

序言

TMS320x2803x Piccolo **增强型局控制器域网 (eCAN)**

Reference Guide



Literature Number: ZHCU041
May 2009

序言

Preface	7
1 CAN 概述	11
1.1 特性	11
1.2 方框图	12
1.3 eCAN 与其他 TI CAN 模块兼容	12
2 CAN 网络和模块	13
2.1 CAN 协议概述	13
3 eCAN 控制器概述	14
3.1 标准 CAN 控制器 (SCC) 模式	15
3.2 内存映射	16
3.3 eCAN 控制和状态寄存器	18
4 消息对象	19
5 消息邮箱	19
5.1 发送邮箱	21
5.2 接收邮箱	22
5.3 CAN 模块运行在正常配置中	22
1 eCAN 寄存器	23
1 邮箱使能寄存器 (CANME)	23
2 邮箱方向寄存器 (CANMD)	24
3 发送请求设置寄存器(CANTRS)	25
4 传输请求-复位寄存器 (CANTRR)	26
5 传输确认寄存器 (CANTA)	27
6 中断确认寄存器 (CANAA)	28
7 接收消息等待寄存器 (CANRMP)	29
8 接收信息丢失寄存器 (CANRML)	30
9 远程帧等待寄存器 (CANRFP)	31
9.1 远程帧的处理	31
10 全局接受屏蔽寄存器 (CANGAM)	33
11 主控制寄存器 (CANMC)	34
11.1 CAN 模块在 SUSPEND (中止) 时工作	35
12 位-时序配置寄存器 (CANBTC)	36
13 错误和状态寄存器 (CANES)	38
14 CAN 错误计数寄存器 (CANTEC/CANREC)	40
15 中断寄存器	41
15.1 全局中断标志寄存器 (CANGIF0/CANGIF1)	41
15.2 全局中断屏蔽寄存器 (CANGIM)	43
15.3 邮箱中断屏蔽寄存器(CANMIM)	46
15.4 邮箱中断级别寄存器 (CANMIL)	47
16 写覆盖保护控制寄存器 (CANOPC)	48
17 eCAN I/O 控制寄存器 (CANTIOC, CANRIOC)	49

18	定时器管理单元	51
	18.1 时间戳功能	51
	18.2 超时功能	54
	18.3 用户应用程序中 MTOF 0/1 位的运行状态/用法。	56
19	邮箱布局	57
	19.1 消息标识符寄存器 (MSGID)	57
	19.2 CPU 邮箱访问	58
	19.3 消息控制寄存器 (MSGCTRL)	59
	19.4 消息数据寄存器 (CANMDL, CANMDH)	60
20	接受滤波器	61
	20.1 本地接受屏蔽 (CANLAM)	61
2	eCAN 配置	63
1	CAN 模块初始化	63
	1.1 CAN 位时序配置	64
	1.2 CAN 比特率计算	65
	1.3 针对30MHz 系统时钟的的位配置参数	65
	1.4 EALLOW 保护	67
2	配置 eCAN 的步骤	67
	2.1 配置邮箱用于传输	68
	2.2 传输一个消息	68
	2.3 配置邮箱用于接收	68
	2.4 接收一个消息	69
	2.5 过载情况下的处理	69
3	远程帧邮箱的处理	69
	3.1 从另一个节点请求数据	69
	3.2 应答一个远程请求	70
	3.3 更新数据字段	70
4	中断	70
	4.1 中断方案	72
	4.2 邮箱中断	72
	4.3 中断处理	73
5	CAN 断电模式	74
	5.1 进入和退出本地断电模式	74
	5.2 进入和退出器件低功耗模式的预防措施 (LPM)	75
	5.3 启用或禁用到 CAN 模块的时钟	75
	5.4 CAN 控制器模块的外部可能故障模式	75

图片列表

1	eCAN 方框图和接口电路	12
2	CAN 数据帧	13
3	eCAN 模块的架构	14
4	eCAN-A 内存映射	17
5	邮箱-使能寄存器 (CANME).....	23
6	邮箱方向寄存器 (CANMD).....	24
7	发送请求设置寄存器 (CANTRS).....	25
8	传输请求-复位寄存器 (CANTRR)	26
9	传输确认寄存器 (CANTA)	27
10	中断确认寄存器 (CANAA)	28
11	接收消息等待寄存器 (CANRMP)	29
12	接收信息丢失寄存器 (CANRML)	30
13	远程帧等待寄存器 (CANRFP)	31
14	全局接受屏蔽寄存器 (CANGAM)	33
15	主控制寄存器 (CANMC)	34
16	位-时序配置寄存器 (CANBTC)	36
17	错误和状态寄存器 (CANES)	38
18	发送错误计数器寄存器 (CANTEC)	40
19	接收错误计数器寄存器 (CANREC)	40
20	全局中断标志寄存器 0 (CANGIF0)	42
21	全局中断标志寄存器 1 (CANGIF1)	42
22	全局中断屏蔽寄存器 (CANGIM)	44
23	邮箱中断屏蔽寄存器(CANMIM).....	46
24	邮箱中断级别寄存器 (CANMIL)	47
25	写覆盖保护控制寄存器 (CANOPC)	48
26	TX I/O 控制寄存器 (CANTIOC)	49
27	RX I/O 控制寄存器 (CANRIOC)	50
28	时间戳计数器寄存器 (CANTSC)	52
29	消息对象时间戳寄存器 (MOTS)	53
30	消息对象超时寄存器 (MOTO)	54
31	超时控制寄存器 (CANTOC)	55
32	超时状态寄存器 (CANTOS)	56
33	消息标识符寄存器 (MSGID) 寄存器	57
34	消息控制寄存器 (MSGCTRL)	59
35	DBO=0(CANMDL) 时的消息数据低寄存器	60
36	DBO=0 (CANMDH) 时的消息数据高寄存器	60
37	DBO=1 (CANMDL) 时的消息数据低寄存器	60
38	DBO=1 (CANMDH) 时的消息数据高寄存器	60
39	本地接受屏蔽寄存器 (LAM _n)	62
40	初始化序列	64
41	CAN 位时序	65
42	中断方案	71

图表列表

1	寄存器映射	18
2	eCAN 邮箱 RAM 布局.....	20
3	针对邮箱 (eCAN-A) 的 LAM, MOTS 和 MOTO 的地址	21
4	消息对象运行状态配置	21
5	邮箱-使能寄存器 (CANME) 字段说明	23
6	邮箱-方向寄存器 (CANMD) 字段说明	24
7	发送请求设置寄存器 (CANTRS) 字段说明	25
8	传输请求-复位寄存器 (CANTRR) 字段说明	26
9	传输确认寄存器 (CANTA) 字段说明	27
10	中断确认寄存器 (CANAA) 字段说明.....	28
11	接收消息挂等待寄存器 (CANRMP) 字段说明	29
12	接收信息丢失寄存器 (CANRML) 字段说明	30
13	远程帧等待寄存器 (CANRFP) 字段说明	31
14	全局接受屏蔽寄存器 (CANGAM) 字段说明	33
15	主控制寄存器 (CANMC) 字段说明.....	34
16	位时序配置寄存器 (CANBTC) 字段描述	36
17	错误和状态寄存器 (CANES) 字段说明	38
18	全局中断标志寄存器 (CANGIF0/CANGIF1) 字段说明	42
19	全局中断屏蔽寄存器 (CANGIM) 字段说明.....	44
20	邮箱中断屏蔽寄存器 (CANMIM) 字段说明.....	46
21	邮箱中断级别寄存器 (CANMIL) 字段说明	47
22	写覆盖保护控制寄存器 (CANOPC) 字段说明	48
23	TX I/O 控制寄存器 (CANTIOC) 字段说明	49
24	RX I/O 控制寄存器 (CANRIOC) 字段说明	50
25	时间戳计数器寄存器 (CANTSC) 字段说明	52
26	消息对象时间戳寄存器 (MOTS) 字段说明	53
27	消息对象超时寄存器 (MOTO) 字段说明	54
28	超时控制寄存器 (CANTOC) 字段说明	55
29	超时状态寄存器 (CANTOS) 字段说明.....	56
30	消息标识符寄存器 (MSGID) 字段说明	57
31	消息控制寄存器 (MSGCTRL) 字段说明.....	59
32	本地接受屏蔽寄存器 (LAM _n) 字段说明	62
33	比特率 (BT = 15, TSEG1 _{reg} =10, TSEG2 _{reg} =2, 采样点 = 80%) 的 BRP 字段	66
34	实现与一个 15 BT 的不同采样点	66
35	比特率 (BT=10, TSEG1 _{reg} =6, TSEG2 _{reg} =1, 采样点 = 80%) 的 BRP 字段	66
36	eCAN 的中断置位/清除	73

