DEV_CFG_NVM_PROG	器件 NVM 编程	节 7.6.1.13
1Dh	INT_CFG	GPIO 中断输出配置	节 7.6.1.14
1Eh	DEV_IE_0	器件中断使能 0	节 7.6.1.15
1Fh	DEV_IE_1	器件中断使能 1	节 7.6.1.16
20h	MRAM_IP_CFG	MRAM 和 IP 配置	节 7.6.1.17
21h	IO_CFG_0	GPIO 引脚模式配置	节 7.6.1.18
22h	IO_CFG_1	GPIO 引脚模式配置	节 7.6.1.19
23h	IO_OE_0	GPIO 输出或输入配置	节 7.6.1.20
24h	IO_OE_1	GPIO 输出或输入配置	节 7.6.1.21
25h	IO_OD_0	GPIO 开漏配置	节 7.6.1.22
26h	IO_OD_1	GPIO 开漏配置	节 7.6.1.23
27h	IO_RE_0	GPIO 电阻器使能	节 7.6.1.24
28h	IO_RE_1	GPIO 电阻器使能	节 7.6.1.25
29h	IO_PU_0	GPIO 上拉电阻器使能	节 7.6.1.26
2Ah	IO_PU_1	GPIO 上拉电阻器使能	节 7.6.1.27
2Bh	IO_POL_0	GPIO 极性反转使能	节 7.6.1.28
2Ch	IO_POL_1	GPIO 极性反转使能	节 7.6.1.29
2Dh	IO_OUTPUT_0	GPIO 输出值	节 7.6.1.30
2Eh	IO_OUTPUT_1	GPIO 输出值	节 7.6.1.31

表 7-54. 器件寄存器 (续)

偏移	首字母缩写词	寄存器名称	部分
2Fh	IO_INPUT_0	GPIO 输入值	节 7.6.1.32
30h	IO_INPUT_1	GPIO 输入值	节 7.6.1.33
31h	IR_STATUS	中断状态	节 7.6.1.34
32h	DEV_IR	器件中断寄存器	节 7.6.1.35
33h	SPI_IR	SPI 中断寄存器	节 7.6.1.36
34h	UART_IR	UART 中断寄存器	节 7.6.1.37
35h	I2C_IR	I2C 中断寄存器	节 7.6.1.38
36h	PWM0_IR	PWM0 中断寄存器	节 7.6.1.39
37h	PWM1_IR	PWM1 中断寄存器	节 7.6.1.40

复杂的位访问类型经过编码可适应小型表单元。节 7.6.1 展示了适用于此部分中访问类型的代码。

表 7-55. 器件访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
RH	R H	读取 由硬件置位或清零
写入类型		
W	W	写入
W1C	W 1C	写入 1 以清零
W1S	W 1S	写入 1 以进行设置
WP	W P	写入 需要特权访问
复位或默认值		
-n		复位后的值或默认值
寄存器数组变量		
i、j、k、l、m、n		当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中寄存器是一组重复寄存器的一部分。寄存器组构成分层结构，数组用公式表示。
y		当该变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。

### 7.6.1.1 DEV\_ID[y] 寄存器 ( 偏移 = 0h + formula ) [复位 = 0032303135414354h]

图 7-18 显示了 DEV\_ID[y]，表 7-56 中对其进行了介绍。

返回到[汇总表](#)。

包含 ASCII 格式部件号的器件 ID 寄存器

偏移 = 0h + (y \* 8h)；其中 y = 0h 至 7h

图 7-18. DEV\_ID[y] 寄存器

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
DEV_ID_63:56								DEV_ID_55:48							
R-0h								R-32h							
47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
DEV_ID_40:47								DEV_ID_39:32							
R-30h								R-31h							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DEV_ID_31:24								DEV_ID_23:16							
R-35h								R-41h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEV_ID_15:8								DEV_ID_7:0							
R-43h								R-54h							

表 7-56. DEV\_ID[y] 寄存器字段说明

位	字段	类型	复位	说明
63-56	DEV_ID_63:56	R	0h	空字符
55-48	DEV_ID_55:48	R	32h	2 的 ASCII
47-40	DEV_ID_40:47	R	30h	0 的 ASCII
39-32	DEV_ID_39:32	R	31h	1 的 ASCII
31-24	DEV_ID_31:24	R	35h	5 的 ASCII
23-16	DEV_ID_23:16	R	41h	A 的 ASCII
15-8	DEV_ID_15:8	R	43h	C 的 ASCII
7-0	DEV_ID_7:0	R	54h	T 的 ASCII

### 7.6.1.2 DEV\_ID\_REV 寄存器 ( 偏移 = 8h ) [复位 = 10h]

图 7-19 显示了 DEV\_ID\_REV , 表 7-57 对其进行了介绍。

返回到[汇总表](#)。

图 7-19. DEV\_ID\_REV 寄存器

7	6	5	4	3	2	1	0
DEV_ID_MAJOR				DEV_ID_MINOR			
R-1h				R-0h			

表 7-57. DEV\_ID\_REV 寄存器字段说明

位	字段	类型	复位	说明
7-4	DEV_ID_MAJOR	R	1h	器件的主修订版本
3-0	DEV_ID_MINOR	R	0h	器件的次修订版本



### 7.6.1.3 DEV\_CMD 寄存器 ( 偏移 = 9h ) [复位 = 00h]

图 7-20 显示了 DEV\_CMD , 表 7-58 对其进行了介绍。

返回到[汇总表](#)。

图 7-20. DEV\_CMD 寄存器

7	6	5	4	3	2	1	0
RSVD							GO_TO_SLEEP
R-0h							R/W1S-0h

表 7-58. DEV\_CMD 寄存器字段说明

位	字段	类型	复位	说明
7-1	RSVD	R	0h	保留
0	GO_TO_SLEEP	R/W1S	0h	设置此位以进入睡眠模式 0h = 保持正常模式 1h = 进入睡眠模式

#### 7.6.1.4 DEV\_CFG\_EN 寄存器 ( 偏移 = Ah ) [复位 = 00h]

图 7-21 显示了 DEV\_CFG\_EN , 表 7-59 对其进行了介绍。

返回到[汇总表](#)。

解锁器件配置寄存器

图 7-21. DEV\_CFG\_EN 寄存器

7	6	5	4	3	2	1	0
RSVD					CAN_ID_BCST_CCE	CAN_ID_CCE	CAN_BR_CCE
R-0h					R/W-0h	R/W-0h	R/W-0h

表 7-59. DEV\_CFG\_EN 寄存器字段说明

位	字段	类型	复位	说明
7-3	RSVD	R	0h	保留
2	CAN_ID_BCST_CCE	R/W	0h	0h = 禁用对 DEV_CFG_ID_BCST 和 DEV_CFG_ID_BCST_MASK 的更改 1h = 启用对 DEV_CFG_ID_BCST 和 DEV_CFG_ID_BCST_MASK 的更改
1	CAN_ID_CCE	R/W	0h	0h = 禁用对 DEV_CFG_ID 的更改 1h = 启用对 DEV_CFG_ID 的更改
0	CAN_BR_CCE	R/W	0h	0h = 禁用对 DEV_CFG_BR 的更改 1h = 启用对 DEV_CFG_BR 的更改

### 7.6.1.5 DEV\_CFG\_BR 寄存器 ( 偏移 = Bh ) [复位 = 10h]

图 7-22 显示了 DEV\_CFG\_BR，表 7-60 对其进行了介绍。

返回到[汇总表](#)。

器件 CAN 波特率

**备注**  
必须在 DEV\_CFG\_EN 中启用更改使能后才能进行更改

图 7-22. DEV\_CFG\_BR 寄存器

7	6	5	4	3	2	1	0
RSVD			CAN_ACK_EN	RSVD	CAN_BR		
R-0h			R/WP-1h	R-0h	R/WP-0h		

表 7-60. DEV\_CFG\_BR 寄存器字段说明

位	字段	类型	复位	说明
7-5	RSVD	R	0h	保留
4	CAN_ACK_EN	R/WP	1h	选择是否在接收消息 ( 包括广播消息 ) 时启用 ACK 位。 对于 2M 或 5M，不建议启用此功能，但在速度为 1M 或更低时可选。 0h = ACK 位禁用 1h = ACK 位启用
3	RSVD	R	0h	保留
2-0	CAN_BR	R/WP	0h	选择 CAN 波特率。 变化在响应者发送确认帧后立即发生 0h = 250k 1h = 500k 2h = 1M 3h = 2M ( ACK 位禁用 ) 4h = 5M ( ACK 位禁用 )

#### 7.6.1.6 DEV\_CFG\_ID\_0 寄存器 ( 偏移 = Ch ) [复位 = 00h]

图 7-23 显示了 DEV\_CFG\_ID\_0，表 7-61 中对此进行了介绍。

返回到[汇总表](#)。

器件 CAN ID，响应者将侦听

#### 备注

必须在 DEV\_CFG\_EN 中启用更改使能后才能进行更改

图 7-23. DEV\_CFG\_ID\_0 寄存器

7	6	5	4	3	2	1	0
RSVD					CAN_ID_10:8		
R-0h					R/WP-0h		

表 7-61. DEV\_CFG\_ID\_0 寄存器字段说明

位	字段	类型	复位	说明
7-3	RSVD	R	0h	保留
2-0	CAN_ID_10:8	R/WP	0h	CAN ID[10:8]，用于向此特定响应者发送请求。 必须完全匹配它，才能使响应者回复。 对于请求，LSB 被忽略，始终为 0，响应者将该位设置为 1 以进行响应。

#### 7.6.1.7 DEV\_CFG\_ID\_1 寄存器 ( 偏移 = Dh ) [复位 = 00h]

图 7-24 显示了 DEV\_CFG\_ID\_1，表 7-62 中对此进行了介绍。

返回到[汇总表](#)。

器件 CAN ID，响应者将侦听。

**备注**  
必须在 DEV\_CFG\_EN 中启用更改使能后才能进行更改

图 7-24. DEV\_CFG\_ID\_1 寄存器

7	6	5	4	3	2	1	0
CAN_ID_7:1							RSVD
R/WP-0h							R-0h

表 7-62. DEV\_CFG\_ID\_1 寄存器字段说明

位	字段	类型	复位	说明
7-1	CAN_ID_7:1	R/WP	0h	用于发送给此特定响应者的请求的 CAN ID。 必须完全匹配它，才能使响应者回复。 对于请求，LSB 被忽略，始终为 0，响应者将该位设置为 1 以进行响应。
0	RSVD	R	0h	由于 0 用于请求，1 用于响应帧，因此忽略 CAN ID 的 LSB

7.6.1.8 DEV\_CFG\_ID\_BCST\_0 寄存器 ( 偏移 = Eh ) [复位 = 00h]

图 7-25 显示了 DEV\_CFG\_ID\_BCST\_0，表 7-63 中对此进行了介绍。

返回到[汇总表](#)。

响应器监听的器件广播 CAN ID。器件不会向广播消息发送响应帧

备注

必须在 DEV\_CFG\_EN 中启用更改使能才能进行更改

图 7-25. DEV\_CFG\_ID\_BCST\_0 寄存器

7	6	5	4	3	2	1	0
RSVD					CAN_ID_BCST_10:8		
R-0h					R/WP-0h		

表 7-63. DEV\_CFG\_ID\_BCST\_0 寄存器字段说明

位	字段	类型	复位	说明
7-3	RSVD	R	0h	保留
2-0	CAN_ID_BCST_10:8	R/WP	0h	用于发送给此特定响应器的请求的 CAN ID。 此 ID 与掩码寄存器配合使用，以确定哪些位是否必须匹配

#### 7.6.1.9 DEV\_CFG\_ID\_BCST\_1 寄存器 ( 偏移 = Fh ) [复位 = 00h]

图 7-26 显示了 DEV\_CFG\_ID\_BCST\_1，表 7-64 中对此进行了介绍。

返回到[汇总表](#)。

器件广播 CAN ID，响应者将侦听。器件将不会向广播消息发送响应帧

#### 备注

必须在 DEV\_CFG\_EN 中启用更改使能后才能进行更改

图 7-26. DEV\_CFG\_ID\_BCST\_1 寄存器

7	6	5	4	3	2	1	0
CAN_ID_BCST_7:0							
R/WP-0h							

表 7-64. DEV\_CFG\_ID\_BCST\_1 寄存器字段说明

位	字段	类型	复位	说明
7-0	CAN_ID_BCST_7:0	R/WP	0h	用于发送给此特定响应者的请求的 CAN ID。 此 ID 结合屏蔽寄存器，用于确定哪些位必须匹配或不匹配

**7.6.1.10 DEV\_CFG\_ID\_BCST\_MASK\_0 寄存器 ( 偏移 = 10h ) [复位 = 00h]**

图 7-27 显示了 DEV\_CFG\_ID\_BCST\_MASK\_0，表 7-65 中对此进行了介绍。

返回到[汇总表](#)。

响应器将监听的器件广播 CAN ID 掩码。器件不会向广播消息发送响应帧

**备注**

必须在 DEV\_CFG\_EN 中启用更改使能才能进行更改

**图 7-27. DEV\_CFG\_ID\_BCST\_MASK\_0 寄存器**

7	6	5	4	3	2	1	0
RSVD				CAN_ID_BCST_MSK_10:8			
R-0h				R/WP-0h			

**表 7-65. DEV\_CFG\_ID\_BCST\_MASK\_0 寄存器字段说明**

位	字段	类型	复位	说明
7-3	RSVD	R	0h	保留
2-0	CAN_ID_BCST_MSK_10:8	R/WP	0h	该掩码与 CAN_ID_BCST 一起使用，以创建广播 ID 可匹配的多个 ID。 “0”表示该位必须与 CAN_ID_BCST 中提供的值匹配。 “1”表示该位为“无关位”，即 CAN ID 中该位为 0 或 1 均可被接受。 所有位均为“0”，则广播 ID 必须与 CAN_ID_BCST 字段中输入的值完全匹配。



#### 7.6.1.11 DEV\_CFG\_ID\_BCST\_MASK\_1 寄存器 ( 偏移 = 11h ) [复位 = 00h]

图 7-28 显示了 DEV\_CFG\_ID\_BCST\_MASK\_1，表 7-66 中对此进行了介绍。

返回到[汇总表](#)。

响应器将监听的器件广播 CAN ID 掩码。器件不会向广播消息发送响应帧

#### 备注

必须在 DEV\_CFG\_EN 中启用更改使能才能进行更改

图 7-28. DEV\_CFG\_ID\_BCST\_MASK\_1 寄存器

7	6	5	4	3	2	1	0
CAN_ID_BCST_MSK_7:0							
R/WP-0h							

表 7-66. DEV\_CFG\_ID\_BCST\_MASK\_1 寄存器字段说明

位	字段	类型	复位	说明
7-0	CAN_ID_BCST_MSK_7:0	R/WP	0h	该掩码与 CAN_ID_BCST 一起使用，以创建广播 ID 可匹配的多个 ID。 “0”表示该位必须与 CAN_ID_BCST 中提供的值匹配。 “1”表示该位为“无关位”，即 CAN ID 中该位为 0 或 1 均可被接受。 所有位均为“0”，则广播 ID 必须与 CAN_ID_BCST 字段中输入的值完全匹配。

7.6.1.12 DEV\_CFG\_NVM\_PROG\_CODE 寄存器 ( 偏移 = 12h ) [复位 = 00h]

图 7-29 显示了 DEV\_CFG\_NVM\_PROG\_CODE ，表 7-67 对其进行了介绍。

返回到[汇总表](#)。

在尝试对 EEPROM 进行编程之前，必须对该寄存器写入，才能解锁该过程。

图 7-29. DEV\_CFG\_NVM\_PROG\_CODE 寄存器

7	6	5	4	3	2	1	0
PROG_UNLOCK							
R/W-0h							

表 7-67. DEV\_CFG\_NVM\_PROG\_CODE 寄存器字段说明

位	字段	类型	复位	说明
7-0	PROG_UNLOCK	R/W	0h	要刷写 EEPROM，该字段必须写为 0x78。 其他任何值都会导致闪存请求被忽略。

### 7.6.1.13 DEV\_CFG\_NVM\_PROG 寄存器 ( 偏移 = 13h ) [复位 = 00h]

DEV\_CFG\_NVM\_PROG 寄存器的相关信息如表 7-68 所示。

返回到[汇总表](#)。

将现有器件配置保存到器件的一次性可编程存储器中。

**表 7-68. DEV\_CFG\_NVM\_PROG 寄存器字段说明**

位	字段	类型	复位	说明
6	PROG_FAIL	RH	0h	用于报告 EEPROM 编程过程是否存在故障的状态标志。 尝试编程 EEPROM 后应检查此标志。 如果发生任何类型的故障，此位将置位，用户应尝试重新编程 EEPROM。 0h = EEPROM 无故障 1h = EEPROM 编程失败
5	RSVD	R	0h	保留
4	EEPROM_FLASHED	RH	0h	查询 EEPROM 是否已编程的状态标志 0h = EEPROM 未进行编程 1h = EEPROM 已编程
3	RSVD	R	0h	保留
2	LOCK_ID	RH/W1S	0h	若此位置位，器件将永久禁用 CAN_ID_CCE 位，使 CAN ID 和掩码无法更改。 0h = 可以更新 CAN ID 和掩码寄存器 ( 寄存器和 EEPROM 中均如此 ) 1h = 无法更新 CAN ID 和掩码寄存器 ( 寄存器和 EEPROM 中均如此 )
1	LOCK_BR	RH/W1S	0h	如果此位置位，器件将永久禁用 CAN_BR_CCE 位，使 CAN 波特率无法更改。 0h = 可以更新 CAN 数据速率寄存器 ( 寄存器和 EEPROM 中均如此 ) 1h = 无法更新 CAN 数据速率寄存器 ( 寄存器和 EEPROM 中均如此 )
0	PROG	RH/W1S	0h	如果已置位且闪存代码设置正确，则 EEPROM 将被编程。 在闪存过程完成之前，该位将读回 1。 0h = 未在主动对 EEPROM 进行编程/编程已完成 1h = 正在主动对 EEPROM 进行编程

#### 7.6.1.14 INT\_CFG 寄存器 ( 偏移 = 1Dh ) [复位 = 00h]

图 7-30 中显示了 INT\_CFG，表 7-69 中对此进行了介绍。

返回到[汇总表](#)。

GPIO 中断输出

图 7-30. INT\_CFG 寄存器

7	6	5	4	3	2	1	0
RSVD	INT1_SEL		INT1_EN	RSVD	INT0_SEL		INT0_EN
R-0h	R/W-0h		R/W-0h	R-0h	R/W-0h		R/W-0h

表 7-69. INT\_CFG 寄存器字段说明

位	字段	类型	复位	说明
7	RSVD	R	0h	保留
6-5	INT1_SEL	R/W	0h	选择用作中断输出的 GPIO 引脚 0h = GPIO1 作为低电平有效 INT1 1h = GPIO8 作为低电平有效 INT1 2h = GPIO1 作为高电平有效 INT1 3h = GPIO8 作为高电平有效 INT1
4	INT1_EN	R/W	0h	用于使能选定的 GPIO 作为中断输出。 0h = 正常功能 ( 按照配置 ) 1h = 中断功能 ( IO_CFG 必须设置为 GPIO 功能 )
3	RSVD	R	0h	保留
2-1	INT0_SEL	R/W	0h	选择用作中断输出的 GPIO 引脚 0h = GPIO0 作为低电平有效 INT0 1h = GPIO7 作为低电平有效 INT0 2h = GPIO0 作为高电平有效 INT0 3h = GPIO7 作为高电平有效 INT0
0	INT0_EN	R/W	0h	用于使能选定的 GPIO 作为中断输出。 0h = 正常功能 ( 按照配置 ) 1h = 中断功能 ( IO_CFG 必须设置为 GPIO 功能 )

#### 7.6.1.15 DEV\_IE\_0 寄存器 ( 偏移 = 1Eh ) [复位 = 00h]

图 7-31 显示了 DEV\_IE\_0，表 7-70 中对此进行了介绍。

返回到[汇总表](#)。

用于为 INT0 引脚启用特定中断的中断使能位。已启用的中断将在 INT0 引脚上发出信号。请注意，若要在 INT0 引脚上观察到中断输出信号，必须先在 INT\_CFG 寄存器中启用 INT0 功能。

图 7-31. DEV\_IE\_0 寄存器

7	6	5	4	3	2	1	0
RSVD						CANWK	RSVD
R-0h						R/W-0h	R-0h

表 7-70. DEV\_IE\_0 寄存器字段说明

位	字段	类型	复位	说明
7-2	RSVD	R	0h	保留
1	CANWK	R/W	0h	CAN 唤醒 器件因 CAN 总线而退出睡眠模式 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
0	RSVD	R	0h	保留

**7.6.1.16 DEV\_IE\_1 寄存器 ( 偏移 = 1Fh ) [复位 = 00h]**

图 7-32 显示了 DEV\_IE\_1，表 7-71 中对此进行了介绍。

返回到[汇总表](#)。

用于为 INT0 引脚启用特定中断的中断使能位。已启用的中断将在 INT1 引脚上发出信号。请注意，若要在 INT0 引脚上观察到中断输出信号，必须先在 INT\_CFG 寄存器中启用 INT0 功能。

**图 7-32. DEV\_IE\_1 寄存器**

7	6	5	4	3	2	1	0
RSVD						CANWK	RSVD
R-0h						R/W-0h	R-0h

**表 7-71. DEV\_IE\_1 寄存器字段说明**

位	字段	类型	复位	说明
7-2	RSVD	R	0h	保留
1	CANWK	R/W	0h	CAN 唤醒 器件因 CAN 总线而退出睡眠模式 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
0	RSVD	R	0h	保留

### 7.6.1.17 MRAM\_IP\_CFG 寄存器 ( 偏移 = 20h ) [复位 = 00h]

图 7-33 显示了 MRAM\_IP\_CFG 寄存器，表 7-72 中对此进行了介绍。

返回到[汇总表](#)。

MRAM 和 IP 配置。该寄存器用于使能各种串行接口，并为其分配部分 MRAM。

图 7-33. MRAM\_IP\_CFG 寄存器

7	6	5	4	3	2	1	0
RSVD			GPIO_OUT_SY NC	MRAM_IP_EN			
R-0h			R/W-0h		R/W-0h		

表 7-72. MRAM\_IP\_CFG 寄存器字段说明

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4	GPIO_OUT_SYNC	R/W	0h	通过延迟更新直到第二个 GPIO 寄存器被写入，来同步 GPIO 输出值的更新。 这要求所有输出 GPIO 的更改都需写入 0x002D 和 0x002E，且一旦 0x002E 写入完成，所有 GPIO 将更新。 0h = GPIO 在字节写入时更新 1h = GPIO 在写入 0x2E 后更新
3-0	MRAM_IP_EN	R/W	0h	MRAM 和 IP 使能。 如果为串行接口分配了任何内存 (>0%)，这些位将使能所选 IP。 当已使能的 IP 处于活动状态时，请勿更改此值，否则可能发生未定义行为 ( 未来版本将修改此特性 )  <div style="text-align: center;"><b>备注</b></div> <div style="text-align: center;">目标 IP 的 GPIO 需设置为特殊功能，否则 IP 将无法使能</div> 0h = 全部禁用/0% 1h = SPI 0%，UART 100%，I2C 0% 2h = SPI 25%，UART 75%，I2C 0% 3h = SPI 50%，UART 50%，I2C 0% 4h = SPI 75%，UART 25%，I2C 0% 5h = SPI 100%，UART 0%，I2C 0% 6h = SPI 0%，UART 0%，I2C 100% 7h = SPI 0%，UART 25%，I2C 75% 8h = SPI 0%，UART 50%，I2C 50% 9h = SPI 0%，UART 75%，I2C 25% Ah = SPI 25%，UART 0%，I2C 75% Bh = SPI 50%，UART 0%，I2C 50% Ch = SPI 75%，UART 0%，I2C 25%



### 7.6.1.18 IO\_CFG\_0 寄存器 (偏移 = 21h) [复位 = 00h]

图 7-34 显示了 IO\_CFG\_0，表 7-73 中对此进行了介绍。

返回到[汇总表](#)。

GPIO 引脚模式配置

图 7-34. IO\_CFG\_0 寄存器

7	6	5	4	3	2	1	0
GPIO7_CFG	GPIO6_CFG	GPIO5_CFG	GPIO4_CFG	GPIO3_CFG	GPIO2_CFG	GPIO1_CFG	GPIO0_CFG
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-73. IO\_CFG\_0 寄存器字段说明

位	字段	类型	复位	说明
7	GPIO7_CFG	R/W	0h	0h = GPIO 功能 1h = 特殊功能 (UART TXD)
6	GPIO6_CFG	R/W	0h	0h = GPIO 功能 1h = 特殊功能 (SPI PICO)
5	GPIO5_CFG	R/W	0h	0h = GPIO 功能 1h = 特殊功能 (SPI POCI)
4	GPIO4_CFG	R/W	0h	0h = GPIO 功能 1h = 特殊功能 (SPI SCK)
3	GPIO3_CFG	R/W	0h	0h = GPIO 功能 1h = 特殊功能 (SPI CS0)
2	GPIO2_CFG	R/W	0h	0h = GPIO 功能 1h = 特殊功能 (SPI CS1)
1	GPIO1_CFG	R/W	0h	0h = GPIO 功能 1h = 特殊功能 (SPI CS2)
0	GPIO0_CFG	R/W	0h	0h = GPIO 功能 1h = 特殊功能 (SPI CS3)

### 7.6.1.19 IO\_CFG\_1 寄存器 ( 偏移 = 22h ) [复位 = 00h]

图 7-35 显示了 IO\_CFG\_1，表 7-74 中对此进行了介绍。

返回到[汇总表](#)。

GPIO 引脚模式配置

图 7-35. IO\_CFG\_1 寄存器

7	6	5	4	3	2	1	0
RSVD			GPIO12_CFG	GPIO11_CFG	GPIO10_CFG	GPIO9_CFG	GPIO8_CFG
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-74. IO\_CFG\_1 寄存器字段说明

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4	GPIO12_CFG	R/W	0h	0h = GPIO 功能 1h = 特殊功能 ( 如果启用了 PWM1，则为 SPI CS7 或 PWM1 )
3	GPIO11_CFG	R/W	0h	0h = GPIO 功能 1h = 特殊功能 ( 如果启用了 PWM0，则为 SPI CS6 或 PWM0 )
2	GPIO10_CFG	R/W	0h	0h = GPIO 功能 1h = 特殊功能 ( 如果启用了 I2C，则为 SPI CS5 或 I2C SDA )
1	GPIO9_CFG	R/W	0h	0h = GPIO 功能 1h = 特殊功能 ( 如果启用了 I2C，则为 SPI CS4 或 I2C SCL )
0	GPIO8_CFG	R/W	0h	0h = GPIO 功能 1h = 特殊功能 (UART RXD)

**7.6.1.20 IO\_OE\_0 寄存器 ( 偏移 = 23h ) [复位 = 00h]**

图 7-36 显示了 IO\_OE\_0，表 7-75 中对此进行了介绍。

返回到[汇总表](#)。

GPIO 输出或输入配置

**图 7-36. IO\_OE\_0 寄存器**

7	6	5	4	3	2	1	0
GPIO7_OE	GPIO6_OE	GPIO5_OE	GPIO4_OE	GPIO3_OE	GPIO2_OE	GPIO1_OE	GPIO0_OE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**表 7-75. IO\_OE\_0 寄存器字段说明**

位	字段	类型	复位	说明
7	GPIO7_OE	R/W	0h	0h = 输入 1h = 输出
6	GPIO6_OE	R/W	0h	0h = 输入 1h = 输出
5	GPIO5_OE	R/W	0h	0h = 输入 1h = 输出
4	GPIO4_OE	R/W	0h	0h = 输入 1h = 输出
3	GPIO3_OE	R/W	0h	0h = 输入 1h = 输出
2	GPIO2_OE	R/W	0h	0h = 输入 1h = 输出
1	GPIO1_OE	R/W	0h	0h = 输入 1h = 输出
0	GPIO0_OE	R/W	0h	0h = 输入 1h = 输出

### 7.6.1.21 IO\_OE\_1 寄存器 ( 偏移 = 24h ) [复位 = 00h]

图 7-37 显示了 IO\_OE\_1，表 7-76 中对此进行了介绍。

返回到[汇总表](#)。

GPIO 输出或输入配置

图 7-37. IO\_OE\_1 寄存器

7	6	5	4	3	2	1	0
RSVD			GPIO12_OE	GPIO11_OE	GPIO10_OE	GPIO9_OE	GPIO8_OE
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-76. IO\_OE\_1 寄存器字段说明

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4	GPIO12_OE	R/W	0h	0h = 输入 1h = 输出
3	GPIO11_OE	R/W	0h	0h = 输入 1h = 输出
2	GPIO10_OE	R/W	0h	0h = 输入 1h = 输出
1	GPIO9_OE	R/W	0h	0h = 输入 1h = 输出
0	GPIO8_OE	R/W	0h	0h = 输入 1h = 输出

**7.6.1.22 IO\_OD\_0 寄存器 ( 偏移 = 25h ) [复位 = 00h]**

图 7-38 显示了 IO\_OD\_0，表 7-77 中对此进行了介绍。

返回到[汇总表](#)。

GPIO 开漏配置

**图 7-38. IO\_OD\_0 寄存器**

7	6	5	4	3	2	1	0
GPIO7_OD	GPIO6_OD	GPIO5_OD	GPIO4_OD	GPIO3_OD	GPIO2_OD	GPIO1_OD	GPIO0_OD
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**表 7-77. IO\_OD\_0 寄存器字段说明**

位	字段	类型	复位	说明
7	GPIO7_OD	R/W	0h	0h = 推挽模式 1h = 开漏模式
6	GPIO6_OD	R/W	0h	0h = 推挽模式 1h = 开漏模式
5	GPIO5_OD	R/W	0h	0h = 推挽模式 1h = 开漏模式
4	GPIO4_OD	R/W	0h	0h = 推挽模式 1h = 开漏模式
3	GPIO3_OD	R/W	0h	0h = 推挽模式 1h = 开漏模式
2	GPIO2_OD	R/W	0h	0h = 推挽模式 1h = 开漏模式
1	GPIO1_OD	R/W	0h	0h = 推挽模式 1h = 开漏模式
0	GPIO0_OD	R/W	0h	0h = 推挽模式 1h = 开漏模式

### 7.6.1.23 IO\_OD\_1 寄存器 ( 偏移 = 26h ) [复位 = 00h]

图 7-39 显示了 IO\_OD\_1，表 7-78 中对此进行了介绍。

返回到[汇总表](#)。

GPIO 开漏配置

图 7-39. IO\_OD\_1 寄存器

7	6	5	4	3	2	1	0
RSVD			GPIO12_OD	GPIO11_OD	GPIO10_OD	GPIO9_OD	GPIO8_OD
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-78. IO\_OD\_1 寄存器字段说明

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4	GPIO12_OD	R/W	0h	0h = 推挽模式 1h = 开漏模式
3	GPIO11_OD	R/W	0h	0h = 推挽模式 1h = 开漏模式
2	GPIO10_OD	R/W	0h	0h = 推挽模式 1h = 开漏模式
1	GPIO9_OD	R/W	0h	0h = 推挽模式 1h = 开漏模式
0	GPIO8_OD	R/W	0h	0h = 推挽模式 1h = 开漏模式

### 7.6.1.24 IO\_RE\_0 寄存器 ( 偏移 = 27h ) [复位 = 00h]

图 7-40 显示了 IO\_RE\_0，表 7-79 中对此进行了介绍。

返回到[汇总表](#)。

GPIO 电阻器使能

图 7-40. IO\_RE\_0 寄存器

7	6	5	4	3	2	1	0
GPIO7_RE	GPIO6_RE	GPIO5_RE	GPIO4_RE	GPIO3_RE	GPIO2_RE	GPIO1_RE	GPIO0_RE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-79. IO\_RE\_0 寄存器字段说明