关于本手册

本文档描述了增强型局控制器域网 (eCAN) 在 x2803x Piccolo™ 器件。

命名规则

本文档使用以下规则。

- 显示十六进制数时附带一个后缀 h。例如，以下数字表示十六进制的 40（十进制为 64）：40h。

德州仪器 (TI) 提供的相关文档

下列文档描述了 x2803x 器件及相关外设。文档副本可在 www.ti.com 下载获得。

数据指南—

[SPRS584](#) — 《TMS320F28032, TMS320F28033, TMS320F28034, TMS320F28035 Piccolo 微控制器数据手册》包含有 C2803x 器件的引脚分配、信号说明、以及电气和时序技术规范。

[SPRZ295](#) — 《TMS320F28032, TMS320F28033, TMS320F28034, TMS320F28035 Piccolo MCU 芯片勘误表》对与芯片有关的已知报告进行了说明并提供了权变措施。

CPU 用户指南—

[SPRU430](#) — 《TMS320C28x CPU 和指令集参考指南》描述了 TMS320C28x 定点数字信号处理器 (DSP) 的中央处理器 (CPU) 和汇编语言指令。它还描述了这些 DSP 上可用的仿真功能。

外设指南—

[SPRUGL8](#) — 《TMS320x2803x Piccolo 系统控制和中断参考指南》描述了 2803x 微控制器 (MCU) 的不同中断和系统控制特性。

[SPRU566](#) — 《TMS320x28xx, 28xxx DSP 外设参考指南》描述了 28x 数字信号处理器 (DSP) 的外设参考指南。

[SPRUG00](#) — 《TMS320x2803x Piccolo 引导 ROM 参考指南》描述了引导加载程序（由厂家编辑的引导加载软件）的用途和特性并提供代码示例。它还描述了器件的片载引导 ROM 的其它内容，并标识了所有信息在该存储器内的位置。

[SPRUGE6](#) — 《TMS320x2803x Piccolo 控制律加速器 (CLA) 参考指南》描述了控制律加速器 (CLA) 的操作。

[SPRUGE2](#) — 《TMS320x2803x Piccolo 本地互连网络 (LIN) 模块参考指南》描述了本地互连网络 (LIN) 模块的操作。

[SPRUFK8](#) — 《TMS320x2803x Piccolo 增强型正交编码器脉冲 (eQEP) 参考指南》描述了增强型正交编码器脉冲 (eQEP) 的操作。

- [SPRUGL7](#)** — 《TMS320x2803x Piccolo 增强型控制器局域网 (eCAN) 参考指南》描述了增强型控制器局域网 (eCAN) 的操作。
- [SPRUGE5](#)** — 《TMS320x2802x, 2803x Piccolo 模数转换器 (ADC) 和比较器参考指南》描述了如何配置和使用片载 ADC 模块, 此模块是一种 12 位管线型 ADC。
- [SPRUGE9](#)** — 《TMS320x2802x, 2803x Piccolo 增强型脉宽调制器 (ePWM) 模块参考指南》描述了增强型脉宽调制器的主要应用领域, 包括数字电机控制、开关模式电源控制、UPS (不间断电源) 和其它形式的电力转换。
- [SPRUGE8](#)** — 《TMS320x2802x, 2803x Piccolo 高分辨率脉宽调制器 (HRPWM)》描述了脉宽调制器的高分辨率扩展版本 (HRPWM) 的操作。
- [SPRUGH1](#)** — 《TMS320x2802x, 2803x Piccolo 串行通信接口 (SCI) 参考指南》描述了如何使用 SCI。
- [SPRUFZ8](#)** — 《TMS320x2802x, 2803x Piccolo 增强型捕捉 (eCAP) 模块参考指南》描述了增强型捕捉模块。它包括模块描述和寄存器。
- [SPRUG71](#)** — 《TMS320x2802x, 2803x Piccolo 串行外设接口 (SPI) 参考指南》描述了 SPI - 一种高速同步串行输入/输出 (I/O) 端口 - 它允许按照已编程的位传输速率将具有编程长度 (1 到 16 位) 的串行比特流移入或移出器件。
- [SPRUFZ9](#)** — 《TMS320x2802x, 2803x Piccolo 内部集成电路 (I2C) 参考指南》描述了内部集成电路 (I2C) 模块的特性和操作。

工具指南一

- [SPRU513](#)** — 《TMS320C28x 汇编语言工具 v5.0.0 用户指南》描述了用于 TMS320C28x 器件的汇编语言工具 (用于开发汇编语言代码的汇编程序和其它工具)、汇编器指令、宏、通用目标文件格式、和符号调试指令。
- [SPRU514](#)** — 《TMS320C28x 优化 C/C++ 编译器 v5.0.0 用户指南》描述了 TMS320C28x™ C/C++ 编译器。此编译器接受 ANSI 标准 C/C++ 源代码, 并为 TMS320C28x 器件生成 TMS320 DSP 汇编语言源代码。
- [SPRU608](#)** — 《TMS320C28x 指令集模拟器技术概览》描述了用于 TMS320C2000 IDE 的 Code Composer Studio 内提供的模拟器, 此模拟器能够模拟 C28x™ 内核的指令集。

增强控制器局域网 (eCAN) 模块在 C28x™ DSP 内执行，此模块是一个完整的 CAN 控制器并且与 CAN 2.0B 标准（有效）兼容。在电噪声环境下，它使用确定的协议与其它控制器进行串行通信。借助于 32 个完全可配置的邮箱和时间戳功能，eCAN 模块提供一个多用途且耐用的串行通信接口。

eCAN 模块在这份参考指南中是类型 2eCAN。请参阅《TMS320x28xx, 28xxx DSP 外设参考指南》[SPRU566](#) 中其它具有同类型 eCAN 模块的器件列表来确定类型之间的区别，并获得一个类型内器件专用差异的列表。在 28xx/28xxx 的所有器件中，对于给定的 CAN 模块，模块寄存器使用同一个地址空间。

Topic	Page
内容	3
Preface	7
1 CAN 概述	11
2 CAN 网络和模块	13
3 eCAN 控制器概述	14
4 消息对象	19
5 消息邮箱	19
1 邮箱使能寄存器 (CANME)	23
2 邮箱方向寄存器 (CANMD)	24
3 发送请求设置寄存器(CANTRS)	25
4 传输请求-复位寄存器 (CANTRR)	26
5 传输确认寄存器 (CANTA)	27
6 中断确认寄存器 (CANAA)	28
7 接收消息等待寄存器 (CANRMP)	29
8 接收信息丢失寄存器 (CANRML)	30
9 远程帧等待寄存器 (CANRFP)	31
10 全局接受屏蔽寄存器 (CANGAM)	33
11 主控制寄存器 (CANMC)	34
12 位-时序配置寄存器 (CANBTC)	36
13 错误和状态寄存器 (CANES)	38
14 CAN 错误计数寄存器 (CANTEC/CANREC)	40
15 中断寄存器	41
16 写覆盖保护控制寄存器 (CANOPC)	48
17 eCAN I/O 控制寄存器 (CANTIOC, CANRIOC)	49
18 定时器管理单元	51
19 邮箱布局	57
20 接受滤波器	61
1 CAN 模块初始化	63
2 配置 eCAN 的步骤	67
3 远程帧邮箱的处理	69
4 中断	70
5 CAN 断电模式	74