位	字段	类型	复位	说明
7	GPIO7_RE	R/W	0h	0h = 引脚上未连接偏置电阻器 1h = 根据 IO_PU 为引脚连接偏置电阻器
6	GPIO6_RE	R/W	0h	0h = 引脚上未连接偏置电阻器 1h = 根据 IO_PU 为引脚连接偏置电阻器
5	GPIO5_RE	R/W	0h	0h = 引脚上未连接偏置电阻器 1h = 根据 IO_PU 为引脚连接偏置电阻器
4	GPIO4_RE	R/W	0h	0h = 引脚上未连接偏置电阻器 1h = 根据 IO_PU 为引脚连接偏置电阻器
3	GPIO3_RE	R/W	0h	0h = 引脚上未连接偏置电阻器 1h = 根据 IO_PU 为引脚连接偏置电阻器
2	GPIO2_RE	R/W	0h	0h = 引脚上未连接偏置电阻器 1h = 根据 IO_PU 为引脚连接偏置电阻器
1	GPIO1_RE	R/W	0h	0h = 引脚上未连接偏置电阻器 1h = 根据 IO_PU 为引脚连接偏置电阻器
0	GPIO0_RE	R/W	0h	0h = 引脚上未连接偏置电阻器 1h = 根据 IO_PU 为引脚连接偏置电阻器

### 7.6.1.25 IO\_RE\_1 寄存器 ( 偏移 = 28h ) [复位 = 00h]

图 7-41 显示了 IO\_RE\_1，表 7-80 中对此进行了介绍。

返回到[汇总表](#)。

GPIO 电阻器使能

图 7-41. IO\_RE\_1 寄存器

7	6	5	4	3	2	1	0
RSVD			GPIO12_RE	GPIO11_RE	GPIO10_RE	GPIO9_RE	GPIO8_RE
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-80. IO\_RE\_1 寄存器字段说明

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4	GPIO12_RE	R/W	0h	0h = 引脚上未连接偏置电阻器 1h = 根据 IO_PU 为引脚连接偏置电阻器
3	GPIO11_RE	R/W	0h	0h = 引脚上未连接偏置电阻器 1h = 根据 IO_PU 为引脚连接偏置电阻器
2	GPIO10_RE	R/W	0h	0h = 引脚上未连接偏置电阻器 1h = 根据 IO_PU 为引脚连接偏置电阻器
1	GPIO9_RE	R/W	0h	0h = 引脚上未连接偏置电阻器 1h = 根据 IO_PU 为引脚连接偏置电阻器
0	GPIO8_RE	R/W	0h	0h = 引脚上未连接偏置电阻器 1h = 根据 IO_PU 为引脚连接偏置电阻器



### 7.6.1.26 IO\_PU\_0 寄存器 ( 偏移 = 29h ) [复位 = 00h]

图 7-42 显示了 IO\_PU\_0，表 7-81 中对此进行了介绍。

返回到[汇总表](#)。

GPIO 上拉电阻器使能

图 7-42. IO\_PU\_0 寄存器

7	6	5	4	3	2	1	0
GPIO7_PU	GPIO6_PU	GPIO5_PU	GPIO4_PU	GPIO3_PU	GPIO2_PU	GPIO1_PU	GPIO0_PU
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-81. IO\_PU\_0 寄存器字段说明

位	字段	类型	复位	说明
7	GPIO7_PU	R/W	0h	0h = 如果 IO_RE 置位，引脚接下拉电阻器 1h = 如果 IO_RE 置位，引脚接上拉电阻器
6	GPIO6_PU	R/W	0h	0h = 如果 IO_RE 置位，引脚接下拉电阻器 1h = 如果 IO_RE 置位，引脚接上拉电阻器
5	GPIO5_PU	R/W	0h	0h = 如果 IO_RE 置位，引脚接下拉电阻器 1h = 如果 IO_RE 置位，引脚接上拉电阻器
4	GPIO4_PU	R/W	0h	0h = 如果 IO_RE 置位，引脚接下拉电阻器 1h = 如果 IO_RE 置位，引脚接上拉电阻器
3	GPIO3_PU	R/W	0h	0h = 如果 IO_RE 置位，引脚接下拉电阻器 1h = 如果 IO_RE 置位，引脚接上拉电阻器
2	GPIO2_PU	R/W	0h	0h = 如果 IO_RE 置位，引脚接下拉电阻器 1h = 如果 IO_RE 置位，引脚接上拉电阻器
1	GPIO1_PU	R/W	0h	0h = 如果 IO_RE 置位，引脚接下拉电阻器 1h = 如果 IO_RE 置位，引脚接上拉电阻器
0	GPIO0_PU	R/W	0h	0h = 如果 IO_RE 置位，引脚接下拉电阻器 1h = 如果 IO_RE 置位，引脚接上拉电阻器

### 7.6.1.27 IO\_PU\_1 寄存器 ( 偏移 = 2Ah ) [复位 = 00h]

图 7-43 显示了 IO\_PU\_1，表 7-82 中对此进行了介绍。

返回到[汇总表](#)。

GPIO 上拉电阻器使能

**图 7-43. IO\_PU\_1 寄存器**

7	6	5	4	3	2	1	0
RSVD			GPIO12_PU	GPIO11_PU	GPIO10_PU	GPIO9_PU	GPIO8_PU
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**表 7-82. IO\_PU\_1 寄存器字段说明**

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4	GPIO12_PU	R/W	0h	0h = 如果 IO_RE 置位，引脚接下拉电阻器 1h = 如果 IO_RE 置位，引脚接上拉电阻器
3	GPIO11_PU	R/W	0h	0h = 如果 IO_RE 置位，引脚接下拉电阻器 1h = 如果 IO_RE 置位，引脚接上拉电阻器
2	GPIO10_PU	R/W	0h	0h = 如果 IO_RE 置位，引脚接下拉电阻器 1h = 如果 IO_RE 置位，引脚接上拉电阻器
1	GPIO9_PU	R/W	0h	0h = 如果 IO_RE 置位，引脚接下拉电阻器 1h = 如果 IO_RE 置位，引脚接上拉电阻器
0	GPIO8_PU	R/W	0h	0h = 如果 IO_RE 置位，引脚接下拉电阻器 1h = 如果 IO_RE 置位，引脚接上拉电阻器

### 7.6.1.28 IO\_POL\_0 寄存器 ( 偏移 = 2Bh ) [复位 = 00h]

图 7-44 显示了 IO\_POL\_0，表 7-83 中对此进行了介绍。

返回到[汇总表](#)。

GPIO 极性反转使能。这些位仅在引脚配置为输入时生效。它将反转 IO\_INPUT 寄存器中显示的输入信号

图 7-44. IO\_POL\_0 寄存器

7	6	5	4	3	2	1	0
GPIO7_PIE	GPIO6_PIE	GPIO5_PIE	GPIO4_PIE	GPIO3_PIE	GPIO2_PIE	GPIO1_PIE	GPIO0_PIE
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-83. IO\_POL\_0 寄存器字段说明

位	字段	类型	复位	说明
7	GPIO7_PIE	R/W	0h	0h = 输入极性反转功能未使能 1h = 输入极性反转功能已使能
6	GPIO6_PIE	R/W	0h	0h = 输入极性反转功能未使能 1h = 输入极性反转功能已使能
5	GPIO5_PIE	R/W	0h	0h = 输入极性反转功能未使能 1h = 输入极性反转功能已使能
4	GPIO4_PIE	R/W	0h	0h = 输入极性反转功能未使能 1h = 输入极性反转功能已使能
3	GPIO3_PIE	R/W	0h	0h = 输入极性反转功能未使能 1h = 输入极性反转功能已使能
2	GPIO2_PIE	R/W	0h	0h = 输入极性反转功能未使能 1h = 输入极性反转功能已使能
1	GPIO1_PIE	R/W	0h	0h = 输入极性反转功能未使能 1h = 输入极性反转功能已使能
0	GPIO0_PIE	R/W	0h	0h = 输入极性反转功能未使能 1h = 输入极性反转功能已使能

### 7.6.1.29 IO\_POL\_1 寄存器 ( 偏移 = 2Ch ) [复位 = 00h]

图 7-45 显示了 IO\_POL\_1，表 7-84 中对此进行了介绍。

返回到[汇总表](#)。

GPIO 极性反转使能。这些位仅在引脚配置为输入时生效。它将反转 IO\_INPUT 寄存器中显示的输入信号

图 7-45. IO\_POL\_1 寄存器

7	6	5	4	3	2	1	0
RSVD			GPIO12_PIE	GPIO11_PIE	GPIO10_PIE	GPIO9_PIE	GPIO8_PIE
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-84. IO\_POL\_1 寄存器字段说明

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4	GPIO12_PIE	R/W	0h	0h = 输入极性反转功能未使能 1h = 输入极性反转功能已使能
3	GPIO11_PIE	R/W	0h	0h = 输入极性反转功能未使能 1h = 输入极性反转功能已使能
2	GPIO10_PIE	R/W	0h	0h = 输入极性反转功能未使能 1h = 输入极性反转功能已使能
1	GPIO9_PIE	R/W	0h	0h = 输入极性反转功能未使能 1h = 输入极性反转功能已使能
0	GPIO8_PIE	R/W	0h	0h = 输入极性反转功能未使能 1h = 输入极性反转功能已使能

### 7.6.1.30 IO\_OUTPUT\_0 寄存器 ( 偏移 = 2Dh ) [复位 = 00h]

图 7-46 显示了 IO\_OUTPUT\_0，表 7-85 中对此进行了介绍。

返回到[汇总表](#)。

图 7-46. IO\_OUTPUT\_0 寄存器

7	6	5	4	3	2	1	0
GPIO7_OUTPUT T	GPIO6_OUTPUT T	GPIO5_OUTPUT T	GPIO4_OUTPUT T	GPIO3_OUTPUT T	GPIO2_OUTPUT T	GPIO1_OUTPUT T	GPIO0_OUTPUT T
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-85. IO\_OUTPUT\_0 寄存器字段说明

位	字段	类型	复位	说明
7	GPIO7_OUTPUT	R/W	0h	0h = 如果引脚配置为输出，则输出被驱动为低电平。 1h = 如果引脚配置为输出，则输出被驱动为高电平或高阻态 ( 取决于 IO_OD )
6	GPIO6_OUTPUT	R/W	0h	0h = 如果引脚配置为输出，则输出被驱动为低电平。 1h = 如果引脚配置为输出，则输出被驱动为高电平或高阻态 ( 取决于 IO_OD )
5	GPIO5_OUTPUT	R/W	0h	0h = 如果引脚配置为输出，则输出被驱动为低电平。 1h = 如果引脚配置为输出，则输出被驱动为高电平或高阻态 ( 取决于 IO_OD )
4	GPIO4_OUTPUT	R/W	0h	0h = 如果引脚配置为输出，则输出被驱动为低电平。 1h = 如果引脚配置为输出，则输出被驱动为高电平或高阻态 ( 取决于 IO_OD )
3	GPIO3_OUTPUT	R/W	0h	0h = 如果引脚配置为输出，则输出被驱动为低电平。 1h = 如果引脚配置为输出，则输出被驱动为高电平或高阻态 ( 取决于 IO_OD )
2	GPIO2_OUTPUT	R/W	0h	0h = 如果引脚配置为输出，则输出被驱动为低电平。 1h = 如果引脚配置为输出，则输出被驱动为高电平或高阻态 ( 取决于 IO_OD )
1	GPIO1_OUTPUT	R/W	0h	0h = 如果引脚配置为输出，则输出被驱动为低电平。 1h = 如果引脚配置为输出，则输出被驱动为高电平或高阻态 ( 取决于 IO_OD )
0	GPIO0_OUTPUT	R/W	0h	0h = 如果引脚配置为输出，则输出被驱动为低电平。 1h = 如果引脚配置为输出，则输出被驱动为高电平或高阻态 ( 取决于 IO_OD )

### 7.6.1.31 IO\_OUTPUT\_1 寄存器 ( 偏移 = 2Eh ) [复位 = 00h]

图 7-47 显示了 IO\_OUTPUT\_1，表 7-86 中对此进行了介绍。

返回到汇总表。

图 7-47. IO\_OUTPUT\_1 寄存器

7	6	5	4	3	2	1	0
RSVD			GPIO12_OUTP UT	GPIO11_OUTP UT	GPIO10_OUTP UT	GPIO9_OUTPU T	GPIO8_OUTPU T
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-86. IO\_OUTPUT\_1 寄存器字段说明

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4	GPIO12_OUTPUT	R/W	0h	0h = 如果引脚配置为输出，则输出被驱动为低电平。 1h = 如果引脚配置为输出，则输出被驱动为高电平或高阻态 ( 取决于 IO_OD )
3	GPIO11_OUTPUT	R/W	0h	0h = 如果引脚配置为输出，则输出被驱动为低电平。 1h = 如果引脚配置为输出，则输出被驱动为高电平或高阻态 ( 取决于 IO_OD )
2	GPIO10_OUTPUT	R/W	0h	0h = 如果引脚配置为输出，则输出被驱动为低电平。 1h = 如果引脚配置为输出，则输出被驱动为高电平或高阻态 ( 取决于 IO_OD )
1	GPIO9_OUTPUT	R/W	0h	0h = 如果引脚配置为输出，则输出被驱动为低电平。 1h = 如果引脚配置为输出，则输出被驱动为高电平或高阻态 ( 取决于 IO_OD )
0	GPIO8_OUTPUT	R/W	0h	0h = 如果引脚配置为输出，则输出被驱动为低电平。 1h = 如果引脚配置为输出，则输出被驱动为高电平或高阻态 ( 取决于 IO_OD )

### 7.6.1.32 IO\_INPUT\_0 寄存器 ( 偏移 = 2Fh ) [复位 = 00h]

图 7-48 显示了 IO\_INPUT\_0，表 7-87 中对此进行了介绍。

返回到[汇总表](#)。

图 7-48. IO\_INPUT\_0 寄存器

7	6	5	4	3	2	1	0
GPIO7_INPUT	GPIO6_INPUT	GPIO5_INPUT	GPIO4_INPUT	GPIO3_INPUT	GPIO2_INPUT	GPIO1_INPUT	GPIO0_INPUT
RH-0h	RH-0h	RH-0h	RH-0h	RH-0h	RH-0h	RH-0h	RH-0h

表 7-87. IO\_INPUT\_0 寄存器字段说明

位	字段	类型	复位	说明
7	GPIO7_INPUT	RH	0h	0h = 引脚的当前状态为低电平 1h = 引脚的当前状态为高电平
6	GPIO6_INPUT	RH	0h	0h = 引脚的当前状态为低电平 1h = 引脚的当前状态为高电平
5	GPIO5_INPUT	RH	0h	0h = 引脚的当前状态为低电平 1h = 引脚的当前状态为高电平
4	GPIO4_INPUT	RH	0h	0h = 引脚的当前状态为低电平 1h = 引脚的当前状态为高电平
3	GPIO3_INPUT	RH	0h	0h = 引脚的当前状态为低电平 1h = 引脚的当前状态为高电平
2	GPIO2_INPUT	RH	0h	0h = 引脚的当前状态为低电平 1h = 引脚的当前状态为高电平
1	GPIO1_INPUT	RH	0h	0h = 引脚的当前状态为低电平 1h = 引脚的当前状态为高电平
0	GPIO0_INPUT	RH	0h	0h = 引脚的当前状态为低电平 1h = 引脚的当前状态为高电平

### 7.6.1.33 IO\_INPUT\_1 寄存器 ( 偏移 = 30h ) [复位 = 00h]

图 7-49 显示了 IO\_INPUT\_1，表 7-88 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-49. IO\_INPUT\_1 寄存器**

7	6	5	4	3	2	1	0
RSVD			GPIO12_INPUT	GPIO11_INPUT	GPIO10_INPUT	GPIO9_INPUT	GPIO8_INPUT
R-0h			RH-0h	RH-0h	RH-0h	RH-0h	RH-0h

**表 7-88. IO\_INPUT\_1 寄存器字段说明**

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4	GPIO12_INPUT	RH	0h	0h = 引脚的当前状态为低电平 1h = 引脚的当前状态为高电平
3	GPIO11_INPUT	RH	0h	0h = 引脚的当前状态为低电平 1h = 引脚的当前状态为高电平
2	GPIO10_INPUT	RH	0h	0h = 引脚的当前状态为低电平 1h = 引脚的当前状态为高电平
1	GPIO9_INPUT	RH	0h	0h = 引脚的当前状态为低电平 1h = 引脚的当前状态为高电平
0	GPIO8_INPUT	RH	0h	0h = 引脚的当前状态为低电平 1h = 引脚的当前状态为高电平



#### 7.6.1.34 IR\_STATUS 寄存器 ( 偏移 = 31h ) [复位 = 00h]

IR\_STATUS 如表 7-89 所示。

返回到[汇总表](#)。

中断状态寄存器显示哪个 IP 模块有挂起的中断，该中断在任一中断线上使能且已置位。如果中断未使能，则不会显示该中断

表 7-89. IR\_STATUS 寄存器字段说明

位	字段	类型	复位	说明
7-5	RSVD	R	0h	保留
5	PWM1_IR	R	0h	在 PWM1 中断寄存器中置位中断 0h = PWM1 中断寄存器中无中断置位 1h = PWM1 中断寄存器中有中断置位
4	PWM0_IR	R	0h	在 PWM0 中断寄存器中置位中断 0h = PWM0 中断寄存器中无中断置位 1h = PWM0 中断寄存器中有中断置位
3	I2C_IR	R	0h	在 I2C 中断寄存器中置位中断 0h = I2C 中断寄存器中无中断置位 1h = I2C 中断寄存器中有中断置位
2	UART_IR	R	0h	在 UART 中断寄存器中置位中断 0h = UART 中断寄存器中无中断置位 1h = UART 中断寄存器中有中断置位
1	SPI_IR	R	0h	在 SPI 中断寄存器中置位中断 0h = SPI 中断寄存器中无中断置位 1h = SPI 中断寄存器中有中断置位
0	DEVICE_IR	R	0h	在器件中断寄存器中置位中断 0h = 器件中断寄存器中无中断置位 1h = 器件中断寄存器中有中断置位

### 7.6.1.35 DEV\_IR 寄存器 ( 偏移 = 32h ) [复位 = 00h]

图 7-50 显示了 DEV\_IR，表 7-90 对其进行了介绍。

返回到汇总表。

器件中断状态寄存器

图 7-50. DEV\_IR 寄存器

7	6	5	4	3	2	1	0
RSVD						CANWK	RSVD
R-0h						R/W1C-0h	R-0h

表 7-90. DEV\_IR 寄存器字段说明

位	字段	类型	复位	说明
7-2	RSVD	R	0h	保留
1	CANWK	R/W1C	0h	CAN 唤醒 器件由于 CAN 总线而退出睡眠模式 0h = 未发生中断 1h = 已发生中断
0	RSVD	R	0h	保留

### 7.6.1.36 SPI\_IR 寄存器 ( 偏移 = 33h ) [复位 = 00h]

图 7-51 显示了 SPI\_IR 寄存器，表 7-91 中对此进行了介绍。

返回到[汇总表](#)。

这是 SPI\_IR 寄存器的可写镜像，便于读取和清除中断

图 7-51. SPI\_IR 寄存器

7	6	5	4	3	2	1	0
TXL	TXA	RSVD			RXL	RXFL	RXN
R/W1C-0h	R/W1C-0h	R-0h			R/W1C-0h	R/W1C-0h	R/W1C-0h

表 7-91. SPI\_IR 寄存器字段说明

位	字段	类型	复位	说明
7	TXL	R/W1C	0h	因 FIFO 填满导致 TX 消息丢失的中断 0h = 无 TX FIFO 消息丢失 1h = 尝试写入时因 FIFO 填满，TX FIFO 丢失 1 个消息 ( 溢出 )
6	TXA	R/W1C	0h	TX 空间可用的中断  <div>备注</div> <div>当 SPI IP 被使能时，由于 FIFO 为空，此中断将置位</div> 0h = 无 TX FIFO 空间可用的中断 1h = TX FIFO 已达到触发级别阈值
5-3	RSVD	R	0h	
2	RXL	R/W1C	0h	RX 溢出/丢失消息中断 0h = RX FIFO 消息未丢失 ( 无溢出 ) 中断 1h = 由于 RX FIFO 填满 ( 溢出中断 )，至少丢失了 1 个消息
1	RXFL	R/W1C	0h	RX 填充级别中断。 级别由 SPI_CTRL.RX_LVL_INT 设定 0h = 无 RX FIFO 填充级别中断 1h = RX FIFO 已达到填充级别
0	RXN	R/W1C	0h	RX 新消息中断 0h = 无新消息接收中断 1h = 已接收到新消息

### 7.6.1.37 UART\_IR 寄存器 ( 偏移 = 34h ) [复位 = 00h]

图 7-52 显示了 UART\_IR 寄存器，表 7-92 中对此进行了介绍。

返回到[汇总表](#)。

这是 UART\_IR 寄存器的可写镜像，便于读取和清除中断

**图 7-52. UART\_IR 寄存器**

7	6	5	4	3	2	1	0
TXL	TXA	RXBR	RXFE	RXPE	RXL	RXFL	RXN
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

**表 7-92. UART\_IR 寄存器字段说明**

位	字段	类型	复位	说明
7	TXL	R/W1C	0h	因已满而 TX 丢失了字节 0h = TX FIFO 字节未丢失 1h = 尝试写入时因 FIFO 满，TX FIFO 丢失 1 个字节 ( 溢出 )
6	TXA	R/W1C	0h	TX 空间可用  <div style="text-align: center;"><b>备注</b></div> <div style="text-align: center;">当 UART IP 被使能时，由于 FIFO 为空，此中断将置位</div> 0h = 无 TX FIFO 空间可用的中断 1h = TX FIFO 已达到触发级别阈值
5	RXBR	R/W1C	0h	接收到 RX 中断 0h = 未接收到中断 1h = 接收到中断
4	RXFE	R/W1C	0h	RX 组帧错误 0h = 未发生接收组帧错误 1h = 发生了 RX 组帧错误
3	RXPE	R/W1C	0h	RX 奇偶校验错误 0h = 未发生接收到的奇偶校验错误 1h = 发生了 RX 奇偶校验错误
2	RXL	R/W1C	0h	RX 溢出/丢失字节 0h = 无 RX FIFO 字节丢失 ( 无溢出 ) 中断 1h = 因 RX FIFO 满，至少 1 个字节丢失 ( 溢出中断 )
1	RXFL	R/W1C	0h	RX 填充级别 0h = 无 RX FIFO 填充级别中断 1h = RX FIFO 达到填充级别
0	RXN	R/W1C	0h	RX 新字节 0h = 无新字节中断 1h = 已接收到新字节

**7.6.1.38 I2C\_IR 寄存器 ( 偏移 = 35h ) [复位 = 00h]**

图 7-53 显示了 I2C\_IR，表 7-93 中对此进行了介绍。

返回到[汇总表](#)。

这是 I2C\_IR 寄存器的可写镜像，便于读取和清除中断

**图 7-53. I2C\_IR 寄存器**

7	6	5	4	3	2	1	0
TXL	TXA	DNACK	ANACK	SBRC	RXL	RXFL	RXN
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

**表 7-93. I2C\_IR 寄存器字段说明**

位	字段	类型	复位	说明
7	TXL	R/W1C	0h	因 FIFO 满导致 TX 消息丢失的中断 0h = 无 TX FIFO 字节丢失 1h = 尝试写入时，因 FIFO 满，TX FIFO 丢失 1 个字节 ( 溢出 )
6	TXA	R/W1C	0h	TX 空间可用的中断。 触发级别由 I2C_FIFO_CTRL.TX_TRG 设置来设定 <div><b>备注</b> 当 I2C IP 被使能时，由于 FIFO 为空，此中断将置位</div> 0h = 无 TX FIFO 空间可用的中断 1h = TX FIFO 已达到触发级别阈值
5	DNACK	R/W1C	0h	数据 NACK 中断 0h = 未检测到 NACK 1h = 在数据阶段检测到 NACK
4	ANACK	R/W1C	0h	地址 NACK 中断 0h = 未检测到 NACK 1h = 在地址阶段检测到 NACK
3	SBRC	R/W1C	0h	总线卡滞恢复完成中断 0h = 未发生总线卡滞恢复 1h = 总线卡滞恢复已完成。检查 I2C_STATUS 寄存器，查看总线是否处于卡滞或静止状态。
2	RXL	R/W1C	0h	RX 溢出/消息丢失中断 0h = 无 RX FIFO 字节丢失 ( 无溢出 ) 中断 1h = 因 RX FIFO 满，至少 1 个字节丢失 ( 溢出中断 )
1	RXFL	R/W1C	0h	RX 填充级别中断 0h = 无 RX FIFO 填充级别中断 1h = RX FIFO 达到填充级别
0	RXN	R/W1C	0h	RX 新消息中断 0h = 无新字节接收中断 1h = 已接收到新字节

### 7.6.1.39 PWM0\_IR 寄存器 ( 偏移 = 36h ) [复位 = 00h]

图 7-54 显示了 PWM0\_IR，表 7-94 中对此进行了介绍。

返回到[汇总表](#)。

这是 PWM0\_IR 寄存器的可写镜像，便于读取和清除中断

**图 7-54. PWM0\_IR 寄存器**

7	6	5	4	3	2	1	0
RSVD					PULSE_OVF	IAS	RC
R-0h					R/W1C-0h	R/W1C-0h	R/W1C-0h

**表 7-94. PWM0\_IR 寄存器字段说明**

位	字段	类型	复位	说明
7-3	RSVD	R	0h	保留
2	PULSE_OVF	R/W1C	0h	当 CUR_PULSE 计数器溢出时设置的当前脉冲溢出 0h = 当前脉冲计数器未溢出 1h = 当前脉冲计数器已溢出
1	IAS	R/W1C	0h	输入自动停止中断 设置输入自动停止条件何时发生，如果 PWM 配置为立即停止，则输出也将停止（由 RC 位反映）。 如果 PWM 配置为在 IAS 触发器时开始停止斜坡，则在出现输入条件时立即设置此中断，并在输出关闭时设置 RC 位。 0h = 未触发 IAS 的输入 1h = 已触发 IAS 的输入
0	RC	R/W1C	0h	PWM 斜坡完成 设置斜坡何时完成。如果 AS_EN=1（自动停止），则当 PWM 通道关闭时会发生这种情况。 如果 AS_EN = 0，则会在 PWM 达到 END 值时设置该位。 如果 IAS_EN = 1（输入自动停止），则在 PWM 输出停止后会设置此位 0h = PWM 通道斜坡未完成 1h = PWM 通道斜坡完成

### 7.6.1.40 PWM1\_IR 寄存器 ( 偏移 = 37h ) [复位 = 00h]

图 7-55 显示了 PWM1\_IR，表 7-95 中对此进行了介绍。

返回到[汇总表](#)。

这是 PWM1\_IR 寄存器的可写镜像，便于读取和清除中断

图 7-55. PWM1\_IR 寄存器

7	6	5	4	3	2	1	0
RSVD					PULSE_OVF	IAS	RC
R-0h					R/W1C-0h	R/W1C-0h	R/W1C-0h

表 7-95. PWM1\_IR 寄存器字段说明

位	字段	类型	复位	说明
7-3	RSVD	R	0h	保留
2	PULSE_OVF	R/W1C	0h	当 CUR_PULSE 计数器溢出时设置的当前脉冲溢出 0h = 当前脉冲计数器未溢出 1h = 当前脉冲计数器已溢出
1	IAS	R/W1C	0h	输入自动停止中断 设置输入自动停止条件何时发生，如果 PWM 配置为立即停止，则输出也将停止（由 RC 位反映）。 如果 PWM 配置为在 IAS 触发器时开始停止斜坡，则在出现输入条件时立即设置此中断，并在输出关闭时设置 RC 位。 0h = 未触发 IAS 的输入 1h = 已触发 IAS 的输入
0	RC	R/W1C	0h	PWM 斜坡完成 设置斜坡何时完成。如果 AS_EN=1（自动停止），则当 PWM 通道关闭时会发生这种情况。 如果 AS_EN = 0，则会在 PWM 达到 END 值时设置该位。 如果 IAS_EN = 1（输入自动停止），则在 PWM 输出停止后会设置此位 0h = PWM 通道斜坡未完成 1h = PWM 通道斜坡完成

### 7.6.2 SPI 寄存器

节 7.6.2 列出了 SPI 寄存器的存储器映射寄存器。节 7.6.2 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 7-96. SPI 寄存器

偏移	首字母缩写词	寄存器名称	部分
1000h	SPI_CREL	SPI 内核版本	<a href="#">节 7.6.2.1</a>
1001h	SPI_SCRATCH	SPI 暂存区	<a href="#">节 7.6.2.2</a>
1002h	SPI_CTRL	SPI 控制	<a href="#">节 7.6.2.3</a>
1003h	SPI_CFG_0	SPI 通道配置 0	<a href="#">节 7.6.2.4</a>
1004h	SPI_CFG_1	SPI 通道配置 1	<a href="#">节 7.6.2.5</a>
1005h	SPI_DR_0	SPI 数据速率 0	<a href="#">节 7.6.2.6</a>
1006h	SPI_DR_1	SPI 数据速率 1	<a href="#">节 7.6.2.7</a>
1007h	SPI_DR_2	SPI 数据速率 2	<a href="#">节 7.6.2.8</a>
1008h	SPI_DR_3	SPI 数据速率 3	<a href="#">节 7.6.2.9</a>
1009h	SPI_CHAN_EN	SPI 通道使能	<a href="#">节 7.6.2.10</a>
100Ah	SPI_CS_POL	SPI 片选极性	<a href="#">节 7.6.2.11</a>
100Bh	SPI_FIFO_CTRL	SPI FIFO 控制	<a href="#">节 7.6.2.12</a>

**表 7-96. SPI 寄存器 (续)**

偏移	首字母缩写词	寄存器名称	部分
100Ch	SPI_IE_0	SPI 中断使能 0	<a href="#">节 7.6.2.13</a>
100Dh	SPI_IE_1	SPI 中断使能 1	<a href="#">节 7.6.2.14</a>
100Eh	SPI_IR	SPI 中断寄存器	<a href="#">节 7.6.2.15</a>
100Fh	SPI_FS	SPI FIFO 状态	<a href="#">节 7.6.2.16</a>
1010h	SPI_RX_FIFO	SPI 接收 FIFO	<a href="#">节 7.6.2.18</a>
1010h	SPI_TX_FIFO	SPI 发送 FIFO	<a href="#">节 7.6.2.17</a>
1011h	SPI_RXFS	SPI RX FIFO 状态	<a href="#">节 7.6.2.19</a>
1012h	SPI_TXFS	SPI TX FIFO 状态	<a href="#">节 7.6.2.20</a>
1013h	SPI_TXES	SPI TX 元素状态	<a href="#">节 7.6.2.21</a>

复杂的位访问类型经过编码可适应小型表单元。[节 7.6.2](#) 展示了适用于此部分中访问类型的代码。

**表 7-97. SPI 访问类型代码**

访问类型	代码	说明
读取类型		
R	R	读取
RH	R H	读取 由硬件置位或清零
写入类型		
W	W	写入
W1C	W 1C	写入 1 以清零
W1S	W 1S	写入 1 以进行设置
WP	W P	写入 需要特权访问
复位或默认值		
-n		复位后的值或默认值