1 CAN 概述

图 1 描述了 eCAN 的主要功能块和接口电路。

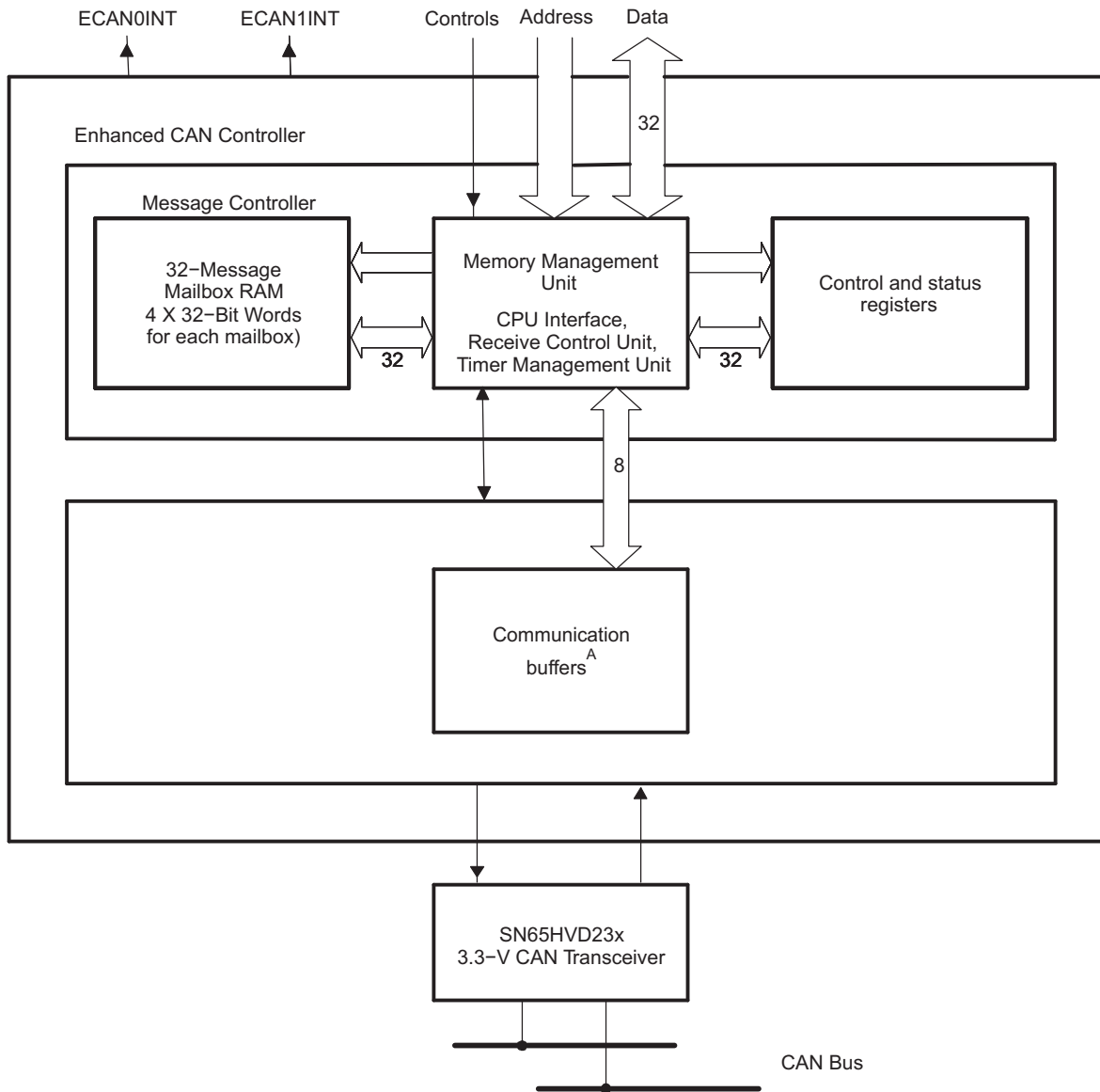
1.1 特性

eCAN 模块有下列特性：

- 与 CAN 协议，版本 2.0B 完全兼容
- 支持高达 1Mbps 的数据速率
- 32 个邮箱，每一个邮箱有下列属性：
 - 可配置为接收或者发送
 - 可使用标准或者扩展标识符进行配置
 - 有一个可编程的接受过滤器屏蔽
 - 支持数据和远程帧
 - 支持 0 到 8 位数据
 - 在接收和发送消息上使用一个 32 位时间戳
 - 防止接收新消息
 - 可以动态设定发送消息优先级
 - 采用一个具有两个中断级别的可编辑中断机制
 - 在发送或者接收超时时采用可编辑中断
- 低功耗模式
- 可编辑的总线活动时唤醒
- 对一个远程请求消息的自动应答
- 丢失仲裁或者错误情况下的帧自动重传
- 32 位时间戳计数器由一个特定的消息（与信箱 16 协同通信）同步处理
- 自检测模式
 - 运行在接收其自身消息的回路模式。提供一个“假”确认，从而消除了对另外节点提供确认位的需要。

1.2 方框图

图 1. eCAN 方框图和接口电路



A 对用户来说，通信缓冲区是透明的，用户代码访问也不可访问。

1.3 eCAN 与其他 TI CAN 模块兼容

eCAN 模块和在 TMS470 中使用的“高端 CAN 控制器 (HECC)”相同。TMS470™ 系列微控制器，德州仪器 (TI) 对此类器件做出了一些细微改变。eCAN 模块特有多项增强功能（如增加了具有独立接受屏蔽、时间戳等功能的邮箱数量），这些改进使 eCAN 模块在性能上超过了 240x™ DSP 系列所特有的 CAN 模块。所以，为 240x CAN 模块编写的代码不能直接移植到 eCAN 上。然而，eCAN 沿循着 240x CAN 同样的寄存器位布局结构和位功能（因为这两个器件中都装有寄存器）也就是说，在这两种平台上许多寄存器和位执行一样的功能。这就使得代码的迁移变的相对容易，对于用 C 语言编写的代码更是如此。

2 CAN 网络和模块

控制器局域网 (CAN) 使用一个串行多主机通信协议，此协议有效地支持分布式实时控制，具有非常高的安全级别，并且通信速率可达 1Mbps。CAN 总线是嘈杂和恶劣的环境，如汽车和其他要求可靠通信的工业领域应用的理想选择。

高达 8 字节数据长度的已设定优先级的消息可以通过多主机串行总线发送，此总线使用一个仲裁协议和一个错误检测机制来确保高度的数据完整性。

2.1 CAN 协议概述

CAN 协议支持四种不同的通信帧类型：

- 数据帧将数据从发射器节点发往数据接收器节点
- 节点发出远程帧请求发送具有同一识别符的数据帧
- 一个总线错误检测上任一节点发出的错误帧
- 过载帧用以在先行的和后续的数据帧或远程帧之间提供附加延时

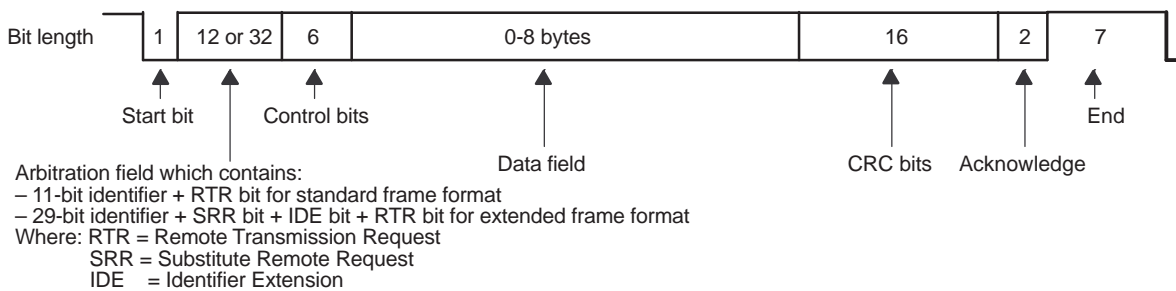
此外，CAN 2.0B 版技术规范定义了两种标识符字段长度不同的格式帧：11 位标识符的标准帧和 29 位标识符的扩展帧。

CAN 标准数据帧包含 44 到 108 位而 CAN 扩展数据帧包含 64 到 128 位。另外，标准数据帧最高可插入 23 个填充位，而扩展数据帧最高可插入 28 个填充位，这主要取决于数据流编码。标准帧的最大总数据帧长度为 131 位，而扩展帧为 156 位。

位字段组成了标准或扩展数据帧，同时它们的位置也在图 2 中显示如下：

- 帧起始
- 包含标识符和发送消息类型的仲裁字段
- 表示已传输位数的控制字段。
- 多达 8 个数据字节
- 循环冗余校验 (CRC)
- 确认
- 帧结束位

图 2. CAN 数据帧

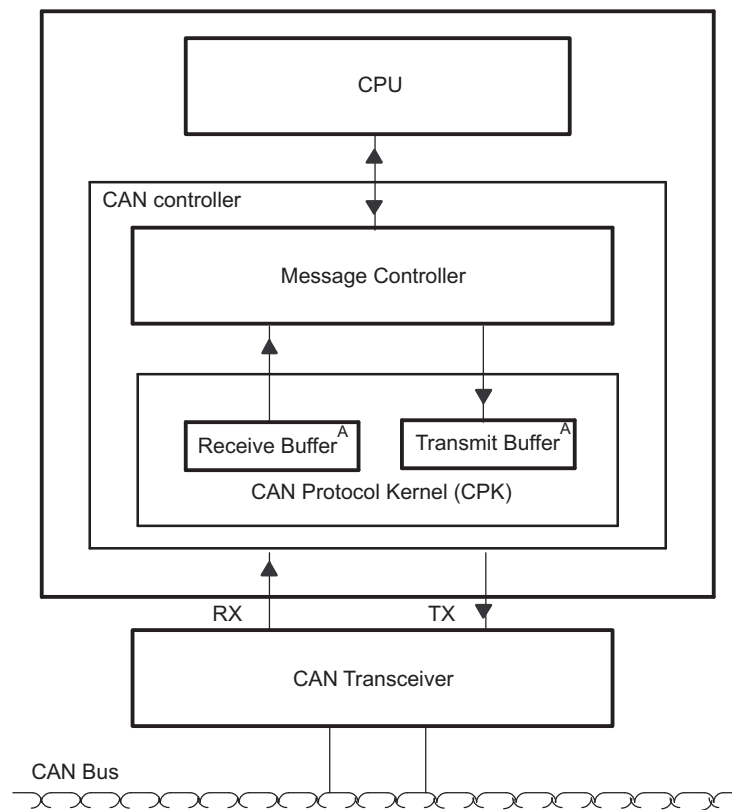


Note: Unless otherwise noted, numbers are amount of bits in field.

eCAN 控制器为 CPU 提供 CAN 协议 2.0B 版本的完全功能。CAN 控制器最大限度地减少了 CPU 在通信开销中的负载，并通过提供额外的特性提高了 CAN 标准。

eCAN 模块的架构，如图 3 所示，是由一个 CAN 协议内核 (CPK) 和一个消息控制器组成。

图 3. eCAN 模块的架构



A 接收和发送缓冲器对用户是透明的，用户代码不能访问。

根据 CAN 协议，CPK 的两个功能的是解码在 CAN 总线上接收到的所有消息，并把这些消息转移到接收缓冲器。CPK 的另一个功能是根据 CAN 协议传输 CAN 总线上的消息。

CAN 控制器的消息控制器是负责确定 CPK 接收到的任何信息是否必须被保留以便用于 CPU 使用还是被丢弃。在初始化阶段，CPU 指定消息控制器应用所使用的所有消息标识符。根据消息的优先级，消息控制器还负责发送下一条消息传输到 CPK。

3 eCAN 控制器概述

eCAN 是一个带有内部 32 位架构的 CAN 控制器。

eCAN 模块组成：

- CAN 协议内核 (CPK)
- 消息控制器包括：
 - 内存管理单元 (MMU)，包括 CPU 接口和接收控制单元（接受过滤），定时器管理单元
 - 能存储 32 条消息的邮箱 RAM
 - 控制和状态寄存器

在接收一个有效的 CPK 的消息后，消息控制器的接收控制单元确定接收到的消息是否必须存储到邮箱 RAM 中的 32 个消息对象中的一个对象中。接收控制单元检查状态、标识符、和所有消息对象的屏蔽，以确定相应邮箱位置。收到的信息通过接受过滤后被存储在第一个邮箱。如果接收控制单元无法找到任何邮箱来存储接收到的消息，则该消息将被丢弃。

一条消息由 11 或 29 位标识符、一个控制字段，以及多达 8 个数据字节组成。

当一条消息必须传输时，消息控制器将该消息传输到 CPK 的发送缓冲区，以便在下次总线空闲状态时开始传输该消息。当必须传输多个消息时，已做好传输准备的、具有最高优先级的消息将由消息控制器传输到 CPK 中。如果两个邮箱具有相同的优先级，那么具有较高的号码的邮箱优先传输。

定时器管理单元包含一个时间戳计数器并且将一个时间戳置于所有接收到或已发送消息的附近。当一条消息还未收到或在允许的时间期限（超时）尚未传输时，会生成一个中断。时间戳的功能是只有在 eCAN 模式时可用。

若要启动数据传输，必须在相应的控制寄存器设置传输请求 (TRS.n)。整个传输过程中可能出现的错误处理在没有 CPU 介入的情况下执行。如果一个邮箱已经被配置用来接收消息，CPU 就能很容易地使用 CPU 读指令读取其数据寄存器。在每一个成功的消息发送或接收到后，邮箱可配置为用来中断 CPU。

3.1 标准 CAN 控制器 (SCC) 模式

SCC 模式是 eCAN 的一种精简功能模式。在这种模式下，只有 16 个邮箱（0 到 15）可用。不提供时间戳功能，并且可用的接受屏蔽数量也被减少。缺省情况下，这种模式被选中。通过 SCB 位 (CANMC.13)，SCC 模式或功能齐全的 eCAN 模式被选择。

3.2 内存映射

eCAN 模块有映射到存储器中的两个不同地址段。第一段地址空间分配给控制寄存器、状态寄存器、接受屏蔽、时间戳和消息对象超时。到控制寄存器和状态寄存器的访问被限制为 32 位位宽访问。本地接受屏蔽、时间戳寄存器、和超时寄存器均可以 8 位, 16 位, 32 位宽访问。第二个地址段用于访问邮箱。这种内存范围可以 8 位, 16 位, 32 位宽访问。图 4 所示的两个内存区块中的任一个使用的地址空间均为 512 字节。

消息的存储由一个可由 CAN 控制器或 CPU 寻址的 RAM 执行。CPU 通过修改 RAM 或附加寄存器中的各种邮箱来控制 CAN 控制器。各存储元件的内容都被用来进行接受过滤、消息传输、和中断处理。

eCAN 内的邮箱模块提供了 32 个消息邮箱, 这些邮箱有 8 字节数据长度、29 位标识符, 和几个控制位。每个邮箱都可配置为发送或接收。在 eCAN 模式下, 每个邮箱都有独立的接受屏蔽。