7.6.2.1 SPI\_CREL 寄存器 ( 偏移 = 1000h ) [复位 = 87h]

图 7-56 显示了 SPI\_CREL , 表 7-98 对其进行了介绍。

返回到[汇总表](#)。

SPI IP 发布版本

图 7-56. SPI\_CREL 寄存器

7	6	5	4	3	2	1	0
CREL							
R-87h							

表 7-98. SPI\_CREL 寄存器字段说明

位	字段	类型	复位	说明
7-0	CREL	R	87h	SPI 内核发布版本

### 7.6.2.2 SPI\_SCRATCH 寄存器 ( 偏移 = 1001h ) [复位 = 00h]

图 7-57 显示了 SPI\_SCRATCH，表 7-99 对其进行了介绍。

返回到[汇总表](#)。

客户暂存寄存器

图 7-57. SPI\_SCRATCH 寄存器

7	6	5	4	3	2	1	0
SCRATCH							
R/W-0h							

表 7-99. SPI\_SCRATCH 寄存器字段说明

位	字段	类型	复位	说明
7-0	SCRATCH	R/W	0h	可供用户写入的寄存器，用于读写测试。 除读写测试外，该寄存器无其他功能。

### 7.6.2.3 SPI\_CTRL 寄存器 ( 偏移 = 1002h ) [复位 = 00h]

图 7-58 显示了 SPI\_CTRL 寄存器，表 7-100 中对此进行了介绍。

返回到[汇总表](#)。

该寄存器控制 SPI IP

图 7-58. SPI\_CTRL 寄存器

7	6	5	4	3	2	1	0
TX_TRG		RX_TRG		RESERVED		SPI_EN	SPI_CCE
R/W-0h		R/W-0h		R-0h		RH-0h	R/W-0h

表 7-100. SPI\_CTRL 寄存器字段说明

位	字段	类型	复位	说明
7-6	TX_TRG	R/W	0h	<p>设定当 TX FIFO 中的空闲字节数增加到指定数量时的触发级别。用于在 FIFO 中有一定数量的空闲空间时产生中断，以便处理器可以将多个字节加载到 FIFO 中，而不是一次仅加载 1 个字节。</p> <hr/> <p><b>备注</b></p> <p>该中断触发方式类似于边沿触发型中断。</p> <p>如果用户始终保持不低于 8 个空闲空间</p> <hr/> <p>0h = 16 字节空闲 1h = 32 字节空闲 2h = [ ( 最大大小 ) - 128] 字节空闲 3h = [ ( 最大大小 ) - 64] 字节空闲</p>
5-4	RX_TRG	R/W	0h	<p>设定当 RX FIFO 达到 FIFO 中所存储的指定字节数时，RX FIFO 的触发级别。这允许 CPU 批量移出字节，而不是一次 1 个字节，从而可以减少中断次数</p> <p>0h = RX FIFO 中有 4 字节 1h = RX FIFO 中有 8 字节 2h = RX FIFO 中有 [ ( 最大大小 ) - 128] 字节 3h = RX FIFO 中有 [ ( 最大大小 ) - 64] 字节</p>
3-2	RESERVED	R	0h	保留
1	SPI_EN	RH	0h	<p>SPI IP 使能状态标志。</p> <p>该标志不可写，但当通过从 MRAM 向 IP 分配内存以启用 SPI IP 时，可将此标志置位。</p> <p>这通过 <a href="#">MRAM_IP_CFG</a> 寄存器完成</p> <p>0h = SPI IP 被禁用，所有 SPI 功能均禁用 1h = SPI IP 被使能</p>
0	SPI_CCE	R/W	0h	<p>SPI IP 更改控制使能位。</p> <p>仅当 SPI_EN 为 0 时才能设置</p> <p>0h = SPI 配置寄存器受写保护 1h = 可以更改 SPI 配置寄存器</p>

#### 7.6.2.4 SPI\_CFG\_0 寄存器 ( 偏移 = 1003h ) [复位 = 00h]

图 7-59 显示了 SPI\_CFG\_0，表 7-101 中对此进行了介绍。

返回到[汇总表](#)。

SPI 通道配置。设置 SPI 模式和其他通道特定设置。

图 7-59. SPI\_CFG\_0 寄存器

7	6	5	4	3	2	1	0
RSVD	LSB_1	CLKPL_1	CLKPH_1	RSVD	LSB_0	CLKPL_0	CLKPH_0
R-0h	R/WP-0h	R/WP-0h	R/WP-0h	R-0h	R/WP-0h	R/WP-0h	R/WP-0h

表 7-101. SPI\_CFG\_0 寄存器字段说明

位	字段	类型	复位	说明
7	RSVD	R	0h	保留
6	LSB_1	R/WP	0h	LSB 优先选择。 控制接收和发送移位寄存器的方向 0h = MSB 在前 1h = LSB 在前
5	CLKPL_1	R/WP	0h	时钟极性选择 0h = SCK 的无效状态为低电平 1h = SCK 的无效状态为高电平
4	CLKPH_1	R/WP	0h	时钟相位选择 0h = 数据在第一个 CLK 边沿发生变化，在下一个边沿被捕获 1h = 数据在第一个 CLK 边沿被捕获，在下一个边沿发生变化
3	RSVD	R	0h	保留
2	LSB_0	R/WP	0h	LSB 优先选择。 控制接收和发送移位寄存器的方向 0h = MSB 在前 1h = LSB 在前
1	CLKPL_0	R/WP	0h	时钟极性选择 0h = SCK 的无效状态为低电平 1h = SCK 的无效状态为高电平
0	CLKPH_0	R/WP	0h	时钟相位选择 0h = 数据在第一个 CLK 边沿发生变化，在下一个边沿被捕获 1h = 数据在第一个 CLK 边沿被捕获，在下一个边沿发生变化

### 7.6.2.5 SPI\_CFG\_1 寄存器 ( 偏移 = 1004h ) [复位 = 00h]

图 7-60 显示了 SPI\_CFG\_1，表 7-102 中对此进行了介绍。

返回到[汇总表](#)。

SPI 通道配置。设置 SPI 模式和其他通道特定设置。

图 7-60. SPI\_CFG\_1 寄存器

7	6	5	4	3	2	1	0
RSVD	LSB_3	CLKPL_3	CLKPH_3	RSVD	LSB_2	CLKPL_2	CLKPH_2
R-0h	R/WP-0h	R/WP-0h	R/WP-0h	R-0h	R/WP-0h	R/WP-0h	R/WP-0h

表 7-102. SPI\_CFG\_1 寄存器字段说明

位	字段	类型	复位	说明
7	RSVD	R	0h	保留
6	LSB_3	R/WP	0h	LSB 优先选择。 控制接收和发送移位寄存器的方向 0h = MSB 在前 1h = LSB 在前
5	CLKPL_3	R/WP	0h	时钟极性选择 0h = SCK 的无效状态为低电平 1h = SCK 的无效状态为高电平
4	CLKPH_3	R/WP	0h	时钟相位选择 0h = 数据在第一个 CLK 边沿发生变化，在下一个边沿被捕获 1h = 数据在第一个 CLK 边沿被捕获，在下一个边沿发生变化
3	RSVD	R	0h	保留
2	LSB_2	R/WP	0h	LSB 优先选择。 控制接收和发送移位寄存器的方向 0h = MSB 在前 1h = LSB 在前
1	CLKPL_2	R/WP	0h	时钟极性选择 0h = SCK 的无效状态为低电平 1h = SCK 的无效状态为高电平
0	CLKPH_2	R/WP	0h	时钟极性选择 0h = SCK 的无效状态为低电平 1h = SCK 的无效状态为高电平

#### 7.6.2.6 SPI\_DR\_0 寄存器 ( 偏移 = 1005h ) [复位 = 00h]

图 7-61 显示了 SPI\_DR\_0，表 7-103 中对此进行了介绍。

返回到[汇总表](#)。

SPI 通道 0 的 SPI 数据速率寄存器。此寄存器中的值被解释为比输入值大 1。该值应解释为 SPI 时钟的半个周期。频率可通过公式  $f = 20\text{MHz} / (x+1)$  计算得出。其中 x 是 SPI\_DR 中的值。通过  $x = (20\text{MHz} / f) - 1$  可找到最接近的寄存器值

图 7-61. SPI\_DR\_0 寄存器

7	6	5	4	3	2	1	0
预分频器							
R/WP-0h							

表 7-103. SPI\_DR\_0 寄存器字段说明

位	字段	类型	复位	说明
7-0	预分频器	R/WP	0h	SPI 时钟预分频器的低位字节。 预分频器值被解释为比输入值大 1。 该值默认为 EEPROM 中编程的值。

#### 7.6.2.7 SPI\_DR\_1 寄存器 ( 偏移 = 1006h ) [复位 = 00h]

图 7-62 显示了 SPI\_DR\_1，表 7-104 中对此进行了介绍。

返回到[汇总表](#)。

SPI 通道 1 的 SPI 数据速率寄存器。此寄存器中的值被解释为比输入值大 1。该值应解释为 SPI 时钟的半个周期。频率可通过公式  $f = 20\text{MHz} / (x+1)$  计算得出。其中 x 是 SPI\_DR 中的值。通过  $x = (20\text{MHz} / f) - 1$  可找到最接近的寄存器值

图 7-62. SPI\_DR\_1 寄存器

7	6	5	4	3	2	1	0
预分频器							
R/WP-0h							

表 7-104. SPI\_DR\_1 寄存器字段说明

位	字段	类型	复位	说明
7-0	预分频器	R/WP	0h	SPI 时钟预分频器的低位字节。 预分频器值被解释为比输入值大 1。 该值默认为 EEPROM 中编程的值。

### 7.6.2.8 SPI\_DR\_2 寄存器 ( 偏移 = 1007h ) [复位 = 00h]

图 7-63 显示了 SPI\_DR\_2，表 7-105 中对此进行了介绍。

返回到[汇总表](#)。

SPI 通道 2 的 SPI 数据速率寄存器。此寄存器中的值被解释为比输入值大 1。该值应解释为 SPI 时钟的半个周期。频率可通过公式  $f = 20\text{MHz} / (x+1)$  计算得出。其中 x 是 SPI\_DR 中的值。通过  $x = (20\text{MHz} / f) - 1$  可找到最接近的寄存器值

图 7-63. SPI\_DR\_2 寄存器

7	6	5	4	3	2	1	0
预分频器							
R/WP-0h							

表 7-105. SPI\_DR\_2 寄存器字段说明

位	字段	类型	复位	说明
7-0	预分频器	R/WP	0h	SPI 时钟预分频器的低位字节。 预分频器值被解释为比输入值大 1。 该值默认为 EEPROM 中编程的值。



### 7.6.2.9 SPI\_DR\_3 寄存器 ( 偏移 = 1008h ) [复位 = 00h]

图 7-64 显示了 SPI\_DR\_3，表 7-106 中对此进行了介绍。

返回到[汇总表](#)。

SPI 通道 3 的 SPI 数据速率寄存器。此寄存器中的值被解释为比输入值大 1。该值应解释为 SPI 时钟的半个周期。频率可通过公式  $f = 20\text{MHz} / (x+1)$  计算得出。其中 x 是 SPI\_DR 中的值。通过  $x = (20\text{MHz} / f) - 1$  可找到最接近的寄存器值

图 7-64. SPI\_DR\_3 寄存器

7	6	5	4	3	2	1	0
预分频器							
R/WP-0h							

表 7-106. SPI\_DR\_3 寄存器字段说明

位	字段	类型	复位	说明
7-0	预分频器	R/WP	0h	SPI 时钟预分频器的低位字节。 预分频器值被解释为比输入值大 1。 该值默认为 EEPROM 中编程的值。 <div>备注 SPI 配置与通道 4-7 共享</div>

#### 7.6.2.10 SPI\_CHAN\_EN 寄存器 ( 偏移 = 1009h ) [复位 = 00h]

图 7-65 显示了 SPI\_CHAN\_EN 寄存器，表 7-107 中对此进行了介绍。

返回到[汇总表](#)。

SPI 通道使能寄存器。一个 SPI 通道对应一个独立的片选通道。

**备注**  
片选通道的 GPIO 需要在 IO\_CFG\_0 和 IO\_CFG\_1 寄存器中设置为特殊功能

图 7-65. SPI\_CHAN\_EN 寄存器

7	6	5	4	3	2	1	0
C7_EN	C6_EN	C5_EN	C4_EN	C3_EN	C2_EN	C1_EN	C0_EN
R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h

表 7-107. SPI\_CHAN\_EN 寄存器字段说明

位	字段	类型	复位	说明
7	C7_EN	R/WP	0h	CS 通道 7 使能 0h = SPI 通道禁用/未使用 1h = SPI 通道被使能
6	C6_EN	R/WP	0h	CS 通道 6 使能 0h = SPI 通道禁用/未使用 1h = SPI 通道被使能
5	C5_EN	R/WP	0h	CS 通道 5 使能 0h = SPI 通道禁用/未使用 1h = SPI 通道被使能
4	C4_EN	R/WP	0h	CS 通道 4 使能 0h = SPI 通道禁用/未使用 1h = SPI 通道被使能
3	C3_EN	R/WP	0h	CS 通道 3 使能 0h = SPI 通道禁用/未使用 1h = SPI 通道被使能
2	C2_EN	R/WP	0h	CS 通道 2 使能 0h = SPI 通道禁用/未使用 1h = SPI 通道被使能
1	C1_EN	R/WP	0h	CS 通道 1 使能 0h = SPI 通道禁用/未使用 1h = SPI 通道被使能
0	C0_EN	R/WP	0h	CS 通道 0 使能 0h = SPI 通道禁用/未使用 1h = SPI 通道被使能

### 7.6.2.11 SPI\_CS\_POL 寄存器 ( 偏移 = 100Ah ) [复位 = 00h]

图 7-66 显示了 SPI\_CS\_POL 寄存器，表 7-108 中对此进行了介绍。

返回到[汇总表](#)。

图 7-66. SPI\_CS\_POL 寄存器

7	6	5	4	3	2	1	0
RSVD				CS3_AH	CS2_AH	CS1_AH	CS0_AH
R-0h				R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h

表 7-108. SPI\_CS\_POL 寄存器字段说明

位	字段	类型	复位	说明
7-4	RSVD	R	0h	
3	CS3_AH	R/WP	0h	配置通道 CS 引脚的有效状态  <div>备注</div> <div>CS3 片选配置与 CS4-7 共享</div> 0h = 片选引脚功能为低电平有效 ( nCS 或 $\overline{\text{CS}}$ ) 1h = 片选引脚功能为高电平有效 (CS)
2	CS2_AH	R/WP	0h	配置通道 CS 引脚的有效状态 0h = 片选引脚功能为低电平有效 ( nCS 或 $\overline{\text{CS}}$ ) 1h = 片选引脚功能为高电平有效 (CS)
1	CS1_AH	R/WP	0h	配置通道 CS 引脚的有效状态 0h = 片选引脚功能为低电平有效 ( nCS 或 $\overline{\text{CS}}$ ) 1h = 片选引脚功能为高电平有效 (CS)
0	CS0_AH	R/WP	0h	配置通道 CS 引脚的有效状态 0h = 片选引脚功能为低电平有效 ( nCS 或 $\overline{\text{CS}}$ ) 1h = 片选引脚功能为高电平有效 (CS)

**7.6.2.12 SPI\_FIFO\_CTRL 寄存器 ( 偏移 = 100Bh ) [复位 = 0h]**

表 7-109 显示了 SPI\_FIFO\_CTRL。

返回到[汇总表](#)。

**表 7-109. SPI\_FIFO\_CTRL 寄存器字段说明**

位	字段	类型	复位	说明
2	CLR_TX	RH/W1S	0h	清除发送 FIFO 的内容。 清除完成后，硬件将该位复位为 0。 0h = 不执行任何操作 1h = 清除发送 FIFO
1	CLR_RX	RH/W1S	0h	清除接收 FIFO 的内容。 清除完成后，硬件将该位复位为 0。 0h = 不执行任何操作 1h = 清除接收 FIFO
0	RSVD	R	0h	

### 7.6.2.13 SPI\_IE\_0 寄存器 ( 偏移 = 100Ch ) [复位 = 00h]

图 7-67 显示了 SPI\_IE\_0，表 7-110 中对此进行了介绍。

返回到[汇总表](#)。

用于为 INT0 引脚启用特定中断的中断使能位。已启用的中断将在 INT0 引脚上发出信号。请注意，若要在 INT0 引脚上观察到中断输出信号，必须先在 INT\_CFG 寄存器中启用 INT0 功能。

图 7-67. SPI\_IE\_0 寄存器

7	6	5	4	3	2	1	0
TXLIE0	TXAIE0	RSVD			RXLIE0	RXFLIE0	RXNIE0
R/W-0h	R/W-0h	R-0h			R/W-0h	R/W-0h	R/W-0h

表 7-110. SPI\_IE\_0 寄存器字段说明

位	字段	类型	复位	说明
7	TXLIE0	R/W	0h	因 FIFO 满导致 TX 消息丢失的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
6	TXAIE0	R/W	0h	TX 空间可用的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
5-3	RSVD	R	0h	
2	RXLIE0	R/W	0h	RX 溢出/丢失消息的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
1	RXFLIE0	R/W	0h	RX 填充级别的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
0	RXNIE0	R/W	0h	RX 新消息的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。

#### 7.6.2.14 SPI\_IE\_1 寄存器 ( 偏移 = 100Dh ) [复位 = 00h]

图 7-68 显示了 SPI\_IE\_1，表 7-111 中对此进行了介绍。

返回到[汇总表](#)。

用于为 INT0 引脚启用特定中断的中断使能位。已启用的中断将在 INT1 引脚上发出信号。请注意，若要在 INT0 引脚上观察到中断输出信号，必须先在 INT\_CFG 寄存器中启用 INT0 功能。

**图 7-68. SPI\_IE\_1 寄存器**

7	6	5	4	3	2	1	0
TXLIE1	TXAIE1	RSVD			RXLIE1	RXFLIE1	RXNIE1
R/W-0h	R/W-0h	R-0h			R/W-0h	R/W-0h	R/W-0h

**表 7-111. SPI\_IE\_1 寄存器字段说明**

位	字段	类型	复位	说明
7	TXLIE1	R/W	0h	因 FIFO 满导致 TX 消息丢失的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
6	TXAIE1	R/W	0h	TX 空间可用的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
5-3	RSVD	R	0h	
2	RXLIE1	R/W	0h	RX 溢出/丢失消息的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
1	RXFLIE1	R/W	0h	RX 填充级别的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
0	RXNIE1	R/W	0h	RX 新消息的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。

### 7.6.2.15 SPI\_IR 寄存器 ( 偏移 = 100Eh ) [复位 = 00h]

图 7-69 显示了 SPI\_IR 寄存器，表 7-112 中对此进行了介绍。

返回到[汇总表](#)。

当前中断寄存器，包含所有当前已置位的中断对应的位。写入 1 以进行清除。

图 7-69. SPI\_IR 寄存器

7	6	5	4	3	2	1	0
TXL	TXA	RSVD			RXL	RXFL	RXN
R/W1C-0h	R/W1C-0h	R-0h			R/W1C-0h	R/W1C-0h	R/W1C-0h

表 7-112. SPI\_IR 寄存器字段说明

位	字段	类型	复位	说明
7	TXL	R/W1C	0h	因 FIFO 填满导致 TX 消息丢失的中断 0h = 无 TX FIFO 消息丢失 1h = 尝试写入时，由于 FIFO 填满，TX FIFO 丢失了一个消息 ( 溢出 )
6	TXA	R/W1C	0h	TX 空间可用的中断 <div>备注</div> <div>当 SPI IP 被使能时，由于 FIFO 为空，此中断将置位</div> 0h = 无 TX FIFO 空间可用中断 1h = TX FIFO 已达到触发级别阈值
5-3	RSVD	R	0h	
2	RXL	R/W1C	0h	RX 溢出/丢失消息中断 0h = 无 RX FIFO 消息丢失 ( 无溢出 ) 中断 1h = 因 RX FIFO 填满，至少丢失了 1 个消息 ( 溢出中断 )
1	RXFL	R/W1C	0h	RX 填充级别中断。 级别由 SPI_CTRL.RX_LVL_INT 设定 0h = 无 RX FIFO 填充级别中断 1h = RX FIFO 达到填充级别
0	RXN	R/W1C	0h	RX 新消息中断 0h = 无新消息接收中断 1h = 已接收到新消息

#### 7.6.2.16 SPI\_FS 寄存器 ( 偏移 = 100Fh ) [复位 = 80h]

图 7-70 显示了 SPI\_FS，表 7-113 对其进行了介绍。

返回到[汇总表](#)。

SPI FIFO 状态

图 7-70. SPI\_FS 寄存器

7	6	5	4	3	2	1	0
TXFSRE	RXFD	RXFB					
RH-1h	RH-0h	RH-0h					

表 7-113. SPI\_FS 寄存器字段说明

位	字段	类型	复位	说明
7	TXFSRE	RH	1h	<p>TX FIFO 与移位寄存器是否为空。</p> <p>这意味着已传输所有排队的数据，且控制器处于空闲状态。</p> <p>FIFO 中没有任何内容，移位寄存器中也没有任何内容。</p> <p>如果设置了此位，则可以安全地配置或禁用 SPI 控制器</p> <div><p><b>备注</b></p><p>这不会考虑未完全写入的 TX FIFO 元素。</p><p>如果有一个已部分写入的 TX FIFO 元素，则该未传输的元素在元素已完全写入之前不会清除 IDLE 标志</p></div> <p>0h = 至少有 1 个正在进行的传输和/或待处理的传输。</p> <p>1h = TX FIFO 和移位寄存器为空，IP 为空闲。可安全地配置或关断控制器</p>
6	RXFD	RH	0h	<p>RX FIFO 是否包含 1 个或多个未读元素 ( SPI 传输 )</p> <p>0h = RX FIFO 为空</p> <p>1h = RX FIFO 中至少有 1 个元素</p>
5-0	RXFB	RH	0h	<p>存储在下一个 RX FIFO 元素中的 SPI 帧大小。</p> <p>有效值为 0 至 63。</p> <p>任何大于 63 的值都将反映为 63，剩余的值将在处理器读取数据时更新。</p>



### 7.6.2.17 SPI\_TX\_FIFO 寄存器 ( 偏移 =1010h ) [复位 = 0000h]

图 7-71 显示了 SPI\_TX\_FIFO，表 7-114 对其进行了介绍。

返回到[汇总表](#)。

SPI 发送 FIFO。在对该地址执行写入操作时，会将数据写入 SPI 发送 FIFO。有关更多信息，请参阅 [SPI 发送 FIFO](#)；这仅显示 SPI 帧的标头

图 7-71. SPI\_TX\_FIFO 寄存器

15	14	13	12	11	10	9	8
STORE_RX	RSVD				CHAN		
W-0h	R-0h				W-0h		
7	6	5	4	3	2	1	0
NUM_BYTES							
W-0h							

表 7-114. SPI\_TX\_FIFO 寄存器字段说明

位	字段	类型	复位	说明
15	STORE_RX	W	0h	将传输数据存储于 RX FIFO 中。 如果不需要，可以用来忽略传入的数据 <a href="#">SPI 发送 FIFO</a> 提供了更多信息。
14-11	RSVD	R	0h	保留
10-8	CHAN	W	0h	用于发送的 SPI 通道 SPI 通道控制芯片选择引脚 <a href="#">SPI 发送 FIFO</a> 提供了更多信息。
7-0	NUM_BYTES	W	0h	SPI 发送中的字节数 <a href="#">SPI 发送 FIFO</a> 提供了更多信息。

### 7.6.2.18 SPI\_RX\_FIFO 寄存器 ( 偏移 = 1010h ) [复位 = 0000h]

图 7-72 显示了 SPI\_RX\_FIFO，表 7-115 对其进行了介绍。

返回到[汇总表](#)。

在执行对该地址的读取操作时，将从 SPI 接收 FIFO 中读取数据。更多信息，请参阅 [SPI 接收 FIFO](#)；这仅显示 SPI 帧的标头

**图 7-72. SPI\_RX\_FIFO 寄存器**

15	14	13	12	11	10	9	8
CONT	RSVD				CHAN		
R-0h	R-0h				R-0h		
7	6	5	4	3	2	1	0
NUM_BYTES_REMAINING							
R-0h							

**表 7-115. SPI\_RX\_FIFO 寄存器字段说明**

位	字段	类型	复位	说明
15	CONT	R	0h	指示这是否为部分 RX FIFO 元素读取的标志。 如果为 0，则这是对该 FIFO 元素的第一次读取 <a href="#">SPI 接收 FIFO</a> 提供了更多信息。
14-11	RSVD	R	0h	保留
10-8	CHAN	R	0h	传输所用的 SPI 通道 <a href="#">SPI 接收 FIFO</a> 提供了更多信息。
7-0	NUM_BYTES_REMAINING	R	0h	FIFO 元素中要读取的剩余字节数 ( 包括此事务 ) <a href="#">SPI 接收 FIFO</a> 提供了更多信息。

### 7.6.2.19 SPI\_RXFS 寄存器 ( 偏移 = 1011h ) [复位 = 00h]

图 7-73 显示了 SPI\_RXFS , 表 7-116 对其进行了介绍。

返回到[汇总表](#)。

包含 RX FIFO 中存储的未读消息数量计数器。

图 7-73. SPI\_RXFS 寄存器

7	6	5	4	3	2	1	0
RX_FL							
R-0h							

表 7-116. SPI\_RXFS 寄存器字段说明

位	字段	类型	复位	说明
7-0	RX_FL	R	0h	RX FIFO 填充级别表示当前 RX FIFO 中存储的消息数量。 取值范围为 1 至 255。 列出的最大值取决于器件存储器的配置方式。 如果报告 255 , 则表示 FIFO 中有 255 个或更多帧。

#### 7.6.2.20 SPI\_TXFS 寄存器 ( 偏移 = 1012h ) [复位 = 00h]

图 7-74 显示了 SPI\_TXFS，表 7-117 对其进行了介绍。

返回到[汇总表](#)。

包含 TX FIFO 中可用空间数量的计数器。

图 7-74. SPI\_TXFS 寄存器

7	6	5	4	3	2	1	0
TX_SA							
R-0h							

表 7-117. SPI\_TXFS 寄存器字段说明

位	字段	类型	复位	说明
7-0	TX_SA	R	0h	TX FIFO 填充级别表示当前 TX FIFO 中的空闲字节数。 <div>备注 若可用字节数超过 255，则显示可用空间为 255 字节</div>

### 7.6.2.21 SPI\_TXES 寄存器 ( 偏移 = 1013h ) [复位 = 00h]

图 7-75 显示了 SPI\_TXES，表 7-118 对其进行了介绍。

返回到[汇总表](#)。

包含有关当前/下一个发送 FIFO 元素的信息

图 7-75. SPI\_TXES 寄存器

7	6	5	4	3	2	1	0
TXEIP	TXEBP						
R-0h	R-0h						

表 7-118. SPI\_TXES 寄存器字段说明

位	字段	类型	复位	说明
7	TXEIP	R	0h	进行中的 TX FIFO 元素写入 用于指示 TX FIFO 元素是否已部分写入。 如果没有足够的数据字节写入 FIFO ( 根据标头 )，则设置该标志。 如果该位被清除，则对 TX FIFO 的下次写入将是新元素，应从 SPI 标头字段开始 0h = TX 元素未在进行中 ( 对 TX FIFO 的下次写入是新元素 ) 1h = TX 元素正在进行中，具有预期剩余字节数
6-0	TXEBP	R	0h	TX 元素字节待处理 如果 TX 元素写入已开始，这是完成 FIFO 元素还需要的字节数。 如果剩余的预期字节数超过 127 个字节，则仍会显示剩余 127 个字节 ( 由于可以显示的最大数量 )  <div>备注</div> <p>这包括标头字节，如果传输中的字节数尚未写入，则可以更改值。 例如，如果仅写入了 SPI 通道标头字节，则该字段将读取 1，因为我们不知道在告知要传输的数据字节数量之前有多少字节处于挂起状态。 传输中的字节数已知后，会在写入数据时对此进行重新计算</p>

### 7.6.3 SPI 数据 FIFO

先进先出 (FIFO) 数据结构用于向器件内外传输数据。主要有两个 FIFO：接收 (RX) FIFO 和发送 (TX) FIFO。访问两者的地址相同，但执行写操作时，数据被复制到 TX FIFO；执行读操作时，从 RX FIFO 中读取数据。

#### 7.6.3.1 SPI 发送 FIFO (地址 = h1010)

发送 FIFO 为只写模式，当向其地址发送写入命令时，数据会被写入其中。当整个帧写入缓冲区后，传输将开始。在帧结束后写入的所有数据字节都将被忽略，直到开始新的发送 FIFO 写入。对发送 FIFO 的所有写入都应仅指向主地址，因为每个写入的字节都会自动移入 FIFO。例如，当写入的 SPI 大于单个 CAN 消息所能传输的大小时，会通过同一地址的多次 CAN 写入来完成，因为 SPI 报头字节会告知设备预期的数据字节数。

TX 缓冲区中每个帧的大小由消息的数据长度决定。每个 SPI FIFO 元素的开头都需要一个 2 字节的报头。缓冲区中 TX 帧的总长度为以下各项之和

- SPI 报头的 2 字节
- 数据字节的消息长度

有关更多信息和示例，请参阅 [SPI 控制协议](#)。

#### 备注

如果发送 FIFO 已满，写入的任何新消息都将被丢弃，且对应的中断会被置位。

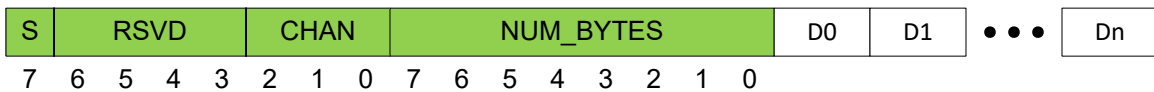


图 7-76. SPI TX FIFO 写入报头 (仅限报头和数据)

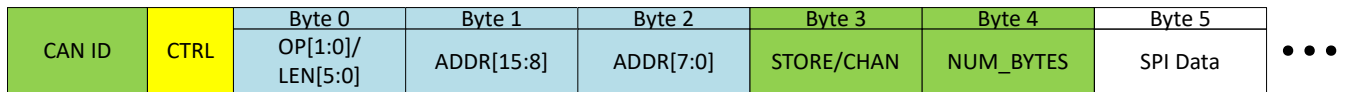


图 7-77. SPI TX FIFO 报头示例 (标准 CAN 帧格式)

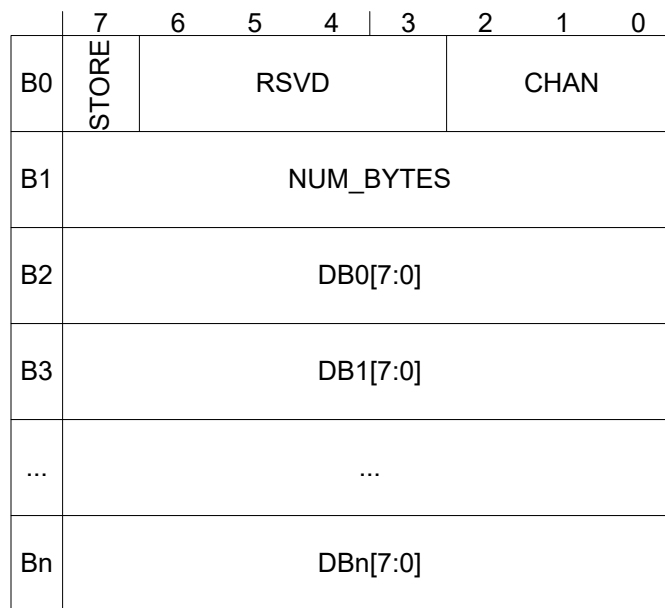


图 7-78. SPI 发送 FIFO

表 7-119. SPI 发送 FIFO 说明

字节	位	字段	类型	复位	说明
0	7	STORE_RX	W	0	是否将此消息的 POCI 数据存储在接收 FIFO 中 0 = 不将任何数据存储在接收 FIFO 中 1 = 将接收到的 POCI 数据保存在该消息的接收 FIFO 中
	6:3	RSVD	R	0	保留
	2:0	CHAN	W	0x0	用于该消息的 SPI 通道 有效值 0-3 0 = SPI 通道 0 1 = SPI 通道 1 2 = SPI 通道 2 3 = SPI 通道 3 4 = SPI 通道 4 5 = SPI 通道 5 6 = SPI 通道 6 7 = SPI 通道 7 <div><b>备注</b> 一个 SPI 通道对应一个特定的 GPIO，且要求该 GPIO 被分配为特殊功能</div>
1	7:0	NUM_BYTES	W	0x0	SPI 帧中的字节数（不包括上述报头字节） 有效值 1-255。0 为无效值。