注: 不在应用 (在 CANME 寄存器中禁用) 中使用的 LAMn, MOTSn 和 MOTOn 寄存器可被 CPU 用作通用数据存储。

3.2.1 到控制和状态寄存器的 32 位访问

正如 3.2 节中指出的那样, 控制和状态寄存器只允许 32 位访问。到这些寄存器的 16 位访问可能会破坏该寄存器的内容或返回错误数据。TI 发布的 DSP 头文件采用了一个有助于 32 位访问的影子寄存器结构。下面的例子将说明如何采用影子寄存器结构来执行 32 位读取和写入:

Example 1. 修改寄存器中的位

```
ECanaShadow.CANTIOC.all = ECanaRegs.CANTIOC.all; // Step 1
ECanaShadow.CANTIOC.bit.TXFUNC = 1; // Step 2
ECanaRegs.CANTIOC.all = ECanaShadow.CANTIOC.all; // Step 3
```

第 1 步: 执行一个 32 位读取以复制整个寄存器内容到它的影子寄存器中

第 2 步: 修改影子寄存器中所需要的一个或多个位

第 3 步: 执行一个 32 位写入以复制修改后的影子寄存器内容到原始寄存器中。

注: 对于像 TAN 和 RMPn 这类的位, 通过写入一个 1 来清除。应小心不要意外地清除某些位。

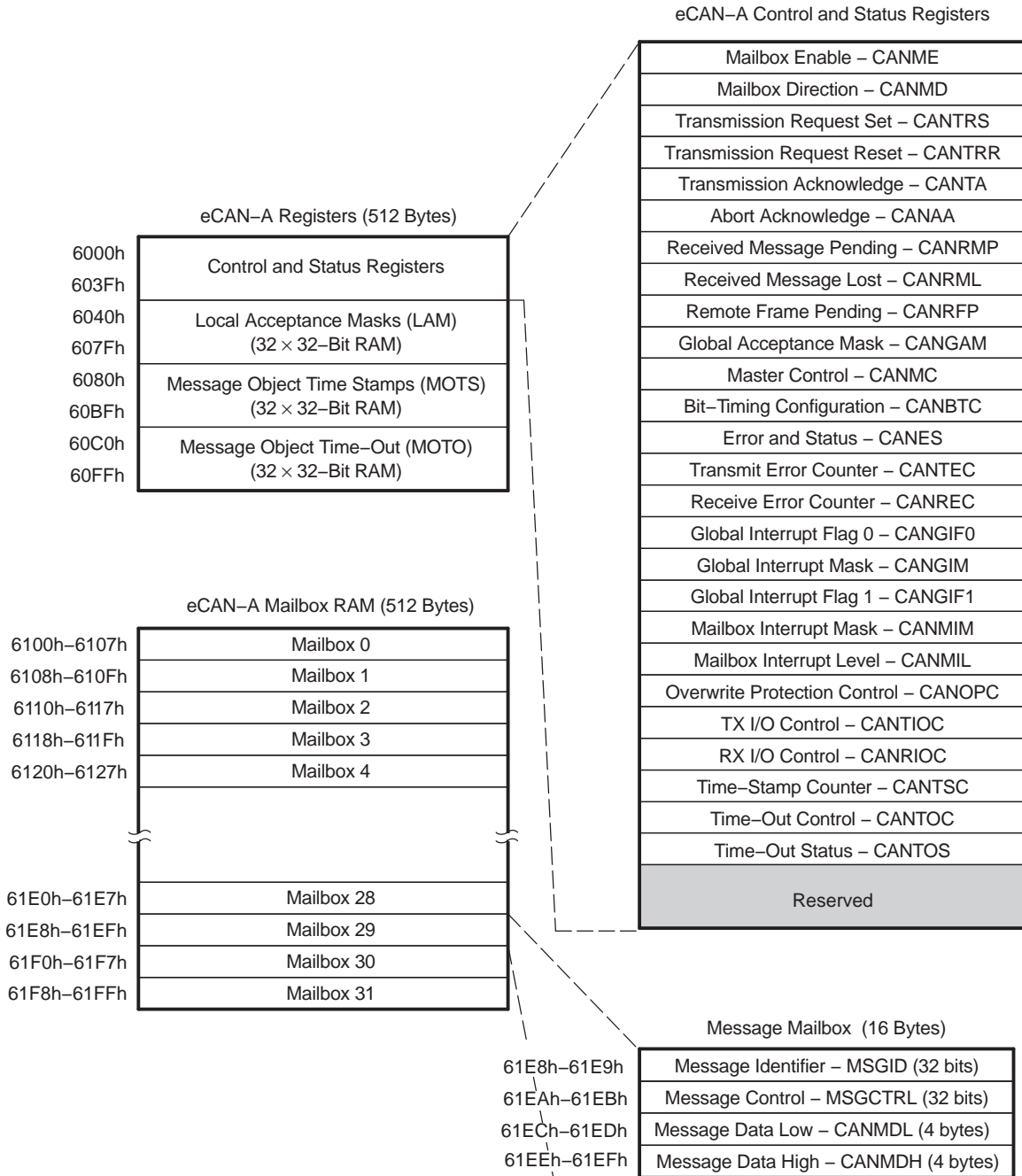
Example 2. 检查一个寄存器中的位值

```
do
{
    ECanaShadow.CANTA.all = ECanaRegs.CANTA.all;
} while(ECanaShadow.CANTA.bit.TA25 == 0); // Wait for TA5 bit to be set..
```

在上述例子中, TA25 的位值就需要进行检查。完成步骤如下, 首先复制整个 CANTA 寄存器到它的影子寄存器 (使用 32 位读取), 然后检查相关位, 重复此操作, 直到该条件得到满足。下面的情况中不应检查 TA25 位:

```
while(ECanaRegs.CANTA.bit.TA25 == 0);
```


图 4. eCAN-A 内存映射



3.3 eCAN 控制和状态寄存器

CPU 使用表 1 中列出的 CAN 寄存器来配置和控制 CAN 控制器和消息目标

表 1. 寄存器映射

寄存器名称 ⁽¹⁾	ECAN-A 地址	尺寸 (x32)	说明
CANME	0x6000	1	邮箱启用
CANMD	0x6002	1	邮箱方向
CANTRS	0x6004	1	发送请求设定
CANTRR	0x6006	1	发送请求复位
CANTA	0x6008	1	传输确认
CANAA	0x600A	1	中止确认
CANRMP	0x600C	1	接收消息等待
CANRML	0x600E	1	接收消息丢失
CANRFP	0x6010	1	远程帧等待
CANGAM	0x6012	1	全局接受屏蔽
CANMC	0x6014	1	主器件控制
CANBTC	0x6016	1	位时序配置
CANES	0x6018	1	错误和状态
CANTEC	0x601A	1	发送错误计数器
CANREC	0x601C	1	接收错误计数器
CANGIF0	0x601E	1	全局中断标志 0
CANGIM	0x6020	1	全局中断屏蔽
CANGIF1	0x6022	1	全局中断标志 1
CANMIM	0x6024	1	邮箱中断屏蔽
CANMIL	0x6026	1	邮箱中断级别
CANOPC	0x6028	1	写覆盖保护控制
CANTIOC	0x602A	1	TX I/O 控制
CANRIOC	0x602C	1	RX I/O 控制
CANTSC	0x602E	1	时间戳计数器 (在 SCC 模式中被保留)
CANTOC	0x6030	1	超时控制 (在 SCC 模式中被保留)
CANTOS	0x6032	1	超时状态 (在 SCC 模式中被保留)