### 7.6.3.2 SPI 接收 FIFO (地址 = h1010)

接收 FIFO 为只读。

RX 缓冲区中每个帧的大小由消息的数据长度决定。缓冲区中 RX 帧的总长度为以下各项之和

- SPI 报头的 2 字节
- 数据字节的消息长度

当从 FIFO 元素中读取了所有数据字节后，对 RX FIFO 的读取被视为完成且该 FIFO 元素被释放。当读取数据时，若读取的字节数超过剩余可读取字节数，超出部分的无效字节将返回 0x00。

从 RX FIFO 中对一帧数据进行部分读取后，下一次对 SPI FIFO 的读取会自动延续该帧的剩余部分，此时 CONT 位会被置位以标识这种延续状态，同时 NUM\_BYTES\_REMAINING 会显示该消息中尚待读取的数据字节数。

从空的 RX 缓冲区读取时，通道 0 的 NUM\_BYTES\_REMAINING 返回 0。帧的读取顺序始终与接收顺序一致。

通过向 SPI 清除 RX FIFO 位 (SPI\_FIFO\_CTRL[1]) 写入 1，可以清空 RX 缓冲区 (丢弃所有帧)。

#### 备注

如果接收 FIFO 已满，所有新传入的消息都会被丢弃，并设置一个中断以提醒处理器有消息丢失。

有关更多信息和示例，请参阅 [SPI 控制协议](#)。

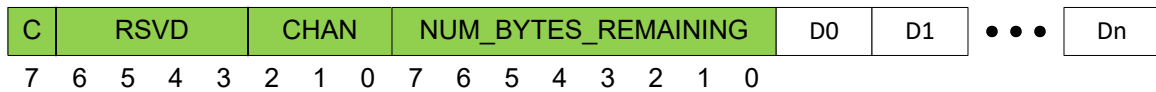


图 7-79. SPI RX FIFO 读取报头 (仅限报头和数据)



图 7-80. SPI RX FIFO 报头示例 (标准 CAN 帧格式)

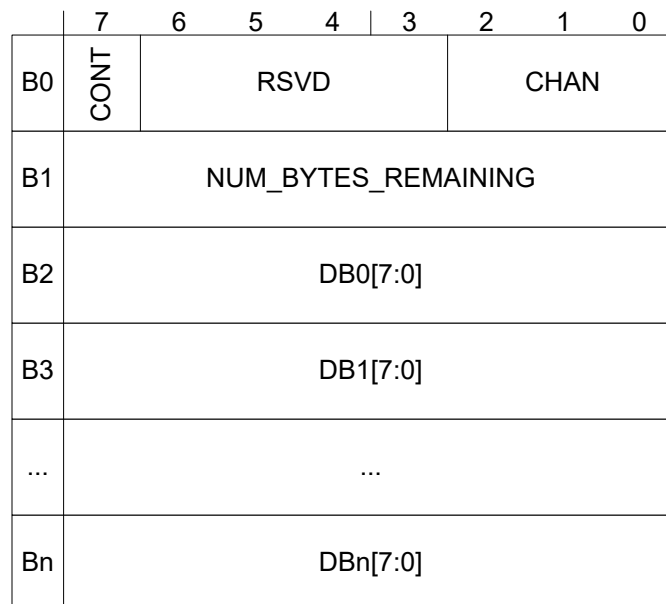


图 7-81. SPI 接收 FIFO



表 7-120. SPI 接收 FIFO 说明 ( SPI 报头开始 )

字节	位	字段	类型	复位	说明
0	7	CONT	R	0	用于指示是消息的第一部分还是延续部分的标志 ( 用于读取拆分为多个 CAN 读取的消息 ) 0 = 新消息的第一部分 1 = 前一次消息读取的延续
	6:3	RSVD	R	0	保留
	2:0	CHAN	R	0x0	用于该消息的 SPI 通道 有效值 0-3 0 = SPI 通道 0 1 = SPI 通道 1 2 = SPI 通道 2 3 = SPI 通道 3 4 = SPI 通道 4 5 = SPI 通道 5 6 = SPI 通道 6 7 = SPI 通道 7  <div>备注 一个 SPI 通道对应一个特定的 GPIO，且要求该 GPIO 被分配为特殊功能</div>
1	7:0	NUM_BYTES_REMAINING	R	0x0	此消息中剩余待读取的数据字节数 ( 包括当前帧中传输的数据字节 )。这不包括报头字节。这使处理器知道当前 CAN 帧中有多少字节是有效的，以及是否需要再次读取以从 FIFO 中获取该消息的更多数据。

#### 7.6.4 UART 寄存器

节 7.6.4 列出了 UART 寄存器的存储器映射寄存器。节 7.6.4 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 7-121. UART 寄存器

偏移	首字母缩写词	寄存器名称	部分
2000h	UART_CREL	UART 内核版本	节 7.6.4.1
2001h	UART_SCRATCH	UART 暂存区	节 7.6.4.2
2002h	UART_CTRL	UART 控制	节 7.6.4.3
2003h	UART_BR_LSB	UART 波特率 (LSB)	节 7.6.4.4
2004h	UART_BR_MSB	UART 波特率 (MSB)	节 7.6.4.5
2005h	UART_BR_FRAC	UART 波特率 ( 小数部分 )	节 7.6.4.6
2006h	UART_FIFO_CTRL	UART FIFO 控制	节 7.6.4.7
2007h	UART_IE_0	UART 中断使能 0	节 7.6.4.8
2008h	UART_IE_1	UART 中断使能 1	节 7.6.4.9
2009h	UART_IR	UART 中断寄存器	节 7.6.4.10
200Ah	UART_STATUS	UART 状态	节 7.6.4.11
200Bh	UART_RXFS	UART RX FIFO 状态	节 7.6.4.12
200Ch	UART_TXFS	UART TX FIFO 状态	节 7.6.4.13
2010h	UART_RX_FIFO	UART RX FIFO	节 7.6.4.14
2010h	UART_TX_FIFO	UART TX FIFO	节 7.6.4.15
2011h	UART_RX_ERR_STATUS	UART RX FIFO 错误状态	节 7.6.4.16

复杂的位访问类型经过编码可适应小型表单元。节 7.6.4 展示了适用于此部分中访问类型的代码。

**表 7-122. UART 访问类型代码**

访问类型	代码	说明
读取类型		
R	R	读取
RH	R H	读取 由硬件置位或清零
写入类型		
W	W	写入
W1C	W 1C	写入 1 以清零
W1S	W 1S	写入 1 以进行设置
WP	W P	写入 需要特权访问
复位或默认值		
-n		复位后的值或默认值

#### 7.6.4.1 UART\_CREL 寄存器 ( 偏移 = 2000h ) [复位 = 10h]

图 7-82 显示了 UART\_CREL , 表 7-123 对其进行了介绍。

返回到[汇总表](#)。

UART IP 发布版本

图 7-82. UART\_CREL 寄存器

7	6	5	4	3	2	1	0
MAJOR				MINOR			
R-1h				R-0h			

表 7-123. UART\_CREL 寄存器字段说明

位	字段	类型	复位	说明
7-4	MAJOR	R	1h	此修订版本的主版本号
3-0	MINOR	R	0h	此修订版本的次版本号

7.6.4.2 UART\_SCRATCH 寄存器 ( 偏移 = 2001h ) [复位 = 00h]

图 7-83 显示了 UART\_SCRATCH , 表 7-124 对其进行了介绍。

返回到[汇总表](#)。

客户暂存寄存器

图 7-83. UART\_SCRATCH 寄存器

7	6	5	4	3	2	1	0
SCRATCHPAD							
R/W-0h							

表 7-124. UART\_SCRATCH 寄存器字段说明

位	字段	类型	复位	说明
7-0	SCRATCHPAD	R/W	0h	客户暂存寄存器。 这可用于通过写入和读取值来验证通信，也可用于存储一些数据。 该寄存器不会在复位/上电复位后保留数据。

#### 7.6.4.3 UART\_CTRL 寄存器 ( 偏移 = 2002h ) [复位 = 04h]

图 7-84 显示了 UART\_CTRL 寄存器，表 7-125 中对此进行了介绍。

返回到[汇总表](#)。

该寄存器控制数据通信格式。包括字长、停止位数量、奇偶校验类型以及使能对 UART 配置的更改。

图 7-84. UART\_CTRL 寄存器

7	6	5	4	3	2	1	0
BRKGEN	FPAREN	PAR	PEN	2BSTOP	8BIT	UART_EN	CCE
R/W-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-0h	R/WP-1h	RH-0h	R/W-0h

表 7-125. UART\_CTRL 寄存器字段说明

位	字段	类型	复位	说明
7	BRKGEN	R/W	0h	中断控制位 0h = 正常 UART 操作 1h = 强制 TXD 为低电平以生成中断
6	FPAREN	R/WP	0h	强制奇偶校验位 ( 如果使能 ) 为特定值 0h = 按配置生成奇偶校验 1h = 对于发送和接收的数据，强制奇偶校验位按照奇偶校验位 (PAR) 配置中定义的值
5	PAR	R/WP	0h	奇校验或偶校验配置 0h = 奇校验 ( 1 位值 ) 1h = 偶校验 ( 0 位值 )
4	PEN	R/WP	0h	使能是否对每个字符使用奇偶校验 0h = 无奇偶校验位 1h = 奇偶校验位已使能
3	2BSTOP	R/WP	0h	定义每个字符使用的停止位数 0h = 1 个停止位 1h = 2 个停止位
2	8BIT	R/WP	1h	定义字符的大小 0h = 7 位数据 1h = 8 位数据
1	UART_EN	RH	0h	UART IP 使能状态标志。 该标志不可写，但当通过从 MRAM 向 IP 分配内存以启用 UART IP 时，可将此标志置位。 这通过 <a href="#">MRAM_IP_CFG</a> 寄存器完成 0h = UART IP 被禁用，所有 UART 功能均禁用 1h = UART IP 被使能
0	CCE	R/W	0h	UART IP 更改控制使能位。 仅当 UART_EN 为 0 时才可置位 0h = UART 配置寄存器受写保护 1h = 可以更改 UART 配置寄存器

7.6.4.4 UART\_BR\_LSB 寄存器 ( 偏移 = 2003h ) [复位 = 00h]

图 7-85 显示了 UART\_BR\_LSB，表 7-126 对其进行了介绍。

返回到[汇总表](#)。

用于波特时钟的 16 位分频值的低字节

图 7-85. UART\_BR\_LSB 寄存器

7	6	5	4	3	2	1	0
DLL							
R/WP-0h							

表 7-126. UART\_BR\_LSB 寄存器字段说明

位	字段	类型	复位	说明
7-0	DLL	R/WP	0h	用于波特率生成的 16 位分频值的最低有效字节。

#### 7.6.4.5 UART\_BR\_MSB 寄存器 ( 偏移 = 2004h ) [复位 = 00h]

图 7-86 显示了 UART\_BR\_MSB , 表 7-127 对其进行了介绍。

返回到[汇总表](#)。

用于波特时钟的 16 位分频值的高字节

图 7-86. UART\_BR\_MSB 寄存器

7	6	5	4	3	2	1	0
DLH							
R/WP-0h							

表 7-127. UART\_BR\_MSB 寄存器字段说明

位	字段	类型	复位	说明
7-0	DLH	R/WP	0h	用于波特率生成的 16 位分频值的最高有效字节。

#### 7.6.4.6 UART\_BR\_FRAC 寄存器 ( 偏移 = 2005h ) [复位 = 00h]

图 7-87 显示了 UART\_BR\_FRAC , 表 7-128 对其进行了介绍。

返回到[汇总表](#)。

波特时钟使用的 6 位小数分频值 ( 即 x/64 值 )

图 7-87. UART\_BR\_FRAC 寄存器

7	6	5	4	3	2	1	0
BD	RSVD	DLF					
R/WP-0h	R-0h	R/WP-0h					

表 7-128. UART\_BR\_FRAC 寄存器字段说明

位	字段	类型	复位	说明
7	BD	R/WP	0h	波特分频位, 用于控制波特率生成过程中所采用的过采样倍数。 0h = 启用 16 分频波特分频值 1h = 启用 8 分频波特分频值
6	RSVD	R	0h	
5-0	DLF	R/WP	0h	用于波特率生成的小数分频值 (x/64)。



#### 7.6.4.7 UART\_FIFO\_CTRL 寄存器 ( 偏移 = 2006h ) [复位 = 01h]

图 7-88 显示了 UART\_FIFO\_CTRL 寄存器，表 7-129 中对此进行了介绍。

返回到[汇总表](#)。

用于使能 FIFO 或清除 FIFO 内容的 FIFO 控制字段。

图 7-88. UART\_FIFO\_CTRL 寄存器

7	6	5	4	3	2	1	0
RX_TRG		TX_TRG		RSVD	CLR_TX	CLR_RX	FIFO_EN
R/WP-0h		R/WP-0h		R-0h	RH/W1S-0h	RH/W1S-0h	R/WP-1h

表 7-129. UART\_FIFO\_CTRL 寄存器字段说明

位	字段	类型	复位	说明
7-6	RX_TRG	R/WP	0h	设置 RX FIFO 的触发级别 0h = RX FIFO 中有 1 字节 1h = RX FIFO 中有 4 字节 2h = ( 最大大小 ) - RX FIFO 中有 16 字节 3h = ( 最大大小 ) - RX FIFO 中有 8 字节
5-4	TX_TRG	R/WP	0h	设置 TX FIFO 中空闲空间数的触发级别 0h = 16 字节空闲 1h = 32 字节空闲 2h = ( 最大大小 ) - 16 字节空闲 3h = ( 最大大小 ) - 8 字节空闲
3	RSVD	R	0h	
2	CLR_TX	RH/W1S	0h	清除发送 FIFO 的内容。 清除完成后，硬件将该位复位为 0。 0h = 不执行任何操作 1h = 清除发送 FIFO
1	CLR_RX	RH/W1S	0h	清除接收 FIFO 的内容。 清除完成后，硬件将该位复位为 0。 0h = 不执行任何操作 1h = 清除接收 FIFO
0	FIFO_EN	R/WP	1h	使能 TX FIFO 和 RX FIFO 0h = 禁用发送和接收 FIFO 1h = 使能发送和接收 FIFO

#### 7.6.4.8 UART\_IE\_0 寄存器 ( 偏移 = 2007h ) [复位 = 00h]

图 7-89 显示了 UART\_IE\_0，表 7-130 中对此进行了介绍。

返回到[汇总表](#)。

用于为 INT0 引脚启用特定中断的中断使能位。此处已启用的中断将通过 INT0 引脚发出信号。请注意，必须启用 INT0 功能。

**图 7-89. UART\_IE\_0 寄存器**

7	6	5	4	3	2	1	0
TXLIE0	TXAIE0	RXBRIE0	RXFEIE0	RXPEIE0	RXLIE0	RXFLIE0	RXNIE0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**表 7-130. UART\_IE\_0 寄存器字段说明**

位	字段	类型	复位	说明
7	TXLIE0	R/W	0h	因已满而丢失 TX 字节的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
6	TXAIE0	R/W	0h	TX 空间可用的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
5	RXBRIE0	R/W	0h	接收到 RX 中断的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
4	RXFEIE0	R/W	0h	RX 组帧错误中断的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
3	RXPEIE0	R/W	0h	RX 奇偶校验错误的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
2	RXLIE0	R/W	0h	RX 溢出/丢失字节的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
1	RXFLIE0	R/W	0h	RX 填充级别的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
0	RXNIE0	R/W	0h	RX 新字节的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。

#### 7.6.4.9 UART\_IE\_1 寄存器 ( 偏移 = 2008h ) [复位 = 00h]

图 7-90 显示了 UART\_IE\_1，表 7-131 中对此进行了介绍。

返回到[汇总表](#)。

用于为 INT1 引脚启用特定中断的中断使能位。此处已启用的中断将通过 INT1 引脚发出信号。请注意，必须启用 INT1 功能。

图 7-90. UART\_IE\_1 寄存器

7	6	5	4	3	2	1	0
TXLIE1	TXAIE1	RXBRIE1	RXFEIE1	RXPEIE1	RXLIE1	RXFLIE1	RXNIE0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-131. UART\_IE\_1 寄存器字段说明

位	字段	类型	复位	说明
7	TXLIE1	R/W	0h	因已满而丢失 TX 字节的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
6	TXAIE1	R/W	0h	TX 空间可用的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
5	RXBRIE1	R/W	0h	接收到 RX 中断的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
4	RXFEIE1	R/W	0h	RX 组帧错误中断的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
3	RXPEIE1	R/W	0h	RX 奇偶校验错误的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
2	RXLIE1	R/W	0h	RX 溢出/丢失字节的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
1	RXFLIE1	R/W	0h	RX 填充级别的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
0	RXNIE0	R/W	0h	RX 新字节的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。

#### 7.6.4.10 UART\_IR 寄存器 ( 偏移 = 2009h ) [复位 = 00h]

图 7-91 显示了 UART\_IR 寄存器，表 7-132 中对此进行了介绍。

返回到[汇总表](#)。

挂起中断寄存器，包含当前置位的所有中断的位。写入 1 以进行清除。

**图 7-91. UART\_IR 寄存器**

7	6	5	4	3	2	1	0
TXL	TXA	RXBR	RXFE	RXPE	RXL	RXFL	RXN
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

**表 7-132. UART\_IR 寄存器字段说明**

位	字段	类型	复位	说明
7	TXL	R/W1C	0h	因已满而导致 TX 丢失字节 0h = 无 TX FIFO 字节丢失 1h = 尝试写入时，由于 FIFO 已满，TX FIFO 丢失了一个字节（溢出）
6	TXA	R/W1C	0h	TX 空间可用  <div style="text-align: center;"><b>备注</b></div> <div style="text-align: center;">当 UART IP 被使能时，由于 FIFO 为空，此中断将置位</div> 0h = 无 TX FIFO 空间可用中断 1h = TX FIFO 已达到触发级别阈值
5	RXBR	R/W1C	0h	收到 RX 中断 0h = 未收到中断 1h = 收到中断
4	RXFE	R/W1C	0h	RX 组帧错误 0h = 未发生接收组帧错误 1h = 发生了 RX 组帧错误
3	RXPE	R/W1C	0h	RX 奇偶校验错误 0h = 未发生接收奇偶校验错误 1h = 发生了 RX 奇偶校验错误
2	RXL	R/W1C	0h	RX 溢出/丢失字节 0h = 无 RX FIFO 字节丢失（无溢出）中断 1h = 因 RX FIFO 已满，至少丢失了 1 个字节（溢出中断）
1	RXFL	R/W1C	0h	RX 填充级别 0h = 无 RX FIFO 填充级别中断 1h = RX FIFO 达到填充级别
0	RXN	R/W1C	0h	RX 新字节 0h = 未接收到新字节中断 1h = 已收到新字节

#### 7.6.4.11 UART\_STATUS 寄存器 ( 偏移 = 200Ah ) [复位 = 60h]

图 7-92 显示了 UART\_STATUS 寄存器，表 7-133 中对此进行了介绍。

返回到[汇总表](#)。

图 7-92. UART\_STATUS 寄存器

7	6	5	4	3	2	1	0
RXBS	TXFSRE	TXFE	RSVD				RXFB
RH-0h	RH-1h	RH-1h	R-0h				RH-0h

表 7-133. UART\_STATUS 寄存器字段说明

位	字段	类型	复位	说明
7	RXBS	RH	0h	RX FIFO 中至少有 1 个字节存在奇偶校验错误、组帧错误或中断 0h = RX FIFO 中的任何字节均无奇偶校验错误、组帧错误或中断 1h = RX FIFO 中至少有 1 个字节存在奇偶校验错误、组帧错误或中断
6	TXFSRE	RH	1h	TX FIFO 与移位寄存器是否为空。 这意味着所有排队的数据都已移出。 FIFO 中无数据，移位寄存器中也无数据 0h = TX FIFO 或移位寄存器中至少有 1 个元素 1h = TX FIFO 和移位寄存器均为空
5	TXFE	RH	1h	TX FIFO 是否为空，不必考虑 TX 移位寄存器 0h = TX FIFO 中至少有 1 个元素 1h = TX FIFO 为空
4-1	RSVD	R	0h	
0	RXFB	RH	0h	RX FIFO 是否包含 1 个或多个未读字节 0h = RX FIFO 为空 1h = RX FIFO 中至少有 1 个元素

7.6.4.12 UART\_RXFS 寄存器 ( 偏移 = 200Bh ) [复位 = 00h]

图 7-93 显示了 UART\_RXFS，表 7-134 对其进行了介绍。

返回到[汇总表](#)。

包含 RX FIFO 中存储的未读消息数量计数器。

图 7-93. UART\_RXFS 寄存器

7	6	5	4	3	2	1	0
RX_FL							
R-0h							

表 7-134. UART\_RXFS 寄存器字段说明

位	字段	类型	复位	说明
7-0	RX_FL	R	0h	RX FIFO 填充级别表示当前 RX FIFO 中存储的字节数量。 有效值为 0 至 255。 列出的最大值取决于器件存储器的配置方式。 如果报告 255，则表示 FIFO 中有 255 个或更多字节。

#### 7.6.4.13 UART\_TXFS 寄存器 ( 偏移 = 200Ch ) [复位 = 00h]

图 7-94 显示了 UART\_TXFS , 表 7-135 对其进行了介绍。

返回到[汇总表](#)。

包含 TX FIFO 中可用空间数量的计数器。

图 7-94. UART\_TXFS 寄存器

7	6	5	4	3	2	1	0
TX_SA							
R-0h							

表 7-135. UART\_TXFS 寄存器字段说明

位	字段	类型	复位	说明
7-0	TX_SA	R	0h	TX FIFO 填充级别表示当前 TX FIFO 中的空闲空间

#### 7.6.4.14 UART\_RX\_FIFO 寄存器 ( 偏移 = 2010h ) [复位 = 00h]

图 7-95 显示了 UART\_RX\_FIFO 寄存器，表 7-136 中对此进行了介绍。

返回到[汇总表](#)。

读取下一个 FIFO 元素，并返回不超过 CAN 消息所能容纳的请求字节数的数据。数据末尾附加一个状态字节，用于指示读取的数据段中是否有任何字节处于非正常状态。

图 7-95. UART\_RX\_FIFO 寄存器

7	6	5	4	3	2	1	0
DATA							
R-0h							

表 7-136. UART\_RX\_FIFO 寄存器字段说明

位	字段	类型	复位	说明
7-0	DATA	R	0h	<p>读取 RX FIFO 中下一个数据字节。</p> <p>建议在执行突发读取之前读取 UART_RXFS 寄存器，以查看 FIFO 中有多少字节。</p> <p>对此寄存器进行的突发读取将返回多个数据字节，最多为请求的字节数。</p> <p>在数据字节末尾，会附加一个全局状态字节，用于指示读出的数据字节中是否有任何记录的记录。</p> <p>如果状态字节返回非正常状态，则对 UART_RX_ERR_STATUS 寄存器进行相同长度的读取，将为每个读出的对应数据字节提供一个状态字节，方便用户确定哪个字节处于非正常状态。</p>



7.6.4.15 UART\_TX\_FIFO 寄存器 ( 偏移 = 2010h ) [复位 = 00h]

图 7-96 显示了 UART\_TX\_FIFO , 表 7-137 对其进行了介绍。

返回到[汇总表](#)。

向 UART 发送 FIFO 写入数据时，字节一经写入缓冲区便会进入发送队列等待传输。

图 7-96. UART\_TX\_FIFO 寄存器

7	6	5	4	3	2	1	0
DATA							
W-0h							

表 7-137. UART\_TX\_FIFO 寄存器字段说明

位	字段	类型	复位	说明
7-0	DATA	W	0h	向此 FIFO 写入数据将开始把数据移出到 UART 总线。 建议用户在写入前读取 UART_TXFS 寄存器，查看有多少可用空间。

#### 7.6.4.16 UART\_RX\_ERR\_STATUS 寄存器 ( 偏移 = 2011h ) [复位 = 00h]

图 7-97 显示了 UART\_RX\_ERR\_STATUS 寄存器，表 7-138 中对此进行了介绍。

返回到[汇总表](#)。

这应在读取 UART\_RX\_FIFO 元素之后读取。如果状态字节返回至少 1 个字节存在非正常状态，则从此寄存器读取相同长度的数据将返回最后从 RX FIFO 读取的每个字节的状态字节。

**图 7-97. UART\_RX\_ERR\_STATUS 寄存器**

7	6	5	4	3	2	1	0
RSVD			BRK	FE	PAR	NO_RX	NO_ERR
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h

**表 7-138. UART\_RX\_ERR\_STATUS 寄存器字段说明**

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4	BRK	R	0h	检测到中断条件时置位。 数据字节作为 0x00 返回、用于中断 0h = 不是中断条件 1h = 是中断
3	FE	R	0h	当置位时，表示接收的字节存在帧错误（停止位数量无效） 0h = 字节组帧有效 1h = 字节组帧无效
2	PAR	R	0h	当置位时，表示接收的字节存在奇偶校验位无效 0h = 奇偶校验位有效 1h = 奇偶校验位无效
1	NO_RX	R	0h	读取空的 RX FIFO 将返回 0x00，此位表示数据无效/未接收到。 这表示未接收到数据 0h = 非空/未接收到字节 1h = 该字节无效/未接收且为空
0	NO_ERR	R	0h	当置位时，表明该字节接收过程无任何错误，为有效字节。 如果没有设置其他位、则会设置该位 0h = 存在一些非正常状态 1h = 成功接收到字节

#### 7.6.5 UART 数据 FIFO

FIFO 用于为 UART 传输数据。该接口使用多个 FIFO

- UART 发送 FIFO (h2010)：用于将数据字节排队以发送到 UART 接口
- UART 接收 FIFO (h2010)：用于读取 UART 上接收的字节，包含全局状态字节。
- UART 接收错误状态 (h2011)：用于检查上一次读取中接收到的每个字节的状态

##### 7.6.5.1 UART 发送 FIFO ( 地址 = h2010 )

发送 FIFO 为只写模式，当向其地址发送写入命令时，数据会被写入其中。若模块处于空闲状态，一旦有字节写入 FIFO，传输将立即开始。UART 接口按字节处理数据。

有关更多信息和示例，请参阅 [UART 控制协议](#)。

#### 备注

如果发送 FIFO 已满，写入的任何新字节都将被丢弃，且对应的中断会被置位。

表 7-139. UART 发送 FIFO 说明

位	字段	类型	复位	说明
7:0	DATA	W	0	向此 FIFO 写入数据将开始把数据移出到 UART 总线。建议用户在写入前读取 UART_TXFS 寄存器，查看有多少可用空间。

### 7.6.5.2 UART 接收 FIFO (地址 = h2010)

接收 FIFO 为只读，用于读出 RX FIFO 中的下一个数据字节。建议在执行突发读取之前读取 UART\_RXFS 寄存器，以查看 FIFO 中有多少字节。对此寄存器进行的突发读取将返回多个数据字节，最多为请求的字节数。

在数据字节的末尾附加了一个全局状态字节，这是所有字节状态的逻辑按位或结果。这用于指示所读取的任何字节是否有任何非正常状态。如果状态字节返回非正常状态，读取 UART\_RX\_ERR\_STATUS 寄存器会得到最近读取的每个对应数据字节的状态字节，便于用户确定哪个字节有非正常状态。

正常的全局状态字节返回 0x01。0x00 表示发生了无效/读取溢出。

从空的 RX 缓冲区读取将返回 0，状态字节为 0。帧的读取顺序始终与接收顺序一致。

通过向 UART 清除 RX FIFO 位 (UART\_FIFO\_CTRL[1]) 写入 1，可以清空 RX 缓冲区 (丢弃所有帧)。

有关更多信息和示例，请参阅 [UART 控制协议](#)。

#### 备注

如果接收 FIFO 已满，所有新传入的消息都会被丢弃，并设置一个中断以提醒处理器有消息丢失。

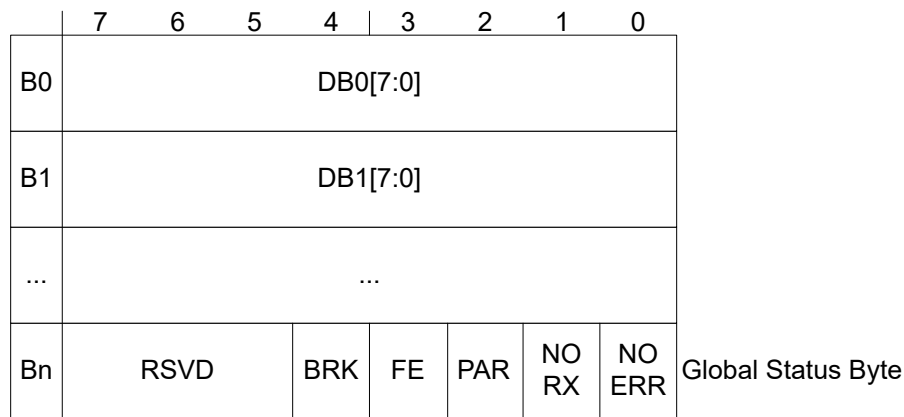


图 7-98. UART 接收 FIFO 突发读取

表 7-140. UART 接收 FIFO 说明 (含全局状态字节)

字节	位	字段	类型	复位	说明
0 到 N-1	7:0	DATA	R	0	接收到的数据字节

表 7-140. UART 接收 FIFO 说明 ( 含全局状态字节 ) ( 续 )

字节	位	字段	类型	复位	说明
N ( 最后一个 )	7:5	RSVD	R	0x0	保留
	4	BRK	R	0	检测到中断条件时置位。中断状态下返回的数据字节为 0x00 0 = 未在字节中检测到中断 1 = 至少有一个字节为中断状态
	3	帧	R	0	当置位时, 表示收到带有帧错误 ( 停止位数量无效 ) 的字节 0 = 未在字节中检测到帧错误 1 = 至少有一个字节存在帧错误
	2	PAR	R	0	奇偶校验错误。当置位时, 表示至少有 1 个字节收到了无效的奇偶校验位。 0 = 所有奇偶校验位均有效 1 = 至少有一个奇偶校验位不正确
	1	NO_RX	R	0	空字节/无 RX。当置位时, 表示至少有 1 个字节为无效字节, 且已发生读取溢出。 0 = 所有字节均有效 1 = 至少有一个字节无效
	0	NO_ERR	R	0	无错误。当被置位时, 表示所有字节均已成功接收。 0 = 至少有一个字节存在非正常状态 1 = 成功接收了所有字节。

### 7.6.5.3 UART 接收错误状态 (地址 = h2011)

UART 接收错误状态寄存器的工作机制与 UART 接收 FIFO 类似，不同之处在于，它返回最近读取的每个 UART 字节对应的独立状态字节。

当处理器从 UART RX FIFO 中读取数据时，对应字节的状态字节会被加载到 UART 接收错误状态 [FIFO] 中。一般步骤是从 UART RX FIFO 读取，检查全局状态寄存器以查看是否有任何字节报告非正常状态。如果返回非正常状态字节，则处理器应读取与从 UART RX FIFO 中读取的 UART 数据字节数量相对应的 UART 接收错误状态。这将返回最近从 UART RX FIFO 读取的每个 UART 数据字节的状态字节。

读取 UART RX FIFO 后，该寄存器的内容将被清除，使用最新读取的 UART 数据字节的状态字节更新该寄存器的内容。

有关更多信息和示例，请参阅 [UART 控制协议](#)。状态字节值如下所示。

图 7-99. UART RX FIFO 错误状态寄存器

7	6	5	4	3	2	1	0
RSVD		BRK		FE	PAR	NO_RX	NO_ERR
R-0h		R-0h		R-0h	R-0h	R-0h	R-0h

表 7-141. UART RX FIFO 错误状态寄存器字段说明

位	字段	类型	默认值	说明
7-5	RSVD	R	0h	
4	BRK	R	0h	检测到中断条件时置位。 中断时数据字节返回 0x00 0h = 不是中断条件 1h = 是中断
3	FE	R	0h	当置位时，表示接收的字节有帧错误 (停止位计数无效) 0h = 字节组帧有效 1h = 字节组帧无效
2	PAR	R	0h	当置位时，表示接收的字节有无效的奇偶校验位 0h = 奇偶校验位有效 1h = 奇偶校验位无效
1	NO_RX	R	0h	读取空的 RX FIFO 将返回 0x00，此位表示数据无效/未接收到。 这表示没有接收到数据 0h = 不是空/未接收的字节 1h = 此字节无效/未接收且为空
0	NO_ERR	R	0h	当置位时，表明该字节接收过程无任何错误，为有效字节。 如果没有设置其他位，则设置此位 0h = 存在一些非正常状态 1h = 字节接收成功

### 7.6.6 I2C 寄存器

节 7.6.6 列出了 I2C 寄存器的存储器映射寄存器。节 7.6.6 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

表 7-142. I2C 寄存器

偏移	首字母缩写词	寄存器名称	部分
3000h	I2C_CREL	I2C 内核版本	<a href="#">节 7.6.6.1</a>
3001h	I2C_SCRATCH	I2C 暂存区	<a href="#">节 7.6.6.2</a>
3002h	I2C_CTRL	I2C 控制	<a href="#">节 7.6.6.3</a>
3003h	I2C_BR	I2C 波特率	<a href="#">节 7.6.6.4</a>
3004h	I2C_FIFO_CTRL	I2C FIFO 控制	<a href="#">节 7.6.6.5</a>
3005h	I2C_IE_0	I2C 中断使能 0	<a href="#">节 7.6.6.6</a>

表 7-142. I2C 寄存器 (续)

偏移	首字母缩写词	寄存器名称	部分
3006h	I2C_IE_1	I2C 中断使能 1	节 7.6.6.7
3007h	I2C_IR	I2C 中断状态	节 7.6.6.8
3008h	I2C_STATUS	I2C 状态	节 7.6.6.9
3009h	I2C_FS	I2C FIFO 状态	节 7.6.6.10
300Ah	I2C_RXFS	I2C RX FIFO 状态	节 7.6.6.11
300Bh	I2C_TXFS	I2C TX FIFO 状态	节 7.6.6.12
300Ch	I2C_TXES	I2C TX 元素状态	节 7.6.6.13

复杂的位访问类型经过编码可适应小型表单元。节 7.6.6 展示了适用于此部分中访问类型的代码。

表 7-143. I2C 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
RH	R H	读取 由硬件置位或清零
写入类型		
W	W	写入
W1C	W 1C	写入 1 以清零
W1S	W 1S	写入 1 以进行设置
WP	W P	写入 需要特权访问
复位或默认值		
-n		复位后的值或默认值

### 7.6.6.1 I2C\_CREL 寄存器 ( 偏移 = 3000h ) [复位 = 10h]

图 7-100 显示了 I2C\_CREL , 表 7-144 中对此进行了介绍。

返回到[汇总表](#)。

I2C IP 发布版本

图 7-100. I2C\_CREL 寄存器

7	6	5	4	3	2	1	0
MAJOR				MINOR			
R-1h				R-0h			

表 7-144. I2C\_CREL 寄存器字段说明

位	字段	类型	复位	说明
7-4	MAJOR	R	1h	此修订版本的主版本号
3-0	MINOR	R	0h	此修订版本的次版本号



#### 7.6.6.2 I2C\_SCRATCH 寄存器 ( 偏移 = 3001h ) [复位 = 00h]

图 7-101 显示了 I2C\_SCRATCH , 表 7-145 中对此进行了介绍。

返回到[汇总表](#)。

客户暂存寄存器

图 7-101. I2C\_SCRATCH 寄存器

7	6	5	4	3	2	1	0
SCRATCHPAD							
R/W-0h							

表 7-145. I2C\_SCRATCH 寄存器字段说明

位	字段	类型	复位	说明
7-0	SCRATCHPAD	R/W	0h	客户暂存寄存器。 这可用于通过写入和读取值来验证通信，也可用于存储一些数据。 该寄存器不会在复位/上电复位后保留数据。

### 7.6.6.3 I2C\_CTRL 寄存器 ( 偏移 = 3002h ) [复位 = 18h]

图 7-102 显示了 I2C\_CTRL，表 7-146 中对此进行了介绍。

返回到[汇总表](#)。

此寄存器控制 I2C IP

**图 7-102. I2C\_CTRL 寄存器**

7	6	5	4	3	2	1	0
SBR_START	AUTO_SBR_EN	RSVD	NACK_FAIL_D	NACK_FAIL_A	LSM	I2C_EN	CCE
RH/W1S-0h	R/WP-0h	R-0h	R/WP-1h	R/WP-1h	R/WP-0h	RH-0h	R/W-0h

**表 7-146. I2C\_CTRL 寄存器字段说明**

位	字段	类型	复位	说明
7	SBR_START	RH/W1S	0h	手动生成 I2C 总线卡滞恢复序列。 复位完成后由 IP 自动清除 0h = 未生成复位模式 1h = 开始 I2C 总线卡滞恢复序列
6	AUTO_SBR_EN	R/WP	0h	如果器件检测到总线数据线路卡滞，则启用自动总线卡滞恢复序列。 0h = 如果检测到总线卡滞，不执行任何操作 1h = 如果检测到总线卡滞，自动生成总线卡滞恢复序列
5	RSVD	R	0h	
4	NACK_FAIL_D	R/WP	1h	如果在数据字段中收到 NACK，则认为消息失败。 失败的消息将终止传输并发出停止信号。 0h = 记录 NACK 标志并继续传输 ( 消息仍被视为成功 ) 1h = 如果意外收到 NACK，则终止事务
3	NACK_FAIL_A	R/WP	1h	如果在地址 ( 第一个字节 ) 之后收到 NACK，则认为消息失败。 失败的消息将终止传输并发出停止信号。 0h = 记录 NACK 标志并继续传输 ( 消息仍被视为成功 ) 1h = 如果意外收到 NACK，则终止事务
2	LSM	R/WP	0h	将输入时钟从 20MHz 改为 5MHz，以获得更低的 I2C 速度 0h = 高速时钟，10MHz 1h = 低速时钟，2.5MHz ( 用于低于 100KHz )
1	I2C_EN	RH	0h	I2C IP 使能状态标志。 此标志不可写，但如果通过从 MRAM 向 IP 分配内存来启用 I2C IP，则会设置此标志。 这通过 <a href="#">MRAM_IP_CFG</a> 寄存器完成 0h=I2C IP 被禁用，所有 I2C 功能被禁用 1h=I2C IP 被启用
0	CCE	R/W	0h	I2C IP 更改控制使能位。 如果设置了该位，则 I2C IP 保持在复位状态 0h = I2C 配置寄存器受写保护 1h = 可以更改 I2C 配置寄存器

**7.6.6.4 I2C\_BR 寄存器 ( 偏移 = 3003h ) [复位 = 18h]**

图 7-103 显示了 I2C\_BR , 表 7-147 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-103. I2C\_BR 寄存器**

7	6	5	4	3	2	1	0
BAUD_RATE							
R/WP-18h							

**表 7-147. I2C\_BR 寄存器字段说明**

位	字段	类型	复位	说明
7-0	BAUD_RATE	R/WP	18h	I2C 的 8 位分频器值。 解释为比输入值大 1

### 7.6.6.5 I2C\_FIFO\_CTRL 寄存器 ( 偏移 = 3004h ) [复位 = 00h]

图 7-104 显示了 I2C\_FIFO\_CTRL，表 7-148 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-104. I2C\_FIFO\_CTRL 寄存器**

7	6	5	4	3	2	1	0
RX_TRG		TX_TRG		RSVD	CLR_TX	CLR_RX	RSVD
R/WP-0h		R/WP-0h		R-0h	RH/W1S-0h	RH/W1S-0h	R-0h

**表 7-148. I2C\_FIFO\_CTRL 寄存器字段说明**

位	字段	类型	复位	说明
7-6	RX_TRG	R/WP	0h	<p>设定当 RX FIFO 达到 FIFO 中所存储的指定字节数时，RX FIFO 的触发级别。</p> <p>这允许 CPU 批量移出字节，而非一次 1 个字节，以减少中断次数</p> <p>0h = RX FIFO 中有 4 个字节</p> <p>1h = RX FIFO 中有 8 个字节</p> <p>2h = RX FIFO 中有 [(最大容量) - 128] 个字节</p> <p>3h = RX FIFO 中有 [(最大容量) - 64] 个字节</p>
5-4	TX_TRG	R/WP	0h	<p>设定当 TX FIFO 中的空闲字节数增加到指定数量时的触发级别。</p> <p>用于在 FIFO 中有一定数量的空闲空间时产生中断，以便处理器可以将多个字节加载到 FIFO 中，而不是一次仅加载 1 个字节。</p> <hr/> <p style="text-align: center;"><b>备注</b></p> <p>该中断触发方式类似于边沿触发型中断。</p> <p>如果用户始终保持不低于 8 个空闲空间</p> <hr/> <p>0h = 16 字节空闲</p> <p>1h = 32 字节空闲</p> <p>2h = [ ( 最大大小 ) - 128] 字节空闲</p> <p>3h = [ ( 最大大小 ) - 64] 字节空闲</p>
3	RSVD	R	0h	
2	CLR_TX	RH/W1S	0h	<p>清除发送 FIFO 的内容。</p> <p>清除完成后，硬件将该位复位为 0。</p> <p>0h = 不执行任何操作</p> <p>1h = 清除发送 FIFO</p>
1	CLR_RX	RH/W1S	0h	<p>清除接收 FIFO 的内容。</p> <p>清除完成后，硬件将该位复位为 0。</p> <p>0h = 不执行任何操作</p> <p>1h = 清除接收 FIFO</p>
0	RSVD	R	0h	

### 7.6.6.6 I2C\_IE\_0 寄存器 ( 偏移 = 3005h ) [复位 = 00h]

图 7-105 显示了 I2C\_IE\_0，表 7-149 中对此进行了介绍。

返回到[汇总表](#)。

图 7-105. I2C\_IE\_0 寄存器

7	6	5	4	3	2	1	0
TXLIE0	TXAIE0	DNACKIE0	ANACKIE0	SBRCIE0	RXLIE0	RXFLIE0	RXNIE0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

表 7-149. I2C\_IE\_0 寄存器字段说明

位	字段	类型	复位	说明
7	TXLIE0	R/W	0h	因 FIFO 满导致 TX 消息丢失的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
6	TXAIE0	R/W	0h	TX 空间可用的 INT0 中断使能。 触发级别由 I2C_FIFO_CTRL.TX_TRG 设置来设定 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
5	DNACKIE0	R/W	0h	数据 NACK 的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
4	ANACKIE0	R/W	0h	地址 NACK 的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
3	SBRCIE0	R/W	0h	总线卡滞恢复完成中断 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
2	RXLIE0	R/W	0h	RX 溢出/丢失消息的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
1	RXFLIE0	R/W	0h	RX 填充级别的 INT0 中断使能。 触发级别由 I2C_FIFO_CTRL.RX_TRG 设置来设定 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
0	RXNIE0	R/W	0h	RX 新消息的 INT0 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。

### 7.6.6.7 I2C\_IE\_1 寄存器 ( 偏移 = 3006h ) [复位 = 00h]

图 7-106 显示了 I2C\_IE\_1，表 7-150 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-106. I2C\_IE\_1 寄存器**

7	6	5	4	3	2	1	0
TXLIE1	TXAIE1	DNACKIE1	ANACKIE1	SBRCIE1	RXLIE1	RXFLIE1	RXNIE1
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**表 7-150. I2C\_IE\_1 寄存器字段说明**

位	字段	类型	复位	说明
7	TXLIE1	R/W	0h	因 FIFO 满导致 TX 消息丢失的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
6	TXAIE1	R/W	0h	TX 空间可用的 INT1 中断使能。 触发级别由 I2C_FIFO_CTRL.TX_TRG 设置来设定 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 中断被使能。标志置位时将触发中断输出。
5	DNACKIE1	R/W	0h	数据 NACK 的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
4	ANACKIE1	R/W	0h	地址 NACK 的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
3	SBRCIE1	R/W	0h	总线卡滞恢复完成中断 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
2	RXLIE1	R/W	0h	RX 溢出/丢失消息的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
1	RXFLIE1	R/W	0h	RX 填充级别的 INT1 中断使能。 触发级别由 I2C_FIFO_CTRL.RX_TRG 设置来设定 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。
0	RXNIE1	R/W	0h	RX 新消息的 INT1 中断使能 0h = 禁用中断。标志置位时不会触发中断输出。 1h = 禁用中断。标志置位时将触发中断输出。

### 7.6.6.8 I2C\_IR 寄存器 ( 偏移 = 3007h ) [复位 = 00h]

图 7-107 显示了 I2C\_IR，表 7-151 中对此进行了介绍。

返回到[汇总表](#)。

图 7-107. I2C\_IR 寄存器

7	6	5	4	3	2	1	0
TXL	TXA	DNACK	ANACK	SBRC	RXL	RXFL	RXN
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

表 7-151. I2C\_IR 寄存器字段说明

位	字段	类型	复位	说明
7	TXL	R/W1C	0h	因 FIFO 满导致 TX 消息丢失的中断 0h = 无 TX FIFO 字节丢失 1h = 尝试写入时因 FIFO 满，TX FIFO 至少丢失 1 个字节 ( 溢出 )
6	TXA	R/W1C	0h	TX 空间可用的中断。 触发级别由 I2C_FIFO_CTRL.TX_TRG 设置来设定  <b>备注</b> 当 I2C IP 被使能时，由于 FIFO 为空，此中断将置位  0h = 无 TX FIFO 空间可用的中断 1h = TX FIFO 已达到触发级别阈值
5	DNACK	R/W1C	0h	数据 NACK 中断 0h = 未检测到 NACK 1h = 在数据阶段检测到 NACK
4	ANACK	R/W1C	0h	地址 NACK 中断 0h = 未检测到 NACK 1h = 在地址阶段检测到 NACK
3	SBRC	R/W1C	0h	总线卡滞恢复完成中断 0h = 未发生总线卡滞恢复 1h = 总线卡滞恢复已完成。检查 I2C_STATUS 寄存器以查看总线是否仍处于卡滞状态。
2	RXL	R/W1C	0h	RX 溢出/丢失消息中断 0h = RX FIFO 字节未丢失 ( 无溢出 ) 中断 1h = 由于 RX FIFO 满 ( 溢出中断 )，至少丢失了 1 字节
1	RXFL	R/W1C	0h	RX 填充级别中断。 触发级别由 I2C_FIFO_CTRL.RX_TRG 设置来设定 0h = 无 RX FIFO 填充级别的中断 1h = RX FIFO 已达到填充级别
0	RXN	R/W1C	0h	RX 新消息中断 0h = 无新字节接收中断 1h = 已接收到新字节

### 7.6.6.9 I2C\_STATUS 寄存器 ( 偏移 = 3008h ) [复位 = XXh]

图 7-108 显示了 I2C\_STATUS , 表 7-152 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-108. I2C\_STATUS 寄存器**

7	6	5	4	3	2	1	0
SDA_S	SCL_S	SDA_V	SCL_V	RSVD			空闲
RH-0h	RH-0h	RH-Xh	RH-Xh	R-0h			RH-Xh

**表 7-152. I2C\_STATUS 寄存器字段说明**

位	字段	类型	复位	说明
7	SDA_S	RH	0h	SDA 卡滞 0h = SDA 未卡滞 1h = SDA 当前卡在低电平
6	SCL_S	RH	0h	SCL 卡滞 0h = SCL 未卡滞 1h = SCL 当前卡在低电平
5	SDA_V	RH	X	SDA 值 0h = SDA 当前为低电平 1h = SDA 当前为高电平
4	SCL_V	RH	X	SCL 值 0h = SCL 当前为低电平 1h = SCL 当前为高电平
3-1	RSVD	R	0h	
0	空闲	RH	X	I2C IP 是否空闲 0h = I2C IP 繁忙, 正进行事务处理 1h = I2C IP 空闲



**7.6.6.10 I2C\_FS 寄存器 ( 偏移 = 3009h ) [复位 = 00h]**

图 7-109 显示了 I2C\_FS，表 7-153 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-109. I2C\_FS 寄存器**

7	6	5	4	3	2	1	0
TXFSRE	RXFD	RXFB					
RH-0h	RH-0h	RH-0h					

**表 7-153. I2C\_FS 寄存器字段说明**

位	字段	类型	复位	说明
7	TXFSRE	RH	0h	<p>TX FIFO 与移位寄存器是否为空。 这意味着所有排队的数据都已传输，且控制器处于空闲状态。 FIFO 中无数据，移位寄存器中也无数据。 如果此位置位，则配置或禁用 I2C 控制器是安全的</p> <hr/> <p><b>备注</b></p> <p>这不会考虑未完全写入的 TX FIFO 元素。 如果存在部分写入的 TX FIFO 元素，该未发送元素在完全写入前不会清除空闲标志</p> <hr/> <p>0h = 至少有 1 个正在进行和/或待处理的传输。 1h = TX FIFO 和移位寄存器为空且 IP 处于空闲状态。配置或关闭控制器是安全的</p>
6	RXFD	RH	0h	<p>RX FIFO 是否包含 1 个或多个未读字节。 0h = RX FIFO 为空 1h = RX FIFO 中至少有 1 个元素</p>
5-0	RXFB	RH	0h	<p>从下一个 RX FIFO 元素中存储的 I2C 帧里，仍需读取的数据字节数。 有效值为 0 至 63。 任何大于 63 的值都将显示为 63，剩余值将在处理器读出数据时更新。</p> <hr/> <p><b>备注</b></p> <p>这仅指 I2C 帧中的数据字节数。 注意，每次读取 RX FIFO 时，都需要对每个 RX FIFO 读取添加 3 字节的 I2C 帧头</p> <hr/>

7.6.6.11 I2C\_RXFS 寄存器 ( 偏移 = 300Ah ) [复位 = 00h]

图 7-110 显示了 I2C\_RXFS，表 7-154 中对此进行了介绍。

返回到[汇总表](#)。

包含 RX FIFO 中存储的未读 I2C 消息数量的计数器。

图 7-110. I2C\_RXFS 寄存器

7	6	5	4	3	2	1	0
RX_FL							
R-0h							

表 7-154. I2C\_RXFS 寄存器字段说明

位	字段	类型	复位	说明
7-0	RX_FL	R	0h	RX FIFO 填充级别表示当前 RX FIFO 中存储的消息数量。 有效值为 0 至 255。 列出的最大值取决于器件存储器的配置方式。 若报告为 255，则 FIFO 中有 255 条或更多消息。

7.6.6.12 I2C\_TXFS 寄存器 ( 偏移 = 300Bh ) [复位 = 00h]

图 7-111 显示了 I2C\_TXFS，表 7-155 中对此进行了介绍。

返回到[汇总表](#)。

包含 TX FIFO 中可用字节数量的计数器。注意，需包含 I2C 数据包报头的 2 个字节

图 7-111. I2C\_TXFS 寄存器

7	6	5	4	3	2	1	0
TX_SA							
R-0h							

表 7-155. I2C\_TXFS 寄存器字段说明

位	字段	类型	复位	说明
7-0	TX_SA	R	0h	TX FIFO 填充级别表示当前 TX FIFO 中的空闲字节数。 大于 255 的数字将显示为 255。 <div> <div>备注</div> <div>I2C 帧所需的字节数除了数据字节外，还必须包括 2 个字节的 I2C 报头</div> </div>

### 7.6.6.13 I2C\_TXES 寄存器 ( 偏移 = 300Ch ) [复位 = 00h]

图 7-112 显示了 I2C\_TXES，表 7-156 中对此进行了介绍。

返回到[汇总表](#)。

包含有关当前/下一个发送 FIFO 元素的信息

图 7-112. I2C\_TXES 寄存器

7	6	5	4	3	2	1	0
TXEIP	TXEBP						
R-0h	R-0h						

表 7-156. I2C\_TXES 寄存器字段说明

位	字段	类型	复位	说明
7	TXEIP	R	0h	<p>进行中的 TX FIFO 元素写入 用于指示 TX FIFO 元素是否已部分写入。 如果没有足够的数据字节写入 FIFO ( 根据标头 )，则设置该标志。 如果它被清除，那么对 TX FIFO 的下次写入将是一个新元素，应该从 I2C 标头字段开始 0h = TX 元素未在进行中 ( 下次写入 TX FIFO 是一个新元素 ) 1h = TX 元素正在进行中，并且有预期的剩余字节数</p>
6-0	TXEBP	R	0h	<p>TX 元素字节待处理 如果 TX 元素写入已开始，这是完成 FIFO 元素还需要的字节数。</p> <hr/> <p><b>备注</b></p> <p>这包括标头字节，如果传输中的字节数尚未写入，则可以更改值。 例如，如果仅写入了 I2C 地址字节，则该字段将读取 1，因为我们不知道在告知要传输的数据字节数量之前有多少字节处于挂起状态。 传输中的字节数已知后，会在写入数据时对此进行重新计算</p> <hr/>

### 7.6.7 I2C 数据 FIFO

FIFO 用于为 I2C 传输数据。该接口使用 2 个 FIFO

- I2C 发送 FIFO (h3010)：用于将数据字节排队以发送到 I2C 接口
- I2C 接收 FIFO (h3010)：用于读取从 I2C 读取接收到的字节

#### 7.6.7.1 I2C 发送 FIFO ( 地址 = h3010 )

发送 FIFO 为只写模式，当向其地址发送写入命令时，数据会被写入其中。当整个帧写入缓冲区后，传输将开始。在帧结束后 ( 由初始报头指定 ) 写入的所有数据字节都将被忽略，直到开始新的发送 FIFO 写入。对发送 FIFO 的所有写入都应仅指向指定的 I2C 发送 FIFO 地址，因为每个写入的字节都会自动移入 FIFO。例如，当写入的 I2C 帧大于单个 CAN 消息所能传输的大小时，会通过同一地址的多次 CAN 写入来完成，因为 SPI 报头字节会告知设备预期的数据字节数。

TX 缓冲区中每个帧的大小由所需的 I2C 帧长度决定。每个 I2C FIFO 元素的开头都需要一个 2 字节的报头。TX FIFO 中帧的总长度为以下各项之和

- I2C 报头的 2 字节
- 数据字节的消息长度 ( 如果帧是 I2C 写入。I2C 读取仅需要报头中指定的待读取字节数，任何额外的字节都会被忽略 )

有关更多信息和示例，请参阅[节 7.5.6.3](#)。

## 备注

如果发送 FIFO 已满，写入的任何新消息都将被丢弃，且对应的中断会被置位。

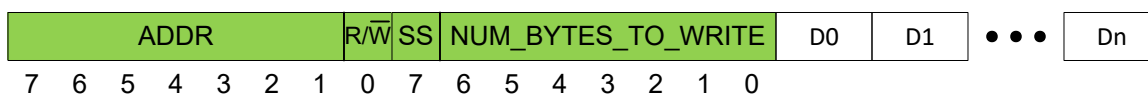


图 7-113. I2C 写入 TX FIFO 报头 ( 仅限报头和数据 )

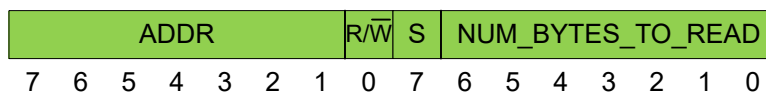


图 7-114. I2C 读取 TX FIFO 报头 ( 仅限报头 )

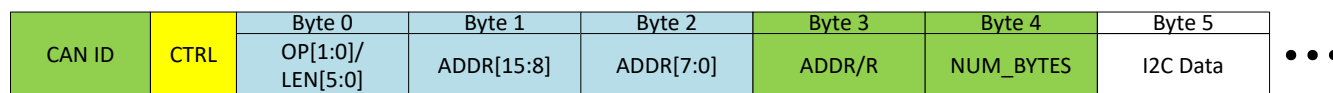


图 7-115. I2C 写入 TX FIFO 报头示例 ( 标准 CAN 帧格式 )

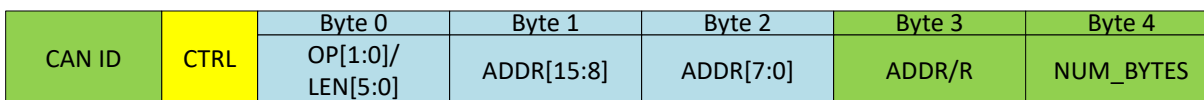


图 7-116. I2C 读取 TX FIFO 报头示例 ( 标准 CAN 帧格式 )

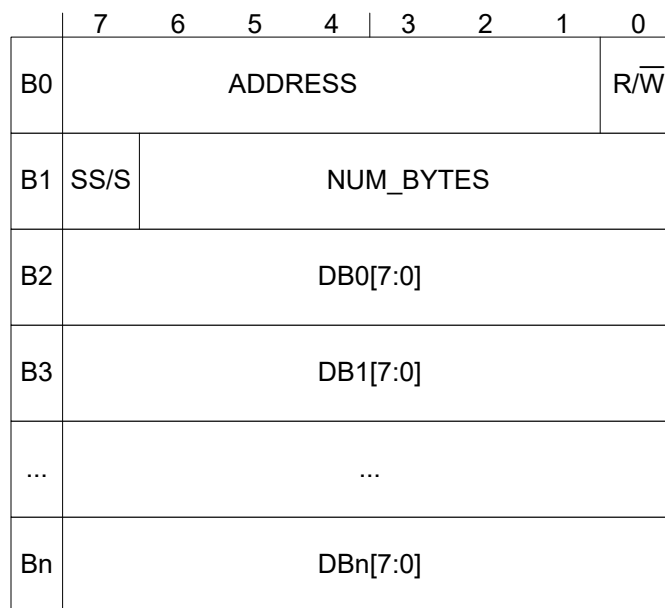


图 7-117. I2C 发送 FIFO

表 7-157. I2C 发送 FIFO 描述

字节	位	字段	类型	复位	说明
0	7:1	地址	W	0x0	I2C 目标器件地址
	0	R/W	W	0	I2C 消息是 I2C 读取还是 I2C 写入 0 = I2C 写入 1 = I2C 读取

**表 7-157. I2C 发送 FIFO 描述 (续)**

字节	位	字段	类型	复位	说明
1	7	S ( I2C 读取 ) SS ( I2C 写入 )	W	0	存储 ( I2C 读取 ) 或存储状态 ( I2C 写入 ) 对于 I2C 读取, 将接收到的数据存储到接收 FIFO 中。对于 I2C 写入, 由于不会从目标器件接收数据, 用户可以选择是否存储 I2C 写入的状态帧, 或者假设写入成功 ( 不在接收 FIFO 中存储任何状态 ) 0 = 不将此事务的任何数据/状态存储到接收 FIFO 1 = 将数据或写入状态存储到接收 FIFO
	6:0	NUM_BYTES_TO_WRITE/READ	W	0x0	I2C 写入 (B0.R/W = 0): 地址之后的 I2C 帧中的字节数 ( 不包括上述报头字节 ) 有效值 0-127。0 将不发送任何数据, 仅传输器件地址 ( 对于检查 ACK 很有用 ) I2C 读取 (B0.R/W = 1): 地址字节之后待读取的字节数 ( 不包括上述报头字节 )

### 7.6.7.2 I2C 接收 FIFO (地址 = h3010)

接收 FIFO 为只读。

RX 缓冲区中每个帧的大小由 I2C 帧的数据长度决定。缓冲区中 RX 帧的总长度为以下各项之和

- I2C 报头的 2 字节
- I2C 状态的 1 字节
- 数据字节的消息长度

当从 FIFO 元素中读取了所有数据字节后，对 RX FIFO 的读取被视为完成且该 FIFO 元素被释放。当读取数据时，若读取的字节数超过剩余可读取字节数，超出部分的无效字节将返回 0x00。

从 RX FIFO 中对一帧数据进行部分读取后，下一次对 I2C 接收 FIFO 的读取会自动延续该帧的剩余部分，此时 CONT 位会被置为 1 以标识这种延续状态，同时 NUM\_BYTES\_REMAINING 会显示该 I2C 帧中尚待读取的数据字节数。

从空的 RX 缓冲区读取时，通道 0 的 NUM\_BYTES\_REMAINING 返回 0。帧的读取顺序始终与接收顺序一致。

通过向 I2C 清除 RX FIFO 位 (I2C\_FIFO\_CTRL[1]) 写入 1，可以清空 RX 缓冲区 (丢弃所有帧)。

#### 备注

如果接收 FIFO 已满，所有新传入的消息都会被丢弃，并设置一个中断以提醒处理器有消息丢失。

有关更多信息和示例，请参阅节 7.5.6.3。



图 7-118. I2C RX FIFO 读取报头 (仅限报头和数据)

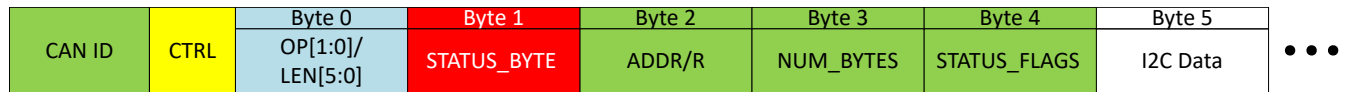


图 7-119. I2C RX FIFO 报头示例 (标准 CAN 帧格式)

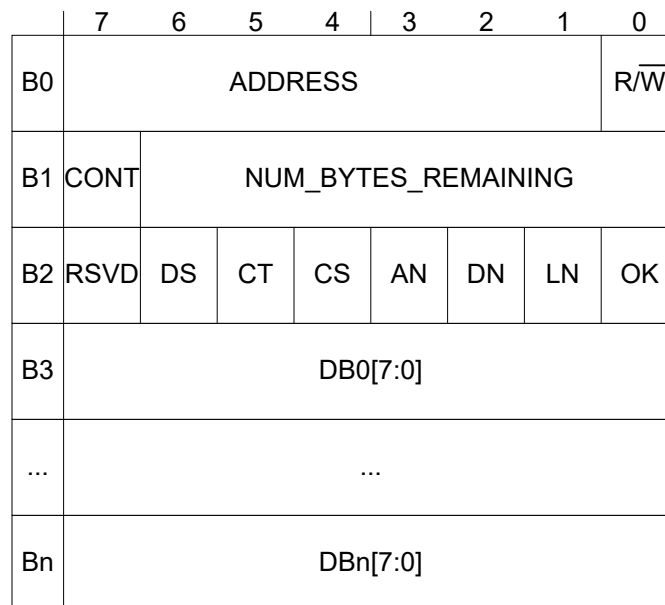


图 7-120. I2C 接收 FIFO 帧格式

**表 7-158. I2C 接收 FIFO 字节说明**

字节	位	字段	类型	复位	说明
0	7:1	地址	R	0x0	I2C 目标器件地址
	0	W/R	R	0	I2C 消息是 I2C 读取还是 I2C 写入 0 = I2C 读取 1 = I2C 写入
1	7	CONT	R	0	继续读取 0 = 此次读取为帧数据的起始部分 1 = 此次读取为前一次读取的继续
	6:0	NUM_BYTES_REMAINING	R	0x0	I2C 读取 (B0.W/R = 0) : 地址字节之后待读取的字节数 (不包括上述任何报头字节) I2C 写入 (B0.W/R = 1) : 地址之后的 I2C 帧中的字节数 (不包括上述报头字节) 有效值 0-127。0 将不发送任何数据, 仅传输器件地址 (对于检查 ACK 很有用)
2	7	RSVD	R	0	保留
	6	DS	R	0	数据卡滞 总线卡滞 (时钟或数据未释放)
	5	CT	R	0	时钟超时 检测到 SMBus 时钟超时
	4	CS	R	0	时钟拉伸 该帧期间发生了时钟拉伸
	3	AN	R	0	地址 NACK I2C 目标器件对地址字节进行了非应答 (NACK)
	2	DN	R	0	数据 NACK I2C 目标器件在写入过程中对至少 1 个字节进行了非应答 (NACK)。
	1	LN	R	0	最后一个字节 NACK I2C 目标器件对 I2C 帧的最后一个字节进行了非应答
	0	OK	R	0	成功 成功完成 I2C 帧