⁽¹⁾ 这些寄存器被映射至外设帧 1。

注: 到控制和状态寄存器的访问只允许 32 位访问。这一限制并不适用于 RAM 区域的邮箱。更多信息请查阅节 3.2.1。

4 消息对象

eCAN 模块有 32 个不同的消息对象（邮箱）。

每个消息对象可以配置为发送或接收。每个消息对象有其各自的接受屏蔽。

一个消息对象由一个消息邮箱组成，其中包括：

- 29 位的消息标识符
- 消息控制寄存器
- 8 字节的信息数据
- 一个 29 位接受屏蔽
- 一个 32 位时间戳
- 一个 32 位超时值

此外，位于寄存器中的相应的控制和状态位允许对消息对象的控制。

5 消息邮箱

消息邮箱是 RAM 区域，这些区域就是消息被接收后或被传输前 CAN 信息实际存储的地方。

CPU 可将消息邮箱中不被用来存储消息的 RAM 区用作普通内存。

每个邮箱包含：

- 消息标识符
 - 29 位扩展标识符
 - 11 位标准标识符
- 标识符扩展位，IDE (MSGID.31)
- 接受屏蔽使能位，AME (MSGID.30)
- 自动应答模式位，AAM (MSGID.29)
- 发送优先级，TPL (MSGCTRL.12-8)
- 远程传输请求位，RTR (MSGCTRL.4)
- 数据长度代码，DLC (MSGCTRL.3-0)
- 多达 8 字节的数据区字段

每个邮箱可以配置为 4 个消息对象类型之一（请见）。发送和接收消息对象被用于一个发送器和多个接收器（一个到 n 个之间的通信连接）间的数据交换，而请求和应答消息对象被用于建立一对一通信连接。表 2 列出了邮箱 RAM 的布局。

表 2. eCAN 邮箱 RAM 布局

邮箱	MSGID MSGIDL-MSGIDH	MSGCTRL MSGCTRL-Rsvd	CANMDL CANMDL_L-CANMDL_H	CANMDH CANMDH_L-CANMDH_H
0	6100-6101h	6102-6103h	6104-6105h	6106-6107h
1	6108-6109h	610A-610Bh	610C-610Dh	610E-610Fh
2	6110-6111h	6112-6113h	6114-6115h	6116-6117h
3	6118-6119h	611A-611Bh	611C-611Dh	611E-611Fh
4	6120-6121h	6122-6123h	6124-6125h	6126-6127h
5	6128-6129h	612A-612Bh	612C-612Dh	612E-612Fh
6	6130-6131h	6132-6133h	6134-6135h	6136-6137h
7	6138-6139h	613A-613Bh	613C-613Dh	613E-613Fh
8	6140-6141h	6142-6143h	6144-6145h	6146-6147h
9	6148-6149h	614A-614Bh	614C-614Dh	614E-614Fh
10	6150-6151h	6152-6153h	6154-6155h	6156-6157h
11	6158-6159h	615A-615Bh	615C-615Dh	615E-615Fh
12	6160-6161h	6162-6163h	6164-6165h	6166-6167h
13	6168-6169h	616A-616Bh	616C-616Dh	616E-616Fh
14	6170-6171h	6172-6173h	6174-6175h	6176-6177h
15	6178-6179h	617A-617Bh	617C-617Dh	617E-617Fh
16	6180-6181h	6182-6183h	6184-6185h	6186-6187h
17	6188-6189h	618A-618Bh	618C-618Dh	618E-618Fh
18	6190-6191h	6192-6193h	6194-6195h	6196-6197h
19	6198-6199h	619A-619Bh	619C-619Dh	619E-619Fh
20	61A0-61A1h	61A2-61A3h	61A4-61A5h	61A6-61A7h
21	61A8-61A9h	61AA-61ABh	61AC-61ADh	61AE-61AFh
22	61B0-61B1h	61B2-61B3h	61B4-61B5h	61B6-61B7h
23	61B8-61B9h	61BA-61BBh	61BC-61BDh	61BE-61BFh
24	61C0-61C1h	61C2-61C3h	61C4-61C5h	61C6-61C7h
25	61C8-61C9h	61CA-61CBh	61CC-61CDh	61CE-61CFh
26	61D0-61D1h	61D2-61D3h	61D4-61D5h	61D6-61D7h
27	61D8-61D9h	61DA-61DBh	61DC-61DDh	61DE-61DFh
28	61E0-61E1h	61E2-61E3h	61E4-61E5h	61E6-61E7h
29	61E8-61E9h	61EA-61EBh	61EC-61EDh	61EE-61EFh
30	61F0-61F1h	61F2-61F3h	61F4-61F5h	61F6-61F7h
31	61F8-61F9h	61FA-61FBh	61FC-61FDh	61FE-61FFh

表 3. 针对邮箱 (eCAN-A) 的 LAM, MOTS 和 MOTO 的地址

邮箱	LAM	MOTS	MOTO
0	6040h-6041h	6080h-6081h	60C0h-60C1h
1	6042h-6043h	6082h-6083h	60C2h-60C3h
2	6044h-6045h	6084h-6085h	60C4h-60C5h
3	6046h-6047h	6086h-6087h	60C6h-60C7h
4	6048h-6049h	6088h-6089h	60C8h-60C9h
5	604Ah-604Bh	608Ah-608Bh	60CAh-60CBh
6	604Ch-604Dh	608Ch-608Dh	60CCh-60CDh
7	604Eh-604Fh	608Eh-608Fh	60CEh-60CFh
8	6050h-6051h	6090h-6091h	60D0h-60D1h
9	6052h-6053h	6092h-6093h	60D2h-60D3h
10	6054h-6055h	6094h-6095h	60D4h-60D5h
11	6056h-6057h	6096h-6097h	60D6h-60D7h
12	6058h-6059h	6098h-6099h	60D8h-60D9h
13	605Ah-605Bh	609Ah-609Bh	60DAh-60DBh
14	605Ch-605Dh	609Ch-609Dh	60DCh-60DDh
15	605Eh-605Fh	609Eh-609Fh	60DEh-60DFh
16	6060h-6061h	60A0h-60A1h	60E0h-60E1h
17	6062h-6063h	60A2h-60A3h	60E2h-60E3h
18	6064h-6065h	60A4h-60A5h	60E4h-60E5h
19	6066h-6067h	60A6h-60A7h	60E6h-60E7h
20	6068h-6069h	60A8h-60A9h	60E8h-60E9h
21	606Ah-606Bh	60AAh-60ABh	60EAh-60EBh
22	606Ch-606Dh	60ACh-60ADh	60ECh-60EDh
23	606Eh-606Fh	60AEh-60AFh	60EEh-60EFh
24	6070h-6071h	60B0h-60B1h	60F0h-60F1h
25	6072h-6073h	60B2h-60B3h	60F2h-60F3h
26	6074h-6075h	60B4h-60B5h	60F4h-60F5h
27	6076h-6077h	60B6h-60B7h	60F6h-60F7h
28	6078h-6079h	60B8h-60B9h	60F8h-60F9h
29	607Ah-607Bh	60BAh-60BBh	60FAh-60FBh
30	607Ch-607Dh	60BCh-60BDh	60FCh-60FDh
31	607Eh-607Fh	60BEh-60BFh	60FEh-60FFh

表 4. 消息对象运行状态配置

消息对象运行状态	邮箱方向寄存器 (CANMD)	自动应答模式位 (AAM)	远程传输请求位 (RTR)
发送消息对象	0	0	0
接收消息对象	1	0	0
请求消息对象	1	0	1
应答消息对象	0	1	0

5.1 发送邮箱

CPU 将要发送的数据存储在一个配置为发送邮箱的邮箱中。在把数据和标识符写入到 RAM 中后, 如果相应的 TRS[n] 位已设置, 假如通过设置相应的 CANME.n 位邮箱被启用的话, 则消息被发送。

如果不止一个邮箱被设为发送邮箱, 并且不止一个 TRS[n] 位被置位, 则消息从具有最高优先级的邮箱按照优先级下降的次序依次发送。

在 SCC 兼容模式下，邮箱传输的优先级取决于邮箱号。最高的邮箱号 (=15) 包含最高优先级。

在 eCAN 模式下，邮箱传输的优先级取决于消息控制器字段 (MSGCTRL) 寄存器中的 TPL 字段的设置。具有最高 TPL 值的邮箱首先被发送。只有当在两个邮箱的 TPL 值相同时，更高号码的邮箱首先发送。