### 7.6.8 PWM0 寄存器

节 7.6.8 列出了 PWM0 寄存器的存储器映射寄存器。节 7.6.8 中未列出的所有寄存器偏移地址都应视为保留的位置, 并且不应修改寄存器内容。

**表 7-159. PWM0 寄存器**

偏移	首字母缩写词	寄存器名称	部分
4000h	PWM1_ACTION		节 7.6.8.1
4000h	PWM0_CTRL	PWM0 控制	节 7.6.8.2
4001h	PWM0_IE0	中断启用	节 7.6.8.3
4002h	PWM0_IE1	中断启用	节 7.6.8.4
4003h	PWM0_IR	中断寄存器	节 7.6.8.5
4004h	PWM0_STATUS	状态	节 7.6.8.6
4005h + 公式	PWM0_CUR_PULSE[y]	当前脉冲计数	节 7.6.8.7
4009h	PWM0_CUR_VAL_MSB	当前值 MSB	节 7.6.8.8
400Ah	PWM0_CUR_VAL_LSB	当前值 LSB	节 7.6.8.9
400Bh	PWM0_CONST_MSB	常量值整数 MSB	节 7.6.8.10
400Ch	PWM0_CONST_LSB	常量值整数 LSB	节 7.6.8.11
400Dh	PWM0_STOP_VAL_FRAC_F	停止小数 (仅频率)	节 7.6.8.12
400Eh	PWM0_STOP_VAL_MSB	停止值整数 MSB	节 7.6.8.13



表 7-159. PWM0 寄存器 (续)

偏移	首字母缩写词	寄存器名称	部分
400Fh	PWM0_STOP_VAL_LSB	停止值整数 LSB	节 7.6.8.14
4010h	PWM0_STOP_SL_MSB	停止斜率 MSB	节 7.6.8.15
4011h	PWM0_STOP_SL_MID	停止斜率中间位字节	节 7.6.8.16
4012h	PWM0_STOP_SL_LSB	停止斜率 LSB	节 7.6.8.17
4013h	PWM0_START_VAL_FRAC_F	起始小数值 (仅频率)	节 7.6.8.18
4014h	PWM0_START_VAL_MSB	起始值 MSB	节 7.6.8.19
4015h	PWM0_START_VAL_LSB	起始值 LSB	节 7.6.8.20
4016h	PWM0_START_SL_MSB	起始斜率 MSB	节 7.6.8.21
4017h	PWM0_START_SL_MID	起始斜率中间字节	节 7.6.8.22
4018h	PWM0_START_SL_LSB	起始斜率 LSB	节 7.6.8.23
4019h	PWM0_END_VAL_CONST_FRAC_F	结束 (频率) 或恒定 (占空比) 小数值	节 7.6.8.24
401Ah	PWM0_END_VAL_MSB	结束值 MSB	节 7.6.8.25
401Bh	PWM0_END_VAL_LSB	结束值 LSB	节 7.6.8.26
401Ch + 公式	PWM0_PULSE_STOP_RAMP[y]	开始停止斜坡的脉冲计数	节 7.6.8.27
4020h + 公式	PWM0_PULSE_MAX[y]	脉冲计数最大值	节 7.6.8.28
4024h	PWM0_ACTION	起始动作寄存器	节 7.6.8.29
4030h	PWM0_IAS_CTRL	输入自动停止控制	节 7.6.8.30

复杂的位访问类型经过编码可适应小型表单元。节 7.6.8 展示了适用于此部分中访问类型的代码。

表 7-160. PWM0 访问类型代码

访问类型	代码	说明
读取类型		
R	R	读取
RH	R H	读取 由硬件置位或清零
写入类型		
W	W	写入
W1C	W 1C	写入 1 以清零
W1S	W 1S	写入 1 以进行设置
WP	W P	写入 需要特权访问
复位或默认值		
-n		复位后的值或默认值
寄存器数组变量		
i、j、k、l、m、n		当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中寄存器是一组重复寄存器的一部分。寄存器组构成分层结构，数组用公式表示。
y		当该变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。

7.6.8.1 PWM1\_ACTION 寄存器 ( 偏移 = 4000h ) [复位 = 00h]

PWM1\_ACTION 如表 7-161 所示。

返回到[汇总表](#)。

表 7-161. PWM1\_ACTION 寄存器字段说明

位	字段	类型	复位	说明
4	COUNT_RST	R/W1S	0h	将脉冲计数器复位为 0 如果设置此位，则脉冲计数器复位。 通常在斜坡开始时执行此操作，以跟踪脉冲总数。 复位完成后，硬件会将其清除。 <div><div>备注</div><div>如果使用 AUTO_STOP，请注意停止，以及最大脉冲计数值。 这些脉冲计数器与当前脉冲计数器进行比较，如果最大脉冲计数小于当前脉冲计数器值，则不会生成 PWM 输出</div></div>
3-0	RESERVED	R	0h	

### 7.6.8.2 PWM0\_CTRL 寄存器 ( 偏移 = 4000h ) [复位 = 11h]

图 7-121 显示了 PWM0\_CTRL，表 7-162 中对此进行了介绍。

返回到[汇总表](#)。

用于配置 PWM0 模块的控制字段

图 7-121. PWM0\_CTRL 寄存器

7	6	5	4	3	2	1	0
RSVD	SLOPE_SCALE			模式		DC_8B	INIT
R-0h	R/WP-1h			R/WP-0h		R/WP-0h	R/W-1h

表 7-162. PWM0\_CTRL 寄存器字段说明

位	字段	类型	复位	说明
7	RSVD	R	0h	
6-4	SLOPE_SCALE	R/WP	1h	斜率标度 调整用于斜率寄存器的整数与小数部分的位数。 枚举值的含义会根据正在进行斜坡变化的是频率还是占空比而有所不同。 有关更多信息，请参阅节 7.5.7.4 0h = 整数 [20-16]，小数 [15-0] ( 频率 ) - 整数 [20]，小数 [19-0] ( 占空比 ) 1h = 整数 [20-12]，小数 [11-0] ( 频率 ) - 整数 [20-19]，小数 [18-0] ( 占空比 ) 2h = 整数 [20-8]，小数 [7-0] ( 频率 ) - 整数 [20-18]，小数 [17-0] ( 占空比 ) 3h = 整数 [20-4]，小数 [3-0] ( 频率 ) - 整数 [20-17]，小数 [16-0] ( 占空比 ) 4h = 整数 [20-0]，无小数 ( 频率 ) - 整数 [20-16]，小数 [15-0] ( 占空比 )
3-2	模式	R/WP	0h	调整用于斜率寄存器整数部分与小数部分的位数。 请注意，如果 GPIO 配置为特殊功能，那么将模式设置为非关闭状态将优先于 SPI 片选。 0h = 关闭 1h = 占空比斜坡控制 2h = 频率斜坡控制 3h = 静态开启
1	DC_8B	R/WP	0h	8 位占空比模式 0h = 10 位占空比分辨率 1h = 8 位占空比分辨率
0	INIT	R/W	1h	初始化模式 将 PWM IP 保持在复位状态，用于配置此 PWM0_CTRL 寄存器中的设置 0h = 禁用初始化模式。不允许写入受保护的 PWM 位 1h = 初始化模式已使能。允许对受保护的 PWM 位进行写入

### 7.6.8.3 PWM0\_IE0 寄存器 ( 偏移 = 4001h ) [复位 = 00h]

图 7-122 显示了 PWM0\_IE0，表 7-163 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-122. PWM0\_IE0 寄存器**

7	6	5	4	3	2	1	0
RSVD					PULSE_OVF_I E	IAS_IE	RC_IE
R-0h					R/W-0h	R/W-0h	R/W-0h

**表 7-163. PWM0\_IE0 寄存器字段说明**

位	字段	类型	复位	说明
7-3	RSVD	R	0h	保留
2	PULSE_OVF_IE	R/W	0h	当前脉冲溢出 当 CUR_PULSE 计数器溢出时置位 0h = 当前脉冲计数器未溢出 1h = 当前脉冲计数器已溢出
1	IAS_IE	R/W	0h	输入自动停止中断 当输入自动停止条件发生时置位；若 PWM 配置为立即停止，则输出也将停止（由 RC 位反映）。 如果 PWM 配置为在 IAS 触发时开始停止斜坡，则此中断在输入条件发生时立即置位，且输出关闭时 RC 位置位。 0h = 未触发 IAS 的输入 1h = 触发了 IAS 的输入
0	RC_IE	R/W	0h	PWM 斜坡完成 当斜坡结束时置位。如果 AS_EN = 1（自动停止），则在 PWM 通道关闭时置位。 如果 AS_EN=0，则在 PWM 达到结束值时置位。 如果 IAS_EN = 1（输入自动停止），则在 PWM 输出停止后置位。 如果手动激活停止斜坡，则 RC 位将在斜坡完成且输出关闭后置位。  <div style="text-align: center;"><b>备注</b></div> 如果正在进行导通斜坡并且发生停止条件，则 RC 位不会为导通斜坡置位，而只会在输出因停止条件而停止时置位  0h = PWM 通道斜坡未完成 1h = PWM 通道斜坡完成

#### 7.6.8.4 PWM0\_IE1 寄存器 ( 偏移 = 4002h ) [复位 = 00h]

图 7-123 显示了 PWM0\_IE1 , 表 7-164 中对此进行了介绍。

返回到[汇总表](#)。

图 7-123. PWM0\_IE1 寄存器

7	6	5	4	3	2	1	0
RSVD					PULSE_OVF_I E	IAS_IE	RC_IE
R-0h					R/W-0h	R/W-0h	R/W-0h

表 7-164. PWM0\_IE1 寄存器字段说明

位	字段	类型	复位	说明
7-3	RSVD	R	0h	保留
2	PULSE_OVF_IE	R/W	0h	当前脉冲溢出 当 CUR_PULSE 计数器溢出时置位 0h = 当前脉冲计数器未溢出 1h = 当前脉冲计数器已溢出
1	IAS_IE	R/W	0h	输入自动停止中断 当输入自动停止条件发生时置位；若 PWM 配置为立即停止，则输出也将停止（由 RC 位反映）。 如果 PWM 配置为在 IAS 触发时开始停止斜坡，则此中断在输入条件发生时立即置位，且输出关闭时 RC 位置位。 如果手动激活停止斜坡，则 RC 位将在斜坡完成且输出关闭后置位。 0h = 未触发 IAS 的输入 1h = 触发了 IAS 的输入
0	RC_IE	R/W	0h	<p>PWM 斜坡完成 当斜坡结束时置位。如果 AS_EN = 1（自动停止），则在 PWM 通道关闭时置位。 如果 AS_EN=0，则在 PWM 达到结束值时置位。 如果 IAS_EN=1（输入自动停止），则在 PWM 输出停止后置位。 如果手动激活停止斜坡，则 RC 位将在斜坡完成且输出关闭后置位。</p> <p><b>备注</b></p> <p>如果正在进行导通斜坡并且发生停止条件，则 RC 位不会为导通斜坡置位，而只会在输出因停止条件而停止时置位</p> <p>0h = PWM 通道斜坡未完成 1h = PWM 通道斜坡完成</p>

### 7.6.8.5 PWM0\_IR 寄存器 ( 偏移 = 4003h ) [复位 = 00h]

图 7-124 显示了 PWM0\_IR，表 7-165 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-124. PWM0\_IR 寄存器**

7	6	5	4	3	2	1	0
RSVD				PULSE_OVF		IAS	RC
R-0h				R/W1C-0h		R/W1C-0h	R/W1C-0h

**表 7-165. PWM0\_IR 寄存器字段说明**

位	字段	类型	复位	说明
7-3	RSVD	R	0h	保留
2	PULSE_OVF	R/W1C	0h	当前脉冲溢出 当 CUR_PULSE 计数器溢出时置位 0h = 当前脉冲计数器未溢出 1h = 当前脉冲计数器已溢出
1	IAS	R/W1C	0h	输入自动停止中断 当输入自动停止条件发生时置位；若 PWM 配置为立即停止，则输出也将停止（由 RC 位反映）。 如果 PWM 配置为在 IAS 触发时开始停止斜坡，则此中断在输入条件发生时立即置位，且输出关闭时 RC 位置位。 0h = 未触发 IAS 的输入 1h = 触发了 IAS 的输入
0	RC	R/W1C	0h	PWM 斜坡完成 当斜坡结束时置位。如果 AS_EN = 1（自动停止），则在 PWM 通道关闭时置位。 如果 AS_EN=0，则在 PWM 达到结束值时置位。 如果 IAS_EN=1（输入自动停止），则在 PWM 输出停止后置位。 如果手动激活停止斜坡，则 RC 位将在斜坡完成且输出关闭后置位。  <div style="text-align: center;"><b>备注</b></div> <div style="text-align: center;">如果正在进行导通斜坡并且发生停止条件，则 RC 位不会为导通斜坡置位，而只会在输出因停止条件而停止时置位</div> 0h = PWM 通道斜坡未完成 1h = PWM 通道斜坡完成

**7.6.8.6 PWM0\_STATUS 寄存器 ( 偏移 = 4004h ) [复位 = 00h]**

图 7-125 显示了 PWM0\_STATUS , 表 7-166 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-125. PWM0\_STATUS 寄存器**

7	6	5	4	3	2	1	0
RSVD				IAS_ACT		BUSY	OEN
R-0h				RH-0h		RH-0h	RH-0h

**表 7-166. PWM0\_STATUS 寄存器字段说明**

位	字段	类型	复位	说明
7-3	RSVD	R	0h	
2	IAS_ACT	RH	0h	输入自动停止激活 当 GPIO 输入切换至有效状态时, 该位被置位, 并在 PWM 输出停止时自动清零。 斜坡完成后应设置 RC 中断位 0h = 输入自动停止未激活 1h = 输入自动停止激活
1	BUSY	RH	0h	斜坡 IP 繁忙 0h = PWM 斜坡完成 1h = PWM 斜坡当前正在进行
0	OEN	RH	0h	输出启用 0h = PWM 输出禁用 1h = PWM 输出已使能且正在生成输出信号

### 7.6.8.7 PWM0\_CUR\_PULSE[y] 寄存器 ( 偏移 = 4005h + 公式 ) [复位 = 00000000h]

图 7-126 展示了 PWM0\_CUR\_PULSE[y]，表 7-167 中对此进行了介绍。

返回到[汇总表](#)。

当前脉冲计数的无符号 32 位计数器值。最低地址对应于计数器的 MSB。

偏移 = 4005h + (y \* 4h)；其中 y = 0h 至 3h

图 7-126. PWM0\_CUR\_PULSE[y] 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM0_CUR_PULSE																															
R-0h																															

表 7-167. PWM0\_CUR\_PULSE[y] 寄存器字段说明

位	字段	类型	复位	说明
31-0	PWM0_CUR_PULSE	R	0h	当前斜坡的当前脉冲计数。 如果斜坡已完成，则此值为上一次斜坡生成的总脉冲数。 每次启动新斜坡时都会重置此值。 最低地址对应 MSB。



#### 7.6.8.8 PWM0\_CUR\_VAL\_MSB 寄存器 ( 偏移 = 4009h ) [复位 = 00h]

图 7-127 显示了 PWM0\_CUR\_VAL\_MSB，表 7-168 中对此进行了介绍。

返回到[汇总表](#)。

图 7-127. PWM0\_CUR\_VAL\_MSB 寄存器

7	6	5	4	3	2	1	0
RSVD					CUR_VAL[10:8]		
R-0h					R-0h		

表 7-168. PWM0\_CUR\_VAL\_MSB 寄存器字段说明

位	字段	类型	复位	说明
7-3	RSVD	R	0h	
2-0	CUR_VAL[10:8]	R	0h	

**7.6.8.9 PWM0\_CUR\_VAL\_LSB 寄存器 ( 偏移 = 400Ah ) [复位 = 00h]**

图 7-128 显示了 PWM0\_CUR\_VAL\_LSB，表 7-169 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-128. PWM0\_CUR\_VAL\_LSB 寄存器**

7	6	5	4	3	2	1	0
CUR_VAL[7:0]							
R-0h							

**表 7-169. PWM0\_CUR\_VAL\_LSB 寄存器字段说明**

位	字段	类型	复位	说明
7-0	CUR_VAL[7:0]	R	0h	

#### 7.6.8.10 PWM0\_CONST\_MSB 寄存器 ( 偏移 = 400Bh ) [复位 = 00h]

图 7-129 显示了 PWM0\_CONST\_MSB , 表 7-170 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为频率变化/斜坡, 则此寄存器用于静态占空比 MSB ( 如果使能 10 位占空比分辨率 )。如果配置为占空比变化/斜坡, 则此寄存器用于开关频率的常量整数分频值 MSB。如果配置为静态 PWM 输出, 则此寄存器用于占空比 MSB ( 如果使能 10 位占空比分辨率 )。

图 7-129. PWM0\_CONST\_MSB 寄存器

7	6	5	4	3	2	1	0
RSVD					CV_F[10]	CV[9:8]	
R-0h					R/W-0h		R/W-0h

表 7-170. PWM0\_CONST\_MSB 寄存器字段说明

位	字段	类型	复位	说明
7-3	RSVD	R	0h	
2	CV_F[10]	R/W	0h	常数值 ( 仅频率 ) 这是一个多用途寄存器。 如果 PWM0_CTRL.MODE = 占空比, 则这是开关频率分频值 CONST_FREQ[10] 的整数值。 如果 PWM0_CTRL.MODE = 频率或静态, 则不使用该位
1-0	CV[9:8]	R/W	0h	常量值 这是一个多用途寄存器 如果 PWM0_CTRL.MODE = 占空比, 则这是开关频率分频值 CONST_FREQ[9:8] 的整数值, 如果 PWM0_CTRL.MODE = ( 频率或静态 ) 且 PWM0_CTRL.MODE = 8B = 0 ( 10 位 ), 则这是占空比值 CONST_DC[9:8] ( CONST_DC/1024 = 占空比 )。 在 8 位分辨率模式下, 此位未使用。

#### 7.6.8.11 PWM0\_CONST\_LSB 寄存器 ( 偏移 = 400Ch ) [复位 = 00h]

图 7-130 显示了 PWM0\_CONST\_LSB，表 7-171 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为频率变化/斜坡，则此寄存器用于静态占空比 LSB。如果配置为占空比变化/斜坡，则此寄存器用于开关频率的常量整数分频值 LSB。如果配置为静态 PWM 输出，则此寄存器用于占空比 LSB。

图 7-130. PWM0\_CONST\_LSB 寄存器

7	6	5	4	3	2	1	0
CV[7:0]							
R/W-0h							

表 7-171. PWM0\_CONST\_LSB 寄存器字段说明

位	字段	类型	复位	说明
7-0	CV[7:0]	R/W	0h	常量值 这是一个多功能寄存器 如果 PWM 0_CTRL.MODE = 占空比，则这是开关频率分频值 CONST_FREQ[7:0] 的整数值。 如果 PWM 0_CTRL.MODE = 频率或静态，则这是占空比分频值 CONST_DC[7:0]。 如果 是 8 位模式，则 CONST_DC/ 256 = 占空比。 如果 是 10 位模式，则 CONST_DC/ 1024 = 占空比

**7.6.8.12 PWM0\_STOP\_VAL\_FRAC\_F 寄存器 ( 偏移 = 400Dh ) [复位 = 00h]**

图 7-131 显示了 PWM0\_STOP\_VAL\_FRAC\_F，表 7-172 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为频率变化/斜坡，则此寄存器用于停止/关断目标时频率的小数分频值。如果配置为占空比斜坡或用于静态 PWM，则不使用此位。

**图 7-131. PWM0\_STOP\_VAL\_FRAC\_F 寄存器**

7	6	5	4	3	2	1	0
RSVD	SPV_FR_F						
R-0h	R/W-0h						

**表 7-172. PWM0\_STOP\_VAL\_FRAC\_F 寄存器字段说明**

位	字段	类型	复位	说明
7	RSVD	R	0h	
6-0	SPV_FR_F	R/W	0h	停止值小数值 ( 仅频率模式 ) 仅当 PWM0_CTRL.MODE = 频率模式时使用。 解释为频率分频值的 x/128 小数部分。 若 PWM0_CTRL.MODE = 占空比模式或静态模式，则不使用该位。

### 7.6.8.13 PWM0\_STOP\_VAL\_MSB 寄存器 ( 偏移 = 400Eh ) [复位 = 00h]

图 7-132 显示了 PWM0\_STOP\_VAL\_MSB，表 7-173 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为占空比变化/斜坡，则此寄存器用于占空比 MSB ( 仅在使能 10 位占空比分辨率时使用 )。如果配置为频率变化/斜坡，则此寄存器用于停止/关断目标时整数分频值的 MSB。如果配置为静态 PWM，则不使用该位

**图 7-132. PWM0\_STOP\_VAL\_MSB 寄存器**

7	6	5	4	3	2	1	0
RSVD					SPV_F[10]	SPV[9:8]	
R-0h					R/W-0h		R/W-0h

**表 7-173. PWM0\_STOP\_VAL\_MSB 寄存器字段说明**

位	字段	类型	复位	说明
7-3	RSVD	R	0h	
2	SPV_F[10]	R/W	0h	停止值 ( 仅频率 ) 这是一个多功能寄存器。 如果 PWM0_CTRL.MODE = 频率，则这是开关频率分频值 STOP_VAL[10] 的整数。 如果 PWM0_CTRL.MODE = 占空比，则不使用该位。 如果 PWM0_CTRL.MODE = 静态，则不使用该位。
1-0	SPV[9:8]	R/W	0h	停止值 这是一个多用途寄存器。 如果 PWM0_CTRL.MODE = 占空比且 PWM0_CTRL.MODE = 频率，则这是开关频率分频值 STOP_VAL[9:8] 的整数。 如果 PWM0_CTRL.MODE = 静态，则不使用该位。

#### 7.6.8.14 PWM0\_STOP\_VAL\_LSB 寄存器 ( 偏移 = 400Fh ) [复位 = 00h]

图 7-133 显示了 PWM0\_STOP\_VAL\_LSB，表 7-174 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为占空比变化/斜坡，则此寄存器用于占空比 LSB。如果配置为频率变化/斜坡，则此寄存器用于停止/关断目标时整数分频值的 LSB。如果配置为静态 PWM，则不使用该位

图 7-133. PWM0\_STOP\_VAL\_LSB 寄存器

7	6	5	4	3	2	1	0
SPV[7:0]							
R/W-0h							

表 7-174. PWM0\_STOP\_VAL\_LSB 寄存器字段说明

位	字段	类型	复位	说明
7-0	SPV[7:0]	R/W	0h	停止值 这是一个多用途寄存器。 PWM 0_CTRL.MODE = 占空比，则这是占空比值 SPV[7:0]。 如果 是 8 位模式，则 SPV/256 = 占空比。 如果 是 10 位模式，则 SPV/1024 = 占空比。 如果 PWM 0_CTRL.MODE = 频率，则这是开关频率分频值 STOP_VAL[7:0] 的整数值。 如果 PWM 0_CTRL.MODE = 静态，则不使用该位。

#### 7.6.8.15 PWM0\_STOP\_SL\_MSB 寄存器 ( 偏移 = 4010h ) [复位 = 00h]

图 7-134 显示了 PWM0\_STOP\_SL\_MSB，表 7-175 中对此进行了介绍。

返回到[汇总表](#)。

用于配置停止斜坡的斜率标度

图 7-134. PWM0\_STOP\_SL\_MSB 寄存器

7	6	5	4	3	2	1	0
RSVD			SP_SL[20:16]				
R-0h			R/W-0h				

表 7-175. PWM0\_STOP\_SL\_MSB 寄存器字段说明

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4-0	SP_SL[20:16]	R/W	0h	停止斜率 停止斜坡的斜率。 这些字段会根据 PWM0_CTRL.SLOPE_SCALE 的值而变化。 有关更多信息，请参阅 <a href="#">节 7.5.7.4</a> 如果 PWM0_CTRL.MODE = 静态，则不使用该位。



**7.6.8.16 PWM0\_STOP\_SL\_MID 寄存器 ( 偏移 = 4011h ) [复位 = 00h]**

图 7-135 显示了 PWM0\_STOP\_SL\_MID , 表 7-176 中对此进行了介绍。

返回到[汇总表](#)。

用于配置停止斜坡的斜率标度

**图 7-135. PWM0\_STOP\_SL\_MID 寄存器**

7	6	5	4	3	2	1	0
SP_SL[15:8]							
R/W-0h							

**表 7-176. PWM0\_STOP\_SL\_MID 寄存器字段说明**

位	字段	类型	复位	说明
7-0	SP_SL[15:8]	R/W	0h	停止斜率 停止斜坡的斜率。 这些字段会根据 PWM0_CTRL.SLOPE_SCALE 的值而变化。 有关更多信息, 请参阅节 7.5.7.4 如果 PWM0_CTRL.MODE = 静态, 则不使用该位。

#### 7.6.8.17 PWM0\_STOP\_SL\_LSB 寄存器 ( 偏移 = 4012h ) [复位 = 00h]

图 7-136 显示了 PWM0\_STOP\_SL\_LSB，表 7-177 中对此进行了介绍。

返回到[汇总表](#)。

用于配置停止斜坡的斜率标度

图 7-136. PWM0\_STOP\_SL\_LSB 寄存器

7	6	5	4	3	2	1	0
SP_SL[7:0]							
R/W-0h							

表 7-177. PWM0\_STOP\_SL\_LSB 寄存器字段说明

位	字段	类型	复位	说明
7-0	SP_SL[7:0]	R/W	0h	停止斜率 停止斜坡的斜率。 这些字段会根据 PWM0_CTRL.SLOPE_SCALE 的值而变化。 有关更多信息，请参阅节 7.5.7.4 如果 PWM0_CTRL.MODE = 静态，则不使用该位。

**7.6.8.18 PWM0\_START\_VAL\_FRAC\_F 寄存器 ( 偏移 = 4013h ) [复位 = 00h]**

图 7-137 显示了 PWM0\_START\_VAL\_FRAC\_F，表 7-178 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为频率变化/斜坡，则此寄存器用于启动时频率的小数分频值。如果配置为占空比斜坡或用于静态 PWM，则不使用此位。

**图 7-137. PWM0\_START\_VAL\_FRAC\_F 寄存器**

7	6	5	4	3	2	1	0
RSVD	SPV_FR_F						
R-0h	R/W-0h						

**表 7-178. PWM0\_START\_VAL\_FRAC\_F 寄存器字段说明**

位	字段	类型	复位	说明
7	RSVD	R	0h	
6-0	SPV_FR_F	R/W	0h	起始值小数部分 ( 仅频率模式 ) 仅当 PWM0_CTRL.MODE = 频率模式时使用。 解释为频率分频值的 x/128 小数部分。 若 PWM0_CTRL.MODE = 占空比模式或静态模式，则不使用该位。

### 7.6.8.19 PWM0\_START\_VAL\_MSB 寄存器 ( 偏移 = 4014h ) [复位 = 00h]

图 7-138 显示了 PWM0\_START\_VAL\_MSB，表 7-179 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为占空比变化/斜坡，则此寄存器用于占空比 MSB ( 仅在使能 10 位占空比分辨率时使用 )。如果配置为频率变化/斜坡，则此寄存器用于启动时频率的整数分频值 MSB。如果配置为静态 PWM，则不使用该位

**图 7-138. PWM0\_START\_VAL\_MSB 寄存器**

7	6	5	4	3	2	1	0
RSVD					STV_F[10]	STV[9:8]	
R-0h					R/W-0h		R/W-0h

**表 7-179. PWM0\_START\_VAL\_MSB 寄存器字段说明**

位	字段	类型	复位	说明
7-3	RSVD	R	0h	
2	STV_F[10]	R/W	0h	起始值 ( 仅频率 ) 这是一个多用途寄存器。 如果 PWM0_CTRL.MODE = 频率，则这是开关频率分频值 START_VAL[10] 的整数值。 如果 PWM0_CTRL.MODE = 占空比，则不使用该位。 如果 PWM0_CTRL.MODE = 静态，则不使用该位。
1-0	STV[9:8]	R/W	0h	起始值 这是一个多用途寄存器。 如果 PWM0_CTRL.MODE = 占空比且 PWM0_CTRL.MODE = 频率，则这是开关频率分频值 START_VAL[9:8] 的整数值。 如果 PWM0_CTRL.MODE = 静态，则不使用该位。

**7.6.8.20 PWM0\_START\_VAL\_LSB 寄存器 ( 偏移 = 4015h ) [复位 = 00h]**

图 7-139 显示了 PWM0\_START\_VAL\_LSB , 表 7-180 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为占空比变化/斜坡, 则此寄存器用于占空比 LSB。如果配置为频率变化/斜坡, 则此寄存器用于启动时频率的整数分频值 LSB。如果配置为静态 PWM, 则不使用该位

**图 7-139. PWM0\_START\_VAL\_LSB 寄存器**

7	6	5	4	3	2	1	0
STV[7:0]							
R/W-0h							

**表 7-180. PWM0\_START\_VAL\_LSB 寄存器字段说明**

位	字段	类型	复位	说明
7-0	STV[7:0]	R/W	0h	起始值 这是一个多用途寄存器。 如果 PWM 0_CTRL.MODE = 占空比, 则这是占空比值 STV[ 7: 0]。 如果 是 8 位模式, 则 STV/ 256 = 占空比。 如果 是 10 位模式, 则 STV/ 1024 = 占空比。 如果 PWM 0_CTRL.MODE = 频率, 则这是开关频率分频值 START_VAL[ 7: 0] 的整数值。 如果 PWM 0_CTRL.MODE = 静态, 则不使用该位。

**7.6.8.21 PWM0\_START\_SL\_MSB 寄存器 ( 偏移 = 4016h ) [复位 = 00h]**

图 7-140 显示了 PWM0\_START\_SL\_MSB，表 7-181 中对此进行了介绍。

返回到[汇总表](#)。

用于配置起始斜坡的斜率标度

**图 7-140. PWM0\_START\_SL\_MSB 寄存器**

7	6	5	4	3	2	1	0
RSVD			ST_SL[20:16]				
R-0h			R/W-0h				

**表 7-181. PWM0\_START\_SL\_MSB 寄存器字段说明**

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4-0	ST_SL[20:16]	R/W	0h	起始斜率 起始斜坡的斜率。 这些字段会根据 PWM0_CTRL.SLOPE_SCALE 的值而变化。 有关更多信息，请参阅 <a href="#">节 7.5.7.4</a> 如果 PWM0_CTRL.MODE = 静态，则不使用该位。

**7.6.8.22 PWM0\_START\_SL\_MID 寄存器 ( 偏移 = 4017h ) [复位 = 00h]**

图 7-141 显示了 PWM0\_START\_SL\_MID，表 7-182 中对此进行了介绍。

返回到[汇总表](#)。

用于配置起始斜坡的斜率标度

**图 7-141. PWM0\_START\_SL\_MID 寄存器**

7	6	5	4	3	2	1	0
ST_SL[15:8]							
R/W-0h							

**表 7-182. PWM0\_START\_SL\_MID 寄存器字段说明**

位	字段	类型	复位	说明
7-0	ST_SL[15:8]	R/W	0h	起始斜率 起始斜坡的斜率。 这些字段会根据 PWM0_CTRL.SLOPE_SCALE 的值而变化。 有关更多信息，请参阅节 7.5.7.4 如果 PWM0_CTRL.MODE = 静态，则不使用该位。

**7.6.8.23 PWM0\_START\_SL\_LSB 寄存器 ( 偏移 = 4018h ) [复位 = 00h]**

图 7-142 显示了 PWM0\_START\_SL\_LSB , 表 7-183 中对此进行了介绍。

返回到[汇总表](#)。

用于配置起始斜坡的斜率标度

**图 7-142. PWM0\_START\_SL\_LSB 寄存器**

7	6	5	4	3	2	1	0
ST_SL[7:0]							
R/W-0h							

**表 7-183. PWM0\_START\_SL\_LSB 寄存器字段说明**

位	字段	类型	复位	说明
7-0	ST_SL[7:0]	R/W	0h	起始斜率 起始斜坡的斜率。 这些字段会根据 PWM0_CTRL.SLOPE_SCALE 的值而变化。 有关更多信息, 请参阅节 <a href="#">7.5.7.4</a> 如果 PWM0_CTRL.MODE = 静态, 则不使用该位。