如果一个传输由于仲裁丢失或错误而失败，则消息将重新尝试传输。在重新传输前，CAN 模块检查是否有另外一个传输被请求，然后传输优先级最高的邮箱。

5.2 接收邮箱

每个进入消息的标识符与保存在使用适当屏蔽的接收邮箱内的标识符相比较。当两者匹配时，接收到的标识符，控制位和数据字节将被写进匹配的 RAM 位置。同时，相应的接收消息等待位，RMP[n](RMP.31-0) 被设置，如果被启用则产生一个接收中断。如果未检测到匹配，则消息不存储。

当消息接收后，消息控制器从邮箱编号最高的邮箱开始搜索一个匹配的标识符。在 SCC 兼容模式下的 eCAN 的邮箱 15 具有最高接收优先级；在 eCAN 模式下，eCAN 的邮箱 31 具有最高接收优先级。

在读取数据后，CPU 必须将 RMP[n] (RMP.31-0) 复位。如果这个邮箱又接收到第二个消息，且接受消息等待位已经被设定，相应的消息丢失位 (RML[n] (RML.31-0)) 被设置。这种情况下，如果写覆盖保护位 OPC[n] (OPC.31-0) 被清除，则存储的消息被新消息写覆盖；否则，检查下一个邮箱。

如果一个邮箱被设置为一个接收邮箱并且发送请求位 (RTR) 被设置，则邮箱可以发送一个远程帧。一旦一个远程帧被发送，则邮箱的 TRS 位被 CAN 模块清除。

5.3 CAN 模块运行在正常配置中

如果 CAN 模块被用于正常配置下（即，非自检测模式），则网络中至少要有两个 CAN 模块，且比特率设置相同。另外一个 CAN 模块不必设置为真正接收来自发送节点的消息。但是，它应被设置为相同的比特率。这是因为，一个 CAN 发送模块期望 CAN 网络中至少一个节点确认已发送消息的正确接收。按照 CAN 协议技术规范，任何接收到信息的 CAN 节点将确认（除非取认证理被明确关闭），不管是否被设置为存储接收的消息。在 C28x DSP 中确认机制中是不可被关闭的。

在自测模式 (STM) 下，不需要另外一个节点。在该模式下，一个传输节点产生自己的确认信号。唯一的要求就是节点必须配置任一有效的比特率。也就是说，位时序寄存器不应该包含一个不被 CAN 协议所允许的值。

不能通过将 CANTX 和 CANRX 引脚连接在一起实现直接外部数字回路（使用 SCI/SPI/McBSP 模块是可能的）。在自检测模式 (STM) 下，可实现内部回路。

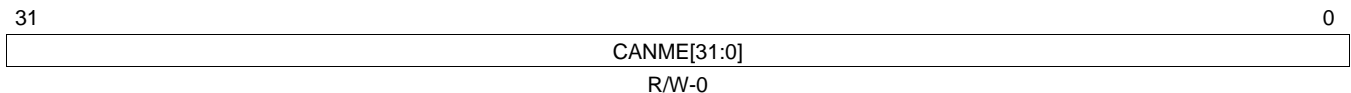
本章包括寄存器及位说明。

Topic	Page

1 邮箱使能寄存器 (CANME)

该寄存器用于启用/禁用个人邮箱。

图 5. 邮箱-使能寄存器 (CANME)



图例：R/W = 读取/写入；-n = 复位后的值

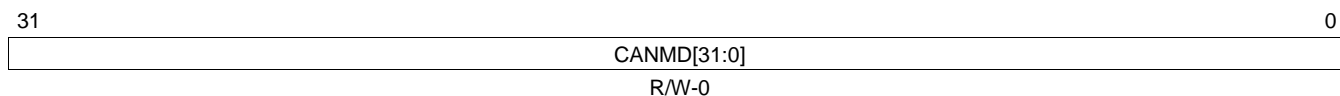
表 5. 邮箱-使能寄存器 (CANME) 字段说明

位	字段	值	说明
31:0	CANME[31:0]		邮箱使能位。加电后，CANME 中的所有位都将被清除。被禁用的邮箱可以被用来作为 CPU 附加内存。
		1	CAN 模块中相应的邮箱被启用。在写入任何标识符字段内容前，邮箱必须被禁用。如果 CANME 的相应位被设定，邮箱标识符的写入权限将被拒绝。
		0	对于 eCAN，相应的邮箱 RAM 区段被禁用；然而，CPU 可将其作为正常 RAM 进行访问。

2 邮箱方向寄存器 (CANMD)

该寄存器被用于配置邮箱的接收或发送操作。

图 6. 邮箱方向寄存器 (CANMD)



图例：R/W = 读取/写入；-n = 复位后的值

表 6. 邮箱-方向寄存器 (CANMD) 字段说明

位	字段	值	说明
31:0	CANMD[31:0]		邮箱方向位 加电后，所有的位被清除。
		1	相应的邮箱被配置为一个接收邮箱。
		0	相应的邮箱被配置为一个发送邮箱。

3 发送请求设置寄存器(CANTRS)

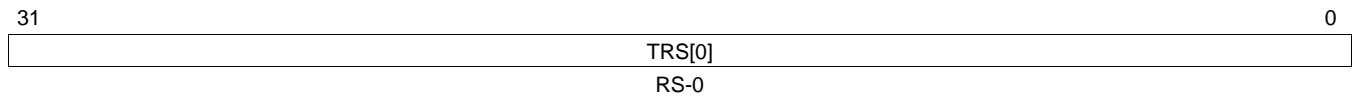
当邮箱 n 准备传送时，CPU 应当设置 TRS_n 位来启动传送。

这些位通常由 CPU 设置和 CAN 模块逻辑操作清除。CAN 模块可以为远程帧请求设置这些位。当传输成功或中止时，这些位被复位。如果邮箱被配置为接收邮箱，则 CANTRS 中相应的位被忽略，除非接收邮箱被配置用来处理远程帧。如果 RTR 位被设置，则接收邮箱的 $TRS[n]$ 位不能被忽视。因此，如果接收邮箱的 TRS 位被设置，那么接收邮箱（RTR 位被设置）也可以用来发送远程帧。一旦一个远程帧被发送，则邮箱的 $TRS[n]$ 位被 CAN 模块清除。因此，同一个邮箱可被用来从其它模式中请求一个数据帧。如果 CPU 试图设置一个位，而 eCAN 试图清除它，则该位被设置。

设置 $CANTRS[n]$ 引起特定消息 n 应该被发送。可以同时设置几个位。因此，所有关于 TRS 位的消息将被依次传送，如果没有设定 TPL 位指令，则首先传送邮箱编号最高的（=最高优先级）的邮箱。

CANTRS 中的这些位由 CPU 写入一个 1 来设定。写入 0 无效。加电后，所有位被清除。

图 7. 发送请求设置寄存器 (CANTRS)



图例：RS = 读取/设置；-n= 复位后的值

表 7. 发送请求设置寄存器 (CANTRS) 字段说明

位	字段	值	说明
31:0	TRS[0]	1	发送-请求-设置位
		0	在那个邮箱中设置 TRS_n 传送消息。可同时设置几个位，所有的消息依次传送。
			无操作

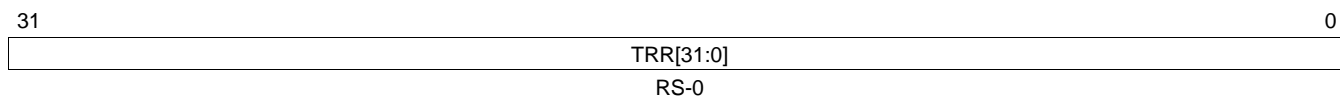
4 传输请求-复位寄存器 (CANTRR)

这些位只能通过 CPU 置位和由内部逻辑复位。当一个传输成功或则中断时，这些位被复位。如果 CPU 试图在 CAN 清除该位时设置该位，则该位被置位。

如果该请求是由相应位 (TRS[n]) 位发起的并且当前并未被处理的话，设置消息对象 n 的 TRR[n] 位将取消传输请求。如果相应的消息当前正在被处理，则当传输完成（正常操作）或者由于丢失仲裁或检测到 CAN 总线线路上存在错误状态而导致传输中止时，该位被复位。当一个传输被中止时，相应的状态位 (AA.31-0) 被置位。当传输成功时，状态位 (TA.31-0) 被置位。传输请求复位状态可从 TRS.31-0 位读取。

CANTRR 寄存器位可以由 CPU 写入一个 1 来设置。

图 8. 传输请求-复位寄存器 (CANTRR)



图例：RS = 读取/设置；- n = 复位后的值

表 8. 传输请求-复位寄存器 (CANTRR) 字段说明

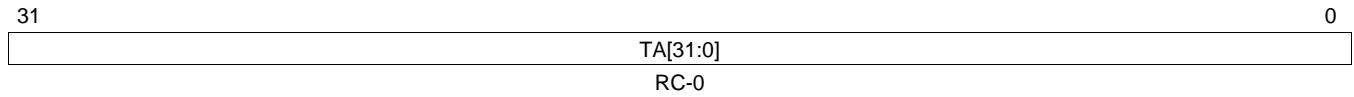
位	字段	值	说明
31:0	TRR[31:0]		发送请求复位位
		1	TRR n 设置会取消传输请求
		0	无操作

5 传输确认寄存器 (CANTA)

如果邮箱 n 的消息被成功发送，则 TA_n 位被置位。如果 CANMIM 寄存器中相应的中断屏蔽位被设置，此操作也将设置 GMIF0/GMIF1 (GIF0.15/GIF1.15) 位。GMIF0/GMIF1 位发起一个中断。

CPU 通过写入一个 1 来复位 CANTA 中的位。如果一个中断已经产生，这也将清除中断。写入 0 无效。如果 CPU 试图在 CAN 对一个位置位时对其复位，则该位被置位。加电后，所有的位被清除。

图 9. 传输确认寄存器 (CANTA)



图例：RC = 读取/写入；- n = 复位后的值

表 9. 传输确认寄存器 (CANTA) 字段说明

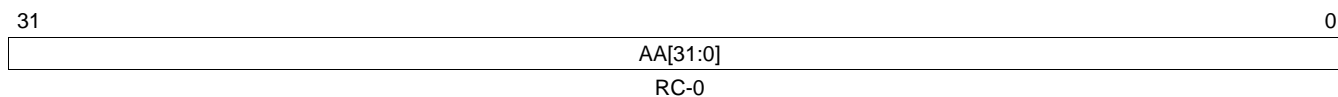
位	字段	值	说明
31:0	TA[31:0]	1	发送确认位
		0	如果 n 邮箱的信息被成功发送，则该寄存器的 n 位被置位。 消息未发送。

6 中断确认寄存器 (CANAA)

如果 n 邮箱中的信息传输被中断，则 AA [n] 位和 AAIF (GIF.14) 被置位，如果被启用，就可能会产生中断。

CANAA 内的位由 CPU 中写入一个 1 来复位。写入 0 无效。如果 CPU 试图在 CAN 对一个位置位时对其复位，则该位被置位。加电后，所有位被清除。

图 10. 中断确认寄存器 (CANAA)



图例：RC = 读取/写入；- n = 复位后的值

表 10. 中断确认寄存器 (CANAA) 字段说明

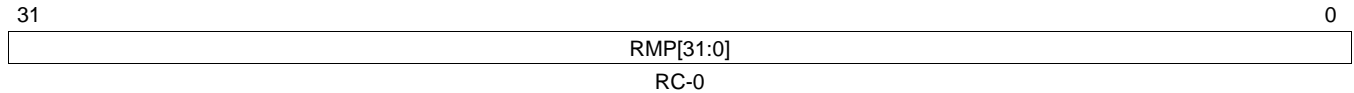
位	字段	值	说明
31:0	AA[31:0]	1	中断确认位
		0	如果邮箱 n 中的消息传输被中止，则该寄存器的 n 位被置位。 传输没被中止。

7 接收消息等待寄存器 (CANRMP)

如果邮箱 n 包含一条接收到的消息，则该寄存器的 $RMPn$ 位被置位。这些位只能由 CPU 和内部逻辑置位。如果 $OPC[M](OPC.31-0)$ 位被清除，则一个新的传入消息会写覆盖已存储的消息，否则下一个邮箱将被检查是否有相匹配的 ID。如果邮箱被写覆盖，相应的状态位 $RML[n]$ 被置位。通过在相应的位位置将 1 写入 CANRMP 寄存器，可清除 CANRMP 和 CANRML 寄存器中的位。如果 CPU 试图在 CAN 对一个位置位时对其复位，则该位被置位。

如果在 CANMIM 寄存器中的相应的中断屏蔽位被置位，CANRMP 寄存器中的位可以设定 $GMIF0/GMIF1(GIF0.15/GIF1.15)$ 。GMIF0/GMIF1 位发起一个中断。

图 11. 接收消息等待寄存器 (CANRMP)



图例：RC = 读取/写入；- n = 复位后的值

表 11. 接收消息挂等待寄存器 (CANRMP) 字段说明

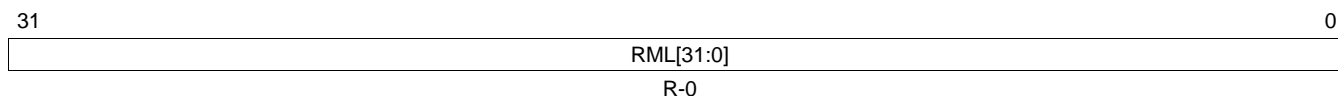
位	字段	值	说明
31:0	RMP[31:0]	1	接收-消息-等待位
		0	如果邮箱 n 包含一条接收到的消息，则该寄存器的 $RMPn$ 位被置位。 邮箱不包含消息。

8 接收信息丢失寄存器 (CANRML)

如果在邮箱 n 中的一条旧信息被新信息写覆盖，则一个 RML_n 位被置位。这些位只能由 CPU 或者内部逻辑置位。通过在相应位位置写入一个 1 到 CANRMP 寄存器，可清除这些位。如果 CPU 试图在 CAN 对一个位置位时对其复位，则该位被置位。如果 OPC_n (OPC.31-0) 位被置位，则 CANRML 寄存器不被更改。

如果一个或多个 CANRMI 寄存器被置位，则 RMLIF (GIF0.11/GIF1.11) 位也被置位。RMLIM (GIM.11) 位被置位会发起一个中断。

图 12. 接收信息丢失寄存器 (CANRML)



图例：R = 读取；-n = 复位后的值

表 12. 接收信息丢失寄存器 (CANRML) 字段说明

位	字段	值	说明
31:0	RML[31:0]	1	接收信息丢失位
		0	邮箱中有一条旧的未读消息被新消息写覆盖。
			没有信息丢失。
			请注意：通过清除设置 RMP_n 位， RML_n 位被清除。

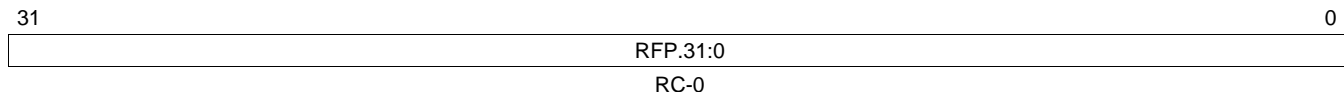