**7.6.8.24 PWM0\_END\_VAL\_CONST\_FRAC\_F 寄存器 ( 偏移 = 4019h ) [复位 = 00h]**

图 7-143 显示了 PWM0\_END\_VAL\_CONST\_FRAC\_F，表 7-184 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为频率变化，则这是频率的结束值 ( 目标值 ) 小数分频值。如果配置为占空比变化/斜坡，则这是开关频率的小数分频值。如果配置为静态 PWM 输出，则此寄存器用于频率分频值的小数部分。

**图 7-143. PWM0\_END\_VAL\_CONST\_FRAC\_F 寄存器**

7	6	5	4	3	2	1	0
RSVD	EVCV_FR_F						
R-0h	R/W-0h						

**表 7-184. PWM0\_END\_VAL\_CONST\_FRAC\_F 寄存器字段说明**

位	字段	类型	复位	说明
7	RSVD	R	0h	
6-0	EVCV_FR_F	R/W	0h	结束值小数 ( 频率模式 ) 或常数小数 ( 直流模式 ) 结束值小数部分。 此寄存器始终使用，但该字段会根据当前使用的模式而变化。 解释为频率分频值的 x/128 小数部分。 如果 PWM0_CTRL.MODE = 占空比，则这是开关频率分频值 CONST_FREQ[10] 的整数值。 如果 PWM0_CTRL.MODE = 频率或静态，则不使用该位

#### 7.6.8.25 PWM0\_END\_VAL\_MSB 寄存器 ( 偏移 = 401Ah ) [复位 = 00h]

图 7-144 显示了 PWM0\_END\_VAL\_MSB，表 7-185 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为斜坡，则可设置变化的 PWM 结束值 ( 目标值 )。如果配置为静态 PWM 输出，则此寄存器用于频率分频值的整数 MSB 部分。

图 7-144. PWM0\_END\_VAL\_MSB 寄存器

7	6	5	4	3	2	1	0
RSVD					EV_FR[10]	END_VAL[9:8]	
R-0h					R/W-0h		R/W-0h

表 7-185. PWM0\_END\_VAL\_MSB 寄存器字段说明

位	字段	类型	复位	说明
7-3	RSVD	R	0h	
2	EV_FR[10]	R/W	0h	结束值 ( 仅频率 ) 这是一个多用途寄存器。 如果 PWM0_CTRL.MODE = 占空比或静态，则这是开关频率值 START_VAL[10] 的整数值。 如果 PWM0_CTRL.MODE = 频率，则不使用该位
1-0	END_VAL[9:8]	R/W	0h	结束值 这是一个多用途寄存器。 如果 PWM0_CTRL.MODE = 占空比或静态，则这是开关频率分频值 END_VAL[9:8] 的整数值。 如果 PWM0_CTRL.MODE = 频率且 PWM0_CTRL.8B = 0 ( 10 位 )，则这是占空比值 END_VAL[9:8]，否则不使用该位。 如果 是 10 位模式，则 END_VAL/1024 = 占空比，但在 8 位模式下不使用此位

**7.6.8.26 PWM0\_END\_VAL\_LSB 寄存器 ( 偏移 = 401Bh ) [复位 = 00h]**

图 7-145 显示了 PWM0\_END\_VAL\_LSB，表 7-186 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为斜坡，则可设置变化的 PWM 结束值 ( 目标值 )。如果配置为静态 PWM 输出，则此寄存器用于频率分频值的整数 LSB 部分。

**图 7-145. PWM0\_END\_VAL\_LSB 寄存器**

7	6	5	4	3	2	1	0
END_VAL[7:0]							
R/W-0h							

**表 7-186. PWM0\_END\_VAL\_LSB 寄存器字段说明**

位	字段	类型	复位	说明
7-0	END_VAL[7:0]	R/W	0h	结束值 这是一个多用途寄存器。 如果 PWM 0_CTRL.MODE = 占空比或静态，则这是开关频率分频值 END_VAL[7:0] 的整数值。 如果 PWM 0_CTRL.MODE = 频率，则这是占空比值 END_VAL[7:0]。 如果 是 8 位模式，则 END_VAL/ 256 = 占空比。 如果 是 10 位模式，则 END_VAL/ 1024 = 占空比

7.6.8.27 PWM0\_PULSE\_STOP\_RAMP[y] 寄存器 ( 偏移 = 401Ch + 公式 ) [复位 = 00000000h]

图 7-146 展示了 PWM0\_PULSE\_STOP\_RAMP[y]，表 7-187 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为自动停止斜坡模式，这是从起始脉冲开始，模块将自动设置停止斜坡命令时已生成的 PWM 脉冲数。最低地址对应于计数器的 MSB。

偏移 = 401Ch + (y \* 4h)；其中 y = 0h 至 3h

图 7-146. PWM0\_PULSE\_STOP\_RAMP[y] 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM0_PULSE_STOP_RAMP																															
R/W-0h																															

表 7-187. PWM0\_PULSE\_STOP\_RAMP[y] 寄存器字段说明

位	字段	类型	复位	说明
31-0	PWM0_PULSE_STOP_RAMP	R/W	0h	一旦生成指定数量的 PWM 脉冲，停止斜坡将自动启动。仅当 PWM0_ACTION.AUTO_STOP 位置位时使用。最低地址对应 MSB。

**7.6.8.28 PWM0\_PULSE\_MAX[y] 寄存器 ( 偏移 = 4020h + 公式 ) [复位 = 00000000h]**

图 7-147 展示了 PWM0\_PULSE\_MAX[y]，表 7-188 中对此进行了介绍。

返回到[汇总表](#)。

若配置为自动停止斜坡模式，该值表示允许的最大 PWM 脉冲数量。当 PWM 脉冲数达到该限制值时，PWM 输出将立即停止。最低地址对应于计数器的 MSB。

偏移 = 4020h + (y \* 4h)；其中 y = 0h 至 3h

**图 7-147. PWM0\_PULSE\_MAX[y] 寄存器**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM0_PULSE_MAX																															
R/W-0h																															

**表 7-188. PWM0\_PULSE\_MAX[y] 寄存器字段说明**

位	字段	类型	复位	说明
31-0	PWM0_PULSE_MAX	R/W	0h	一旦生成指定数量的 PWM 脉冲，PWM 输出将被禁用。 仅当 PWM0_ACTION.AUTO_STOP 位置位时使用。 最低地址对应 MSB。

### 7.6.8.29 PWM0\_ACTION 寄存器 ( 偏移 = 4024h ) [复位 = 00h]

图 7-148 显示了 PWM0\_ACTION，表 7-189 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-148. PWM0\_ACTION 寄存器**

7	6	5	4	3	2	1	0
RSVD			COUNT_RST	AUTO_STOP	UDS	启动	STOP
R-0h			R/W1S-0h	R/W-0h	R/W-0h	RH/W1S-0h	RH/W1S-0h

**表 7-189. PWM0\_ACTION 寄存器字段说明**

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4	COUNT_RST	R/W1S	0h	<p>将脉冲计数器复位为 0 如果此位置位，脉冲计数器将被复位。 这通常在斜坡开始时执行，以跟踪脉冲总数。 复位完成后，硬件将清除此位。</p> <hr/> <p style="text-align: center;"><b>备注</b></p> <p>如果使用 AUTO_STOP，请注意停止和最大脉冲计数值。 这些脉冲计数器将与当前脉冲计数器进行比较，若最大脉冲计数小于当前脉冲计数器值，则不会生成 PWM 输出</p>
3	AUTO_STOP	R/W	0h	<p>使用自动停止斜坡 如果使能，一旦脉冲计数达到指定值 0h = 不使用自动停止斜坡行为 1h = 根据脉冲计数寄存器自动开始停止斜坡</p>
2	UDS	R/W	0h	<p>使用定义的起始点 使用在起始值寄存器中定义的起始点。 用于确保起始点已知，或若希望斜坡从当前值（斜坡中途或现有终点）开始 0h = 使用现有的当前值作为起始点 1h = 从定义的起始点开始</p>
1	启动	RH/W1S	0h	<p>启动 根据配置的设置启动定义的斜坡曲线。 如果同时设置 STOP，则 START 将被忽略。 该位由硬件清除。</p>
0	STOP	RH/W1S	0h	<p>停止 将斜坡运行到指定的停止点，然后关闭。 如果同时设置 START 和 STOP，则会执行 STOP。 该位由硬件清除</p>

**7.6.8.30 PWM0\_IAS\_CTRL 寄存器 (偏移 = 4030h) [复位 = 00h]**

图 7-149 显示了 PWM0\_IAS\_CTRL，表 7-190 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-149. PWM0\_IAS\_CTRL 寄存器**

7	6	5	4	3	2	1	0
RSVD			IN_POL	GPIO_SEL		STOP_MODE	AS_EN
R-0h			R/W-0h	R/W-0h		R/W-0h	R/W-0h

**表 7-190. PWM0\_IAS\_CTRL 寄存器字段说明**

位	字段	类型	复位	说明
7-5	RSVD	R	0h	保留
4	IN_POL	R/W	0h	输入极性 选择用于触发停止行为的电平 0h = 低电平有效输入信号 1h = 高电平有效输入信号
3-2	GPIO_SEL	R/W	0h	GPIO 选择 指定使用哪个 GPIO。 GPIO 必须配置为 GPIO 功能，而不是特殊功能，才能正常工作。 0h = GPIO0 1h = GPIO1 2h = GPIO7 3h = GPIO8
1	STOP_MODE	R/W	0h	停止模式 当发生 GPIO 触发的停止事件时，选择立即禁用 PWM 输出还是请求停止斜坡。 0h = GPIO 触发事件请求停止斜坡 1h = GPIO 触发事件立即禁用 PWM 输出
0	AS_EN	R/W	0h	输入自动停止使能  <div>备注</div> <div>这不要求设置 AUTO_STOP 位，它是一个基于 GPIO 输入停止输出的独立功能</div> 0h = 未通过 GPIO 信号启用自动停止功能 1h = 启用由 GPIO 触发的自动停止功能

**7.6.9 PWM1 寄存器**

节 7.6.9 列出了 PWM1 寄存器的存储器映射寄存器。节 7.6.9 中未列出的所有寄存器偏移地址都应视为保留的位置，并且不应修改寄存器内容。

**表 7-191. PWM1 寄存器**

偏移	首字母缩写词	寄存器名称	部分
4100h	PWM1_CTRL	PWM1 控制	<a href="#">节 7.6.9.1</a>
4101h	PWM1_IE0	中断启用	<a href="#">节 7.6.9.2</a>
4102h	PWM1_IE1	中断启用	<a href="#">节 7.6.9.3</a>
4103h	PWM1_IR	中断寄存器	<a href="#">节 7.6.9.4</a>
4104h	PWM1_STATUS	状态	<a href="#">节 7.6.9.5</a>
4105h + 公式	PWM1_CUR_PULSE[y]	当前脉冲计数	<a href="#">节 7.6.9.6</a>
4109h	PWM1_CUR_VAL_MSB	当前值 MSB	<a href="#">节 7.6.9.7</a>
410Ah	PWM1_CUR_VAL_LSB	当前值 LSB	<a href="#">节 7.6.9.8</a>
410Bh	PWM1_CONST_MSB	常量值整数 MSB	<a href="#">节 7.6.9.9</a>

**表 7-191. PWM1 寄存器 (续)**

偏移	首字母缩写词	寄存器名称	部分
410Ch	PWM1_CONST_LSB	常量值整数 LSB	节 7.6.9.10
410Dh	PWM1_STOP_VAL_FRAC_F	停止小数值 (仅频率)	节 7.6.9.11
410Eh	PWM1_STOP_VAL_MSB	停止值整数 MSB	节 7.6.9.12
410Fh	PWM1_STOP_VAL_LSB	停止值整数 LSB	节 7.6.9.13
4110h	PWM1_STOP_SL_MSB	停止斜率 MSB	节 7.6.9.14
4111h	PWM1_STOP_SL_MID	停止斜率中间位字节	节 7.6.9.15
4112h	PWM1_STOP_SL_LSB	停止斜率 LSB	节 7.6.9.16
4113h	PWM1_START_VAL_FRAC_F	起始小数值 (仅频率)	节 7.6.9.17
4114h	PWM1_START_VAL_MSB	起始值 MSB	节 7.6.9.18
4115h	PWM1_START_VAL_LSB	起始值 LSB	节 7.6.9.19
4116h	PWM1_START_SL_MSB	起始斜率 MSB	节 7.6.9.20
4117h	PWM1_START_SL_MID	起始斜率中间字节	节 7.6.9.21
4118h	PWM1_START_SL_LSB	起始斜率 LSB	节 7.6.9.22
4119h	PWM1_END_VAL_CONST_FRAC_F	结束 (频率) 或恒定 (占空比) 小数值	节 7.6.9.23
411Ah	PWM1_END_VAL_MSB	结束值 MSB	节 7.6.9.24
411Bh	PWM1_END_VAL_LSB	结束值 LSB	节 7.6.9.25
411Ch + 公式	PWM1_PULSE_STOP_RAMP[y]	开始停止斜坡的脉冲计数	节 7.6.9.26
4120h + 公式	PWM1_PULSE_MAX[y]	脉冲计数最大值	节 7.6.9.27
4124h	PWM1_ACTION	起始动作寄存器。	节 7.6.9.28
4130h	PWM1_IAS_CTRL	输入自动停止控制	节 7.6.9.29

复杂的位访问类型经过编码可适应小型表单元。节 7.6.9 展示了适用于此部分中访问类型的代码。

**表 7-192. PWM1 访问类型代码**

访问类型	代码	说明
读取类型		
R	R	读取
RH	R H	读取 由硬件置位或清零
写入类型		
W	W	写入
W1C	W 1C	写入 1 以清零
W1S	W 1S	写入 1 以进行设置
WP	W P	写入 需要特权访问
复位或默认值		
-n		复位后的值或默认值
寄存器数组变量		
i、j、k、l、m、n		当这些变量用于寄存器名称、偏移或地址时，它们指的是寄存器数组的值，其中寄存器是一组重复寄存器的一部分。寄存器组构成分层结构，数组用公式表示。
y		当该变量用于寄存器名称、偏移或地址时，它指的是寄存器数组的值。



### 7.6.9.1 PWM1\_CTRL 寄存器 ( 偏移 = 4100h ) [复位 = 11h]

图 7-150 显示了 PWM1\_CTRL , 表 7-193 中对此进行了介绍。

返回到[汇总表](#)。

用于配置 PWM1 模块的控制字段

图 7-150. PWM1\_CTRL 寄存器

7	6	5	4	3	2	1	0
RSVD	SLOPE_SCALE			模式		DC_8B	INIT
R-0h	R/WP-1h			R/WP-0h		R/WP-0h	R/W-1h

表 7-193. PWM1\_CTRL 寄存器字段说明

位	字段	类型	复位	说明
7	RSVD	R	0h	
6-4	SLOPE_SCALE	R/WP	1h	斜率标度 调整用于斜率寄存器的整数与小数部分的位数。 枚举值的含义会根据正在进行斜坡变化的是频率还是占空比而有所不同。 有关更多信息,请参阅节 7.5.7.4 0h = 整数 [20-16], 小数 [15-0] ( 频率 ) - 整数 [20], 小数 [19-0] ( 占空比 ) 1h = 整数 [20-12], 小数 [11-0] ( 频率 ) - 整数 [20-19], 小数 [18-0] ( 占空比 ) 2h = 整数 [20-8], 小数 [7-0] ( 频率 ) - 整数 [20-18], 小数 [17-0] ( 占空比 ) 3h = 整数 [20-4], 小数 [3-0] ( 频率 ) - 整数 [20-17], 小数 [16-0] ( 占空比 ) 4h = 整数 [20-0], 无小数 ( 频率 ) - 整数 [20-16], 小数 [15-0] ( 占空比 )
3-2	模式	R/WP	0h	调整用于斜率寄存器整数部分与小数部分的位数。 请注意, 如果 GPIO 配置为特殊功能, 那么将模式设置为非关闭状态将优先于 SPI 片选。 0h = 关闭 1h = 占空比斜坡控制 2h = 频率斜坡控制 3h = 静态开启
1	DC_8B	R/WP	0h	8 位占空比模式 0h = 10 位占空比分辨率 1h = 8 位占空比分辨率
0	INIT	R/W	1h	初始化模式 将 PWM IP 保持在复位状态, 用于配置此 PWM1_CTRL 寄存器中的设置 0h = 禁用初始化模式。不允许写入受保护的 PWM 位 1h = 初始化模式已使能。允许对受保护的 PWM 位进行写入

### 7.6.9.2 PWM1\_IE0 寄存器 ( 偏移 = 4101h ) [复位 = 00h]

图 7-151 显示了 PWM1\_IE0 , 表 7-194 中对此进行了介绍。

返回到[汇总表](#)。

图 7-151. PWM1\_IE0 寄存器

7	6	5	4	3	2	1	0
RSVD					PULSE_OVF_I E	IAS_IE	RESERVED
R-0h					R/W-0h	R/W-0h	R-0h

表 7-194. PWM1\_IE0 寄存器字段说明

位	字段	类型	复位	说明
7-3	RSVD	R	0h	保留
2	PULSE_OVF_IE	R/W	0h	当前脉冲溢出 当 CUR_PULSE 计数器溢出时置位 0h = 当前脉冲计数器未溢出 1h = 当前脉冲计数器已溢出
1	IAS_IE	R/W	0h	输入自动停止中断 当输入自动停止条件发生时置位；若 PWM 配置为立即停止，则输出也将停止（由 RC 位反映）。 如果 PWM 配置为在 IAS 触发时开始停止斜坡，则此中断在输入条件发生时立即置位，且输出关闭时 RC 位置位。 0h = 未触发 IAS 的输入 1h = 触发了 IAS 的输入
0	RESERVED	R	0h	

### 7.6.9.3 PWM1\_IE1 寄存器 ( 偏移 = 4102h ) [复位 = 00h]

图 7-152 显示了 PWM1\_IE1 , 表 7-195 中对此进行了介绍。

返回到[汇总表](#)。

图 7-152. PWM1\_IE1 寄存器

7	6	5	4	3	2	1	0
RSVD					PULSE_OVF_I E	IAS_IE	RC_IE
R-0h					R/W-0h	R/W-0h	R/W-0h

表 7-195. PWM1\_IE1 寄存器字段说明

位	字段	类型	复位	说明
7-3	RSVD	R	0h	保留
2	PULSE_OVF_IE	R/W	0h	当前脉冲溢出 当 CUR_PULSE 计数器溢出时置位 0h = 当前脉冲计数器未溢出 1h = 当前脉冲计数器已溢出
1	IAS_IE	R/W	0h	输入自动停止中断 当输入自动停止条件发生时置位；若 PWM 配置为立即停止，则输出也将停止（由 RC 位反映）。 如果 PWM 配置为在 IAS 触发时开始停止斜坡，则此中断在输入条件发生时立即置位，且输出关闭时 RC 位置位。  <b>备注</b> 如果正在进行导通斜坡并且发生停止条件，则 RC 位不会为导通斜坡置位，而只会在输出因停止条件而停止时置位  0h = 未触发 IAS 的输入 1h = 触发了 IAS 的输入
0	RC_IE	R/W	0h	PWM 斜坡完成 当斜坡结束时置位。如果 AS_EN = 1（自动停止），则在 PWM 通道关闭时置位。 如果 AS_EN=0，则在 PWM 达到结束值时置位。 如果 IAS_EN=1（输入自动停止），则在 PWM 输出停止后置位。 如果手动激活停止斜坡，则 RC 位将在斜坡完成且输出关闭后置位。  <b>备注</b> 如果正在进行导通斜坡并且发生停止条件，则 RC 位不会为导通斜坡置位，而只会在输出因停止条件而停止时置位  0h = PWM 通道斜坡未完成 1h = PWM 通道斜坡完成

#### 7.6.9.4 PWM1\_IR 寄存器 ( 偏移 = 4103h ) [复位 = 00h]

图 7-153 显示了 PWM1\_IR，表 7-196 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-153. PWM1\_IR 寄存器**

7	6	5	4	3	2	1	0
RSVD				PULSE_OVF		IAS	RC
R-0h				R/W1C-0h		R/W1C-0h	R/W1C-0h

**表 7-196. PWM1\_IR 寄存器字段说明**

位	字段	类型	复位	说明
7-3	RSVD	R	0h	保留
2	PULSE_OVF	R/W1C	0h	当前脉冲溢出 当 CUR_PULSE 计数器溢出时置位 0h = 当前脉冲计数器未溢出 1h = 当前脉冲计数器已溢出
1	IAS	R/W1C	0h	输入自动停止中断 当输入自动停止条件发生时置位；若 PWM 配置为立即停止，则输出也将停止（由 RC 位反映）。 如果 PWM 配置为在 IAS 触发时开始停止斜坡，则此中断在输入条件发生时立即置位，且输出关闭时 RC 位置位。 0h = 未触发 IAS 的输入 1h = 触发了 IAS 的输入
0	RC	R/W1C	0h	PWM 斜坡完成 当斜坡结束时置位。如果 AS_EN = 1（自动停止），则在 PWM 通道关闭时置位。 如果 AS_EN=0，则在 PWM 达到结束值时置位。 如果 IAS_EN = 1（输入自动停止），则在 PWM 输出停止后置位。 如果手动激活停止斜坡，则在斜坡完成且输出关闭后对 RC 位置位。  <div style="text-align: center;"><b>备注</b></div> <div style="text-align: center;">如果正在进行导通斜坡并且发生停止条件，则 RC 位不会为导通斜坡置位，而只会在输出因停止条件而停止时置位</div> 0h = PWM 通道斜坡未完成 1h = PWM 通道斜坡完成

**7.6.9.5 PWM1\_STATUS 寄存器 ( 偏移 = 4104h ) [复位 = 00h]**

图 7-154 显示了 PWM1\_STATUS , 表 7-197 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-154. PWM1\_STATUS 寄存器**

7	6	5	4	3	2	1	0
RSVD				IAS_ACT		BUSY	OEN
R-0h				RH-0h		RH-0h	RH-0h

**表 7-197. PWM1\_STATUS 寄存器字段说明**

位	字段	类型	复位	说明
7-3	RSVD	R	0h	
2	IAS_ACT	RH	0h	输入自动停止激活 当 GPIO 输入切换至有效状态时, 该位被置位, 并在 PWM 输出停止时自动清零。 斜坡完成后应设置 RC 中断位 0h = 输入自动停止未激活 1h = 输入自动停止激活
1	BUSY	RH	0h	斜坡 IP 繁忙 0h = PWM 斜坡完成 1h = PWM 斜坡当前正在进行
0	OEN	RH	0h	输出启用 0h = PWM 输出禁用 1h = PWM 输出已使能且正在生成输出信号

#### 7.6.9.6 PWM1\_CUR\_PULSE[y] 寄存器 ( 偏移 = 4105h + 公式 ) [复位 = 00000000h]

图 7-155 展示了 PWM1\_CUR\_PULSE[y]，表 7-198 中对此进行了介绍。

返回到[汇总表](#)。

当前脉冲计数的无符号 32 位计数器值。最低地址对应于计数器的 MSB。

偏移 = 4105h + (y \* 4h)；其中 y = 0h 至 3h

图 7-155. PWM1\_CUR\_PULSE[y] 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM0_CUR_PULSE																															
R-0h																															

表 7-198. PWM1\_CUR\_PULSE[y] 寄存器字段说明

位	字段	类型	复位	说明
31-0	PWM0_CUR_PULSE	R	0h	当前斜坡的当前脉冲计数。 如果斜坡已完成，则此值为上一次斜坡生成的总脉冲数。 每次启动新斜坡时都会重置此值。 最低地址对应 MSB。

#### 7.6.9.7 PWM1\_CUR\_VAL\_MSB 寄存器 ( 偏移 = 4109h ) [复位 = 00h]

图 7-156 显示了 PWM1\_CUR\_VAL\_MSB，表 7-199 中对此进行了介绍。

返回到[汇总表](#)。

图 7-156. PWM1\_CUR\_VAL\_MSB 寄存器

7	6	5	4	3	2	1	0
RSVD					CUR_VAL[10:8]		
R-0h					R-0h		

表 7-199. PWM1\_CUR\_VAL\_MSB 寄存器字段说明

位	字段	类型	复位	说明
7-3	RSVD	R	0h	
2-0	CUR_VAL[10:8]	R	0h	

#### 7.6.9.8 PWM1\_CUR\_VAL\_LSB 寄存器 ( 偏移 = 410Ah ) [复位 = 00h]

图 7-157 显示了 PWM1\_CUR\_VAL\_LSB , 表 7-200 中对此进行了介绍。

返回到[汇总表](#)。

图 7-157. PWM1\_CUR\_VAL\_LSB 寄存器

7	6	5	4	3	2	1	0
CUR_VAL[7:0]							
R-0h							

表 7-200. PWM1\_CUR\_VAL\_LSB 寄存器字段说明

位	字段	类型	复位	说明
7-0	CUR_VAL[7:0]	R	0h	



**7.6.9.9 PWM1\_CONST\_MSB 寄存器 ( 偏移 = 410Bh ) [复位 = 00h]**

图 7-158 显示了 PWM1\_CONST\_MSB，表 7-201 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为频率变化/斜坡，则此寄存器用于静态占空比 MSB ( 如果使能 10 位占空比分辨率 )。如果配置为占空比变化/斜坡，则此寄存器用于开关频率的常量整数分频值 MSB。如果配置为静态 PWM 输出，则此寄存器用于占空比 MSB ( 如果使能 10 位占空比分辨率 )。

**图 7-158. PWM1\_CONST\_MSB 寄存器**

7	6	5	4	3	2	1	0
RSVD					CV_F[10]	CV[9:8]	
R-0h					R/W-0h		R/W-0h

**表 7-201. PWM1\_CONST\_MSB 寄存器字段说明**

位	字段	类型	复位	说明
7-3	RSVD	R	0h	
2	CV_F[10]	R/W	0h	常数值 ( 仅频率 ) 这是一个多用途寄存器。 如果 PWM1_CTRL.MODE = 占空比，则这是开关频率分频值 CONST_FREQ[10] 的整数值。 如果 PWM1_CTRL.MODE = 频率或静态，则不使用该位
1-0	CV[9:8]	R/W	0h	常数值 这是一个多用途寄存器。 如果 PWM1_CTRL.MODE = 占空比，则这是开关频率分频值 CONST_FREQ[9:8] 的整数值。 如果 PWM1_CTRL.MODE = ( 频率或静态 ) 且 PWM1_CTRL.MODE = 8B = 0 ( 10 位 )，则这是占空比值 CONST_DC[9:8] ( CONST_DC/1024 = 占空比 )。 在 8 位分辨率模式下，此位未使用。

7.6.9.10 PWM1\_CONST\_LSB 寄存器 ( 偏移 = 410Ch ) [复位 = 00h]

图 7-159 显示了 PWM1\_CONST\_LSB，表 7-202 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为频率变化/斜坡，则此寄存器用于静态占空比 LSB。如果配置为占空比变化/斜坡，则此寄存器用于开关频率的常量整数分频值 LSB。如果配置为静态 PWM 输出，则此寄存器用于占空比 LSB。

图 7-159. PWM1\_CONST\_LSB 寄存器

7	6	5	4	3	2	1	0
CV[7:0]							
R/W-0h							

表 7-202. PWM1\_CONST\_LSB 寄存器字段说明

位	字段	类型	复位	说明
7-0	CV[7:0]	R/W	0h	常量值 这是一个多功能寄存器 如果 PWM 1_CTRL.MODE = 占空比，则这是开关频率分频值 CONST_FREQ[7:0] 的整数值。 如果 PWM 1_CTRL.MODE = 频率或静态，则这是占空比分频值 CONST_DC[7:0]。 如果 是 8 位模式，则 CONST_DC/ 256 = 占空比。 如果 是 10 位模式，则 CONST_DC/ 1024 = 占空比

**7.6.9.11 PWM1\_STOP\_VAL\_FRAC\_F 寄存器 ( 偏移 = 410Dh ) [复位 = 00h]**

图 7-160 显示了 PWM1\_STOP\_VAL\_FRAC\_F，表 7-203 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为频率变化/斜坡，则此寄存器用于停止/关断目标时频率的小数分频值。如果配置为占空比斜坡或用于静态 PWM，则不使用此位。

**图 7-160. PWM1\_STOP\_VAL\_FRAC\_F 寄存器**

7	6	5	4	3	2	1	0
RSVD	SPV_FR_F						
R-0h	R/W-0h						

**表 7-203. PWM1\_STOP\_VAL\_FRAC\_F 寄存器字段说明**

位	字段	类型	复位	说明
7	RSVD	R	0h	
6-0	SPV_FR_F	R/W	0h	停止值小数值 ( 仅频率模式 ) 仅当 PWM1_CTRL.MODE = 频率模式时使用。 解释为频率分频值的 x/128 小数部分。 若 PWM1_CTRL.MODE = 占空比模式或静态模式，则不使用该位。

返回到[汇总表](#)。

如果配置为占空比变化/斜坡，则此寄存器用于占空比 **MSB**（仅在使能 10 位占空比分辨率时使用）。如果配置为频率变化/斜坡，则此寄存器用于停止/关断目标时整数分频值的 **MSB**。如果配置为静态 PWM，则不使用该位。

图 7-161. PWM1\_STOP\_VAL\_MSB 寄存器

7	6	5	4	3	2	1	0
RSVD					SPV_F[10]	SPV[9:8]	
R-0h					R/W-0h	R/W-0h	

**表 7-204. PWM1\_STOP\_VAL\_MSB 寄存器字段说明**

位	字段	类型	复位	说明
7-3	RSVD	R	0h	
2	SPV_F[10]	R/W	0h	<p>停止值 (仅频率)</p> <p>这是一个多功能寄存器。</p> <p>如果 PWM1_CTRL.MODE = 频率, 则这是开关频率分频值 STOP_VAL[10] 的整数值。</p> <p>如果 PWM1_CTRL.MODE = 占空比, 则不使用该位。</p> <p>如果 PWM1_CTRL.MODE = 静态, 则不使用该位。</p>
1-0	SPV[9:8]	R/W	0h	<p>停止值</p> <p>这是一个多用途寄存器。</p> <p>如果 PWM1_CTRL.MODE = 占空比且 PWM1_CTRL.MODE = 8B = 0 (10 位), 则这是占空比值 SPV[9:8], 否则不使用该位。</p> <p>SPV[10:24] = 占空比, 在 8 位模式下未使用。</p> <p>如果 PWM1_CTRL.MODE = 频率, 则这是开关频率分频值 STOP_VAL[9:8] 的整数值。</p> <p>如果 PWM1_CTRL.MODE = 静态, 则不使用该位。</p>

**7.6.9.13 PWM1\_STOP\_VAL\_LSB 寄存器 ( 偏移 = 410Fh ) [复位 = 00h]**

图 7-162 显示了 PWM1\_STOP\_VAL\_LSB，表 7-205 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为占空比变化/斜坡，则此寄存器用于占空比 LSB。如果配置为频率变化/斜坡，则此寄存器用于停止/关断目标时整数分频值的 LSB。如果配置为静态 PWM，则不使用该位

**图 7-162. PWM1\_STOP\_VAL\_LSB 寄存器**

7	6	5	4	3	2	1	0
SPV[7:0]							
R/W-0h							

**表 7-205. PWM1\_STOP\_VAL\_LSB 寄存器字段说明**

位	字段	类型	复位	说明
7-0	SPV[7:0]	R/W	0h	停止值 这是一个多用途寄存器。 PWM 1_CTRL.MODE = 占空比，则这是占空比值 SPV[7:0]。 如果 是 8 位模式，则 SPV/256 = 占空比。 如果 是 10 位模式，则 SPV/1024 = 占空比。 如果 PWM 1_CTRL.MODE = 频率，则这是开关频率分频值 STOP_VAL[7:0] 的整数值。 如果 PWM 1_CTRL.MODE = 静态，则不使用该位。

#### 7.6.9.14 PWM1\_STOP\_SL\_MSB 寄存器 ( 偏移 = 4110h ) [复位 = 00h]

图 7-163 显示了 PWM1\_STOP\_SL\_MSB，表 7-206 中对此进行了介绍。

返回到[汇总表](#)。

用于配置停止斜坡的斜率标度

图 7-163. PWM1\_STOP\_SL\_MSB 寄存器

7	6	5	4	3	2	1	0
RSVD				SP_SL[20:16]			
R-0h				R/W-0h			

表 7-206. PWM1\_STOP\_SL\_MSB 寄存器字段说明

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4-0	SP_SL[20:16]	R/W	0h	停止斜率 停止斜坡的斜率。 这些字段会根据 PWM1_CTRL.SLOPE_SCALE 的值而变化。 有关更多信息，请参阅 <a href="#">节 7.5.7.4</a> 如果 PWM1_CTRL.MODE = 静态，则不使用该位。

**7.6.9.15 PWM1\_STOP\_SL\_MID 寄存器 ( 偏移 = 4111h ) [复位 = 00h]**

图 7-164 显示了 PWM1\_STOP\_SL\_MID , 表 7-207 中对此进行了介绍。

返回到[汇总表](#)。

用于配置停止斜坡的斜率标度

**图 7-164. PWM1\_STOP\_SL\_MID 寄存器**

7	6	5	4	3	2	1	0
SP_SL[15:8]							
R/W-0h							

**表 7-207. PWM1\_STOP\_SL\_MID 寄存器字段说明**

位	字段	类型	复位	说明
7-0	SP_SL[15:8]	R/W	0h	停止斜率 停止斜坡的斜率。 这些字段会根据 PWM1_CTRL.SLOPE_SCALE 的值而变化。 有关更多信息, 请参阅节 <a href="#">7.5.7.4</a> 如果 PWM1_CTRL.MODE = 静态, 则不使用该位。

**7.6.9.16 PWM1\_STOP\_SL\_LSB 寄存器 ( 偏移 = 4112h ) [复位 = 00h]**

图 7-165 显示了 PWM1\_STOP\_SL\_LSB，表 7-208 中对此进行了介绍。

返回到[汇总表](#)。

用于配置停止斜坡的斜率标度

**图 7-165. PWM1\_STOP\_SL\_LSB 寄存器**

7	6	5	4	3	2	1	0
SP_SL[7:0]							
R/W-0h							

**表 7-208. PWM1\_STOP\_SL\_LSB 寄存器字段说明**

位	字段	类型	复位	说明
7-0	SP_SL[7:0]	R/W	0h	停止斜率 停止斜坡的斜率。 这些字段会根据 PWM1_CTRL.SLOPE_SCALE 的值而变化。 有关更多信息，请参阅节 <a href="#">7.5.7.4</a> 如果 PWM1_CTRL.MODE = 静态，则不使用该位。



**7.6.9.17 PWM1\_START\_VAL\_FRAC\_F 寄存器 ( 偏移 = 4113h ) [复位 = 00h]**

图 7-166 显示了 PWM1\_START\_VAL\_FRAC\_F，表 7-209 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为频率变化/斜坡，则此寄存器用于启动时频率的小数分频值。如果配置为占空比斜坡或用于静态 PWM，则不使用此位。

**图 7-166. PWM1\_START\_VAL\_FRAC\_F 寄存器**

7	6	5	4	3	2	1	0
RSVD	SPV_FR_F						
R-0h	R/W-0h						

**表 7-209. PWM1\_START\_VAL\_FRAC\_F 寄存器字段说明**

位	字段	类型	复位	说明
7	RSVD	R	0h	
6-0	SPV_FR_F	R/W	0h	起始值小数部分 ( 仅频率模式 ) 仅当 PWM1_CTRL.MODE = 频率模式时使用。 解释为频率分频值的 x/128 小数部分。 若 PWM1_CTRL.MODE = 占空比模式或静态模式，则不使用该位。

### 7.6.9.18 PWM1\_START\_VAL\_MSB 寄存器 ( 偏移 = 4114h ) [复位 = 00h]

图 7-167 显示了 PWM1\_START\_VAL\_MSB，表 7-210 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为占空比变化/斜坡，则此寄存器用于占空比 MSB ( 仅在使能 10 位占空比分辨率时使用 )。如果配置为频率变化/斜坡，则此寄存器用于启动时频率的整数分频值 MSB。如果配置为静态 PWM，则不使用该位

**图 7-167. PWM1\_START\_VAL\_MSB 寄存器**

7	6	5	4	3	2	1	0
RSVD					STV_F[10]	STV[9:8]	
R-0h					R/W-0h		R/W-0h

**表 7-210. PWM1\_START\_VAL\_MSB 寄存器字段说明**

位	字段	类型	复位	说明
7-3	RSVD	R	0h	
2	STV_F[10]	R/W	0h	起始值 ( 仅频率 ) 这是一个多用途寄存器。 如果 PWM1_CTRL.MODE = 频率，则这是开关频率分频值 START_VAL[10] 的整数。 如果 PWM1_CTRL.MODE = 占空比，则不使用该位。 如果 PWM1_CTRL.MODE = 静态，则不使用该位。
1-0	STV[9:8]	R/W	0h	起始值 这是一个多用途寄存器。 如果 PWM1_CTRL.MODE = 占空比且 PWM1_CTRL.MODE = 频率，则这是开关频率分频值 START_VAL[9:8] 的整数。 如果 PWM1_CTRL.MODE = 静态，则不使用该位。

**7.6.9.19 PWM1\_START\_VAL\_LSB 寄存器 ( 偏移 = 4115h ) [复位 = 00h]**

图 7-168 显示了 PWM1\_START\_VAL\_LSB , 表 7-211 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为占空比变化/斜坡, 则此寄存器用于占空比 LSB。如果配置为频率变化/斜坡, 则此寄存器用于启动时频率的整数分频值 LSB。如果配置为静态 PWM, 则不使用该位

**图 7-168. PWM1\_START\_VAL\_LSB 寄存器**

7	6	5	4	3	2	1	0
STV[7:0]							
R/W-0h							

**表 7-211. PWM1\_START\_VAL\_LSB 寄存器字段说明**

位	字段	类型	复位	说明
7-0	STV[7:0]	R/W	0h	起始值 这是一个多用途寄存器。 如果 PWM 1_CTRL.MODE = 占空比, 则这是占空比值 STV[7:0]。 如果 是 8 位模式, 则 STV/256 = 占空比。 如果 是 10 位模式, 则 STV/1024 = 占空比。 如果 PWM 1_CTRL.MODE = 频率, 则这是开关频率分频值 START_VAL[7:0] 的整数值。 如果 PWM 1_CTRL.MODE = 静态, 则不使用该位。

### 7.6.9.20 PWM1\_START\_SL\_MSB 寄存器 ( 偏移 = 4116h ) [复位 = 00h]

图 7-169 显示了 PWM1\_START\_SL\_MSB，表 7-212 中对此进行了介绍。

返回到[汇总表](#)。

用于配置起始斜坡的斜率标度

**图 7-169. PWM1\_START\_SL\_MSB 寄存器**

7	6	5	4	3	2	1	0
RSVD			ST_SL[20:16]				
R-0h			R/W-0h				

**表 7-212. PWM1\_START\_SL\_MSB 寄存器字段说明**

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4-0	ST_SL[20:16]	R/W	0h	起始斜率 起始斜坡的斜率。 这些字段会根据 PWM1_CTRL.SLOPE_SCALE 的值而变化。 有关更多信息，请参阅 <a href="#">节 7.5.7.4</a> 如果 PWM1_CTRL.MODE = 静态，则不使用该位。

#### 7.6.9.21 PWM1\_START\_SL\_MID 寄存器 ( 偏移 = 4117h ) [复位 = 00h]

图 7-170 显示了 PWM1\_START\_SL\_MID，表 7-213 中对此进行了介绍。

返回到[汇总表](#)。

用于配置起始斜坡的斜率标度

图 7-170. PWM1\_START\_SL\_MID 寄存器

7	6	5	4	3	2	1	0
ST_SL[15:8]							
R/W-0h							

表 7-213. PWM1\_START\_SL\_MID 寄存器字段说明

位	字段	类型	复位	说明
7-0	ST_SL[15:8]	R/W	0h	起始斜率 起始斜坡的斜率。 这些字段会根据 PWM1_CTRL.SLOPE_SCALE 的值而变化。 有关更多信息，请参阅节 7.5.7.4 如果 PWM1_CTRL.MODE = 静态，则不使用该位。

### 7.6.9.22 PWM1\_START\_SL\_LSB 寄存器 ( 偏移 = 4118h ) [复位 = 00h]

图 7-171 显示了 PWM1\_START\_SL\_LSB , 表 7-214 中对此进行了介绍。

返回到[汇总表](#)。

用于配置起始斜坡的斜率标度

**图 7-171. PWM1\_START\_SL\_LSB 寄存器**

7	6	5	4	3	2	1	0
ST_SL[7:0]							
R/W-0h							

**表 7-214. PWM1\_START\_SL\_LSB 寄存器字段说明**

位	字段	类型	复位	说明
7-0	ST_SL[7:0]	R/W	0h	起始斜率 起始斜坡的斜率。 这些字段会根据 PWM1_CTRL.SLOPE_SCALE 的值而变化。 有关更多信息，请参阅节 <a href="#">7.5.7.4</a> 如果 PWM1_CTRL.MODE = 静态，则不使用该位。

**7.6.9.23 PWM1\_END\_VAL\_CONST\_FRAC\_F 寄存器 ( 偏移 = 4119h ) [复位 = 00h]**

图 7-172 显示了 PWM1\_END\_VAL\_CONST\_FRAC\_F，表 7-215 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为频率变化，则这是频率的结束值 ( 目标值 ) 小数分频值。如果配置为占空比变化/斜坡，则这是开关频率的小数分频值。如果配置为静态 PWM 输出，则此寄存器用于频率分频值的小数部分。

**图 7-172. PWM1\_END\_VAL\_CONST\_FRAC\_F 寄存器**

7	6	5	4	3	2	1	0
RSVD	EVCV_FR_F						
R-0h	R/W-0h						

**表 7-215. PWM1\_END\_VAL\_CONST\_FRAC\_F 寄存器字段说明**

位	字段	类型	复位	说明
7	RSVD	R	0h	
6-0	EVCV_FR_F	R/W	0h	结束值小数 ( 频率模式 ) 或常数小数 ( 直流模式 ) 结束值小数部分。 此寄存器始终使用，但该字段会根据当前使用的模式而变化。 解释为频率分频值的 x/128 小数部分。 如果 PWM1_CTRL.MODE = 占空比，则这是开关频率分频值 CONST_FREQ[10] 的整数值。 如果 PWM1_CTRL.MODE = 频率或静态，则不使用该位

返回到[汇总表](#)。

如果配置为斜坡，则可设置变化的 **PWM** 结束值（目标值）。如果配置为静态 **PWM** 输出，则此寄存器用于频率分频值的整数 **MSB** 部分。

图 7-173. PWM1\_END\_VAL\_MSB 寄存器

7	6	5	4	3	2	1	0
RSVD					EV_FR[10]	END_VAL[9:8]	
R-0h					R/W-0h	R/W-0h	

**表 7-216. PWM1\_END\_VAL\_MSB 寄存器字段说明**

位	字段	类型	复位	说明
7-3	RSVD	R	0h	
2	EV_FR[10]	R/W	0h	结束值（仅频率） 这是一个多用途寄存器。 如果 PWM1_CTRL.MODE = 占空比或静态，则这是开关频率值 START_VAL[10] 的整数值。 如果 PWM1_CTRL.MODE = 频率，则不使用该位
1-0	END_VAL[9:8]	R/W	0h	结束值 这是一个多用途寄存器。 如果 PWM1_CTRL.MODE = 占空比或静态，则这是开关频率分频值 END_VAL[9:8] 的整数值。 如果 PWM1_CTRL.MODE = 频率且 PWM1_CTRL.8B = 0（10 位），则这是占空比值 END_VAL[9:8]，否则不使用该位。 如果是 10 位模式，则 END_VAL/1024 = 占空比，但在 8 位模式下不使用此位



#### 7.6.9.25 PWM1\_END\_VAL\_LSB 寄存器 ( 偏移 = 411Bh ) [复位 = 00h]

图 7-174 显示了 PWM1\_END\_VAL\_LSB，表 7-217 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为斜坡，则可设置变化的 PWM 结束值 ( 目标值 )。如果配置为静态 PWM 输出，则此寄存器用于频率分频值的整数 LSB 部分。

图 7-174. PWM1\_END\_VAL\_LSB 寄存器

7	6	5	4	3	2	1	0
END_VAL[7:0]							
R/W-0h							

表 7-217. PWM1\_END\_VAL\_LSB 寄存器字段说明

位	字段	类型	复位	说明
7-0	END_VAL[7:0]	R/W	0h	结束值 这是一个多用途寄存器。 如果 PWM 1_CTRL.MODE = 占空比或静态，则这是开关频率分频值 END_VAL[7:0] 的整数值。 如果 PWM 1_CTRL.MODE = 频率，则这是占空比值 END_VAL[7:0]。 如果 是 8 位模式，则 END_VAL/ 256 = 占空比。 如果 是 10 位模式，则 END_VAL/ 1024 = 占空比

7.6.9.26 PWM1\_PULSE\_STOP\_RAMP[y] 寄存器 ( 偏移 = 411Ch + 公式 ) [复位 = 00000000h]

图 7-175 展示了 PWM1\_PULSE\_STOP\_RAMP[y]，表 7-218 中对此进行了介绍。

返回到[汇总表](#)。

如果配置为自动停止斜坡模式，这是从起始脉冲开始，模块将自动设置停止斜坡命令时已生成的 PWM 脉冲数。最低地址对应于计数器的 MSB。

偏移 = 411Ch + (y \* 4h)；其中 y = 0h 至 3h

图 7-175. PWM1\_PULSE\_STOP\_RAMP[y] 寄存器

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM0_PULSE_STOP_RAMP																															
R/W-0h																															

表 7-218. PWM1\_PULSE\_STOP\_RAMP[y] 寄存器字段说明

位	字段	类型	复位	说明
31-0	PWM0_PULSE_STOP_RAMP	R/W	0h	一旦生成指定数量的 PWM 脉冲，停止斜坡将自动启动。仅当 PWM1_ACTION.AUTO_STOP 位置位时使用。最低地址对应 MSB。

**7.6.9.27 PWM1\_PULSE\_MAX[y] 寄存器 ( 偏移 = 4120h + 公式 ) [复位 = 00000000h]**

图 7-176 展示了 PWM1\_PULSE\_MAX[y]，表 7-219 中对此进行了介绍。

返回到[汇总表](#)。

若配置为自动停止斜坡模式，该值表示允许的最大 PWM 脉冲数量。当 PWM 脉冲数达到该限制值时，PWM 输出将立即停止。最低地址对应于计数器的 MSB。

偏移 = 4120h + (y \* 4h)；其中 y = 0h 至 3h

**图 7-176. PWM1\_PULSE\_MAX[y] 寄存器**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PWM0_PULSE_MAX																															
R/W-0h																															

**表 7-219. PWM1\_PULSE\_MAX[y] 寄存器字段说明**

位	字段	类型	复位	说明
31-0	PWM0_PULSE_MAX	R/W	0h	一旦生成指定数量的 PWM 脉冲，PWM 输出将被禁用。 仅当 PWM1_ACTION.AUTO_STOP 位置位时使用。 最低地址对应 MSB。

### 7.6.9.28 PWM1\_ACTION 寄存器 ( 偏移 = 4124h ) [复位 = 00h]

图 7-177 显示了 PWM1\_ACTION , 表 7-220 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-177. PWM1\_ACTION 寄存器**

7	6	5	4	3	2	1	0
RSVD			RESERVED	AUTO_STOP	UDS	启动	STOP
R-0h			R-0h	R/W-0h	RH/W1S-0h	RH/W1S-0h	RH/W1S-0h

**表 7-220. PWM1\_ACTION 寄存器字段说明**

位	字段	类型	复位	说明
7-5	RSVD	R	0h	
4	RESERVED	R	0h	
3	AUTO_STOP	R/W	0h	使用自动停止斜坡 如果使能，一旦脉冲计数达到指定值 0h = 不使用自动停止斜坡行为 1h = 根据脉冲计数寄存器自动开始停止斜坡
2	UDS	RH/W1S	0h	使用定义的起始点 使用在起始值寄存器中定义的起始点。 用于确保起始点已知，或若希望斜坡从当前值（斜坡中途或现有终点）开始。 如果在设置 START 的同时设置此位，则斜坡将从定义的起始点开始。 单独设置此位不会启动斜坡 0h = 使用现有的当前值作为起始点 1h = 从定义的起始点开始
1	启动	RH/W1S	0h	启动 根据配置的设置启动定义的斜坡曲线。 如果同时设置 STOP，则 START 将被忽略。 该位由硬件清除。
0	STOP	RH/W1S	0h	停止 将斜坡运行到指定的停止点，然后关闭。 如果同时设置 START 和 STOP，则会执行 STOP。 该位由硬件清除

**7.6.9.29 PWM1\_IAS\_CTRL 寄存器 ( 偏移 = 4130h ) [复位 = 00h]**

图 7-178 显示了 PWM1\_IAS\_CTRL，表 7-221 中对此进行了介绍。

返回到[汇总表](#)。

**图 7-178. PWM1\_IAS\_CTRL 寄存器**

7	6	5	4	3	2	1	0
RSVD			IN_POL	GPIO_SEL		STOP_MODE	AS_EN
R-0h			R/W-0h	R/W-0h		R/W-0h	R/W-0h

**表 7-221. PWM1\_IAS\_CTRL 寄存器字段说明**

位	字段	类型	复位	说明
7-5	RSVD	R	0h	保留
4	IN_POL	R/W	0h	输入极性 选择用于触发停止行为的电平 0h = 低电平有效输入信号 1h = 高电平有效输入信号
3-2	GPIO_SEL	R/W	0h	GPIO 选择 指定使用哪个 GPIO。 GPIO 必须配置为 GPIO 功能，而不是特殊功能，才能正常工作。 0h = GPIO0 1h = GPIO1 2h = GPIO7 3h = GPIO8
1	STOP_MODE	R/W	0h	停止模式 当发生 GPIO 触发的停止事件时，选择立即禁用 PWM 输出还是请求停止斜坡。 0h = GPIO 触发事件请求停止斜坡 1h = GPIO 触发事件立即禁用 PWM 输出
0	AS_EN	R/W	0h	输入自动停止使能 <div><b>备注</b> 这不要求设置 AUTO_STOP 位，它是一个基于 GPIO 输入停止输出的独立功能</div> 0h = 未通过 GPIO 信号启用自动停止功能 1h = 启用由 GPIO 触发的自动停止功能

## 8 应用和实施

### 备注

以下应用部分中的信息不属于 TI 元件规格，TI 不担保其准确性和完整性。TI 的客户负责确定元件是否适合其用途，以及验证和测试其设计实现以确认系统功能。

### 8.1 应用信息

TCAN5102-Q1 是一款控制局域网 (CAN) 灵活数据 (FD) 轻型响应器器件。对响应器节点的所有控制都通过来自命令器节点处理器的 CAN 总线进行，无需响应器节点处理器和软件。因此，它非常灵活，可用于多种应用，如终端节点电机控制。此处，TCAN5102 与 TCAN11625 或 TCAN11623 等 CAN FD 收发器以及 DRV8889 等步进电机驱动器连接。此处，电机的所有控制指令均可通过 CANH 和 CANL 由控制节点发送，TCAN5102-Q1 能够解析这些指令和配置信息，从而操作步进电机驱动器。在此方案中，远程控制端 (RCE) 无需承担处理器软件开销。

### 8.2 典型应用

RCE CAN FD Light 步进电机控制应用

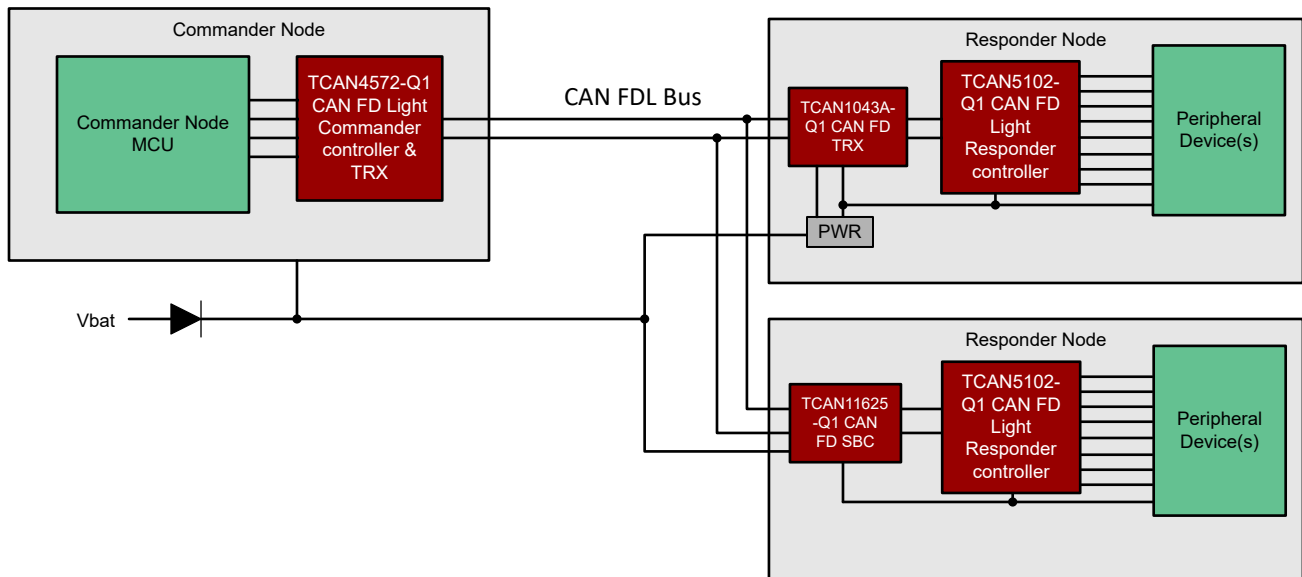


图 8-1. 典型的命令器 - 响应器总线架构

### 8.2.1 设计要求

表 8-1. 设计参数

参数	值
VDD	5V 或 3V
SPI 配置	nSCS、SCLK、SDO、SDI
GPIO 配置	GPIO0 - 故障检测 GPIO1 - 电机禁用 GPIO8 - 电机方向 GPIO11 (PWM) - 电机递增 1 步 GPIO9 - 睡眠使能

### 8.2.2 详细设计过程

要实现步进电机驱动器 RCE，应针对 SPI 模式和 GPIO 适当配置 TCAN5102-Q1。SPI 和 GPIO 命令直接从 CAN 转换到电机驱动器。节 7.6.2 概述了用于将 SPI 数据速率、极性和模式配置到相应电机驱动器的寄存器映射。GPIO 引脚也可通过相应寄存器进行配置。例如，在步进电机示例中，GPIO11 需要配置为 PWM 输出。首先，应将 GPIO11\_CFG 设置为 1h (特殊功能 PWM0)。然后可以根据步进电机的控制/步进方式配置 PWM0，寄存器映射详见节 7.6.8。

(16)

### 8.3 电源相关建议

TCAN5102-Q1 设计为采用单路 5V 或 3V 电源供电。电源可能来自 Vbatt，可通过 CAN FD SBC (如 TCAN11625 或 TCAN11623) 降压，该 SBC 向 VIO/VCCOUT 提供 100mA 或 70mA 电流，足以满足 TCAN5102 (<2mA) 的需求。这部分内容详见节 8.2。

通常，应在 TCAN5102-Q1 的 VDD 引脚附近尽可能靠近地放置一个 1  $\mu$ F 和一个 100nF 的电容。DIGFLTR 引脚上需要一个 1  $\mu$ F 的电容，为内部数字核心稳压器提供滤波。这些电容有助于减少开关模式电源输出中出现的电源电压波纹，并且有助于补偿 PCB 电源层与布线的电阻和电感。

### 8.4 布局

#### 8.4.1 布局指南

当旁路电容和保护器件连接电源和地时，应至少使用两个过孔以更大限度减少布线电感和过孔电感。旁路电容和去耦电容应尽可能靠近收发器的电源端子布置。

布线高速信号线时，要尽量减少过孔和拐角的数量。这将减少阻抗变化量。当不得不采取 90°弯折走线时，采用两个 45°弯折或圆弧形的走线，而非单个 90°的弯折。请勿在晶体、振荡器、外部时钟信号、开关稳压器、安装孔或磁性器件附近布置高速信号线。避免在信号线上出现短线。

#### 8.4.2 布局示例

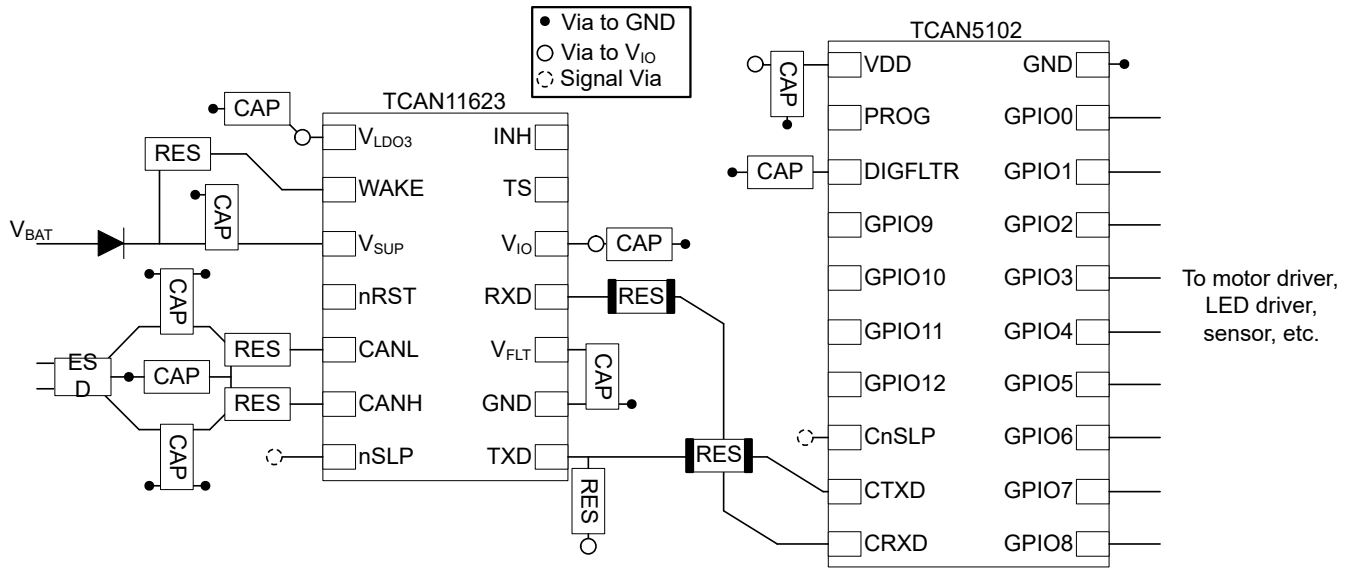


图 8-2. 布局示例



## 9 器件和文档支持

TI 提供广泛的开发工具。下面列出了用于评估器件性能、生成代码和开发解决方案的工具和软件。

### 9.1 文档支持

#### 9.1.1 相关文档

### 9.2 接收文档更新通知

要接收文档更新通知，请导航至 [ti.com](https://www.ti.com) 上的器件产品文件夹。点击 [通知](#) 进行注册，即可每周接收产品信息更改摘要。有关更改的详细信息，请查看任何已修订文档中包含的修订历史记录。

### 9.3 支持资源

[TI E2E™ 中文支持论坛](#) 是工程师的重要参考资料，可直接从专家处获得快速、经过验证的解答和设计帮助。搜索现有解答或提出自己的问题，获得所需的快速设计帮助。

链接的内容由各个贡献者“按原样”提供。这些内容并不构成 TI 技术规范，并且不一定反映 TI 的观点；请参阅 TI 的 [使用条款](#)。

### 9.4 商标

TI E2E™ is a trademark of Texas Instruments.

所有商标均为其各自所有者的财产。

### 9.5 静电放电警告



静电放电 (ESD) 会损坏这个集成电路。德州仪器 (TI) 建议通过适当的预防措施处理所有集成电路。如果不遵守正确的处理和安装程序，可能会损坏集成电路。

ESD 的损坏小至导致微小的性能降级，大至整个器件故障。精密的集成电路可能更容易受到损坏，这是因为非常细微的参数更改都可能会导致器件与其发布的规格不相符。

### 9.6 术语表

[TI 术语表](#) 本术语表列出并解释了术语、首字母缩略词和定义。

## 10 修订历史记录

注：以前版本的页码可能与当前版本的页码不同

Changes from Revision A (October 2025) to Revision B (January 2026)	Page
• 将标题更新为“TCAN5102-Q1 汽车级 CAN FD 灯响应器 (支持 SPI、UART 或 I <sup>2</sup> C 控制器和可编程 PWM)”	<a href="#">1</a>
• 删除了 DGS 封装	<a href="#">224</a>

Changes from Revision * (September 2025) to Revision A (October 2025)	Page
• 将所有提到的名为 4562 的器件更新为 4572	<a href="#">1</a>

## 11 机械、封装和可订购信息

以下页面包含机械、封装和可订购信息。这些信息是指定器件可用的最新数据。数据如有变更，恕不另行通知，且不会对此文档进行修订。有关此数据表的浏览器版本，请查阅左侧的导航栏。

## PACKAGING INFORMATION

Orderable part number	Status (1)	Material type (2)	Package   Pins	Package qty   Carrier	RoHS (3)	Lead finish/ Ball material (4)	MSL rating/ Peak reflow (5)	Op temp (°C)	Part marking (6)
<a href="#">PTCAN5102DGQRQ1</a>	Active	Preproduction	HVSSOP (DGQ)   20	2500   LARGE T&R	-	Call TI	Call TI	-40 to 125	

<sup>(1)</sup> **Status:** For more details on status, see our [product life cycle](#).

<sup>(2)</sup> **Material type:** When designated, preproduction parts are prototypes/experimental devices, and are not yet approved or released for full production. Testing and final process, including without limitation quality assurance, reliability performance testing, and/or process qualification, may not yet be complete, and this item is subject to further changes or possible discontinuation. If available for ordering, purchases will be subject to an additional waiver at checkout, and are intended for early internal evaluation purposes only. These items are sold without warranties of any kind.

<sup>(3)</sup> **RoHS values:** Yes, No, RoHS Exempt. See the [TI RoHS Statement](#) for additional information and value definition.

<sup>(4)</sup> **Lead finish/Ball material:** Parts may have multiple material finish options. Finish options are separated by a vertical ruled line. Lead finish/Ball material values may wrap to two lines if the finish value exceeds the maximum column width.

<sup>(5)</sup> **MSL rating/Peak reflow:** The moisture sensitivity level ratings and peak solder (reflow) temperatures. In the event that a part has multiple moisture sensitivity ratings, only the lowest level per JEDEC standards is shown. Refer to the shipping label for the actual reflow temperature that will be used to mount the part to the printed circuit board.

<sup>(6)</sup> **Part marking:** There may be an additional marking, which relates to the logo, the lot trace code information, or the environmental category of the part.

Multiple part markings will be inside parentheses. Only one part marking contained in parentheses and separated by a "~" will appear on a part. If a line is indented then it is a continuation of the previous line and the two combined represent the entire part marking for that device.

**Important Information and Disclaimer:**The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

## 重要通知和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、与某特定用途的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他安全、安保法规或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 [TI 销售条款](#)、[TI 通用质量指南](#) 或 [ti.com](#) 上其他适用条款或 TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

版权所有 © 2026，德州仪器 (TI) 公司

最后更新日期：2025 年 10 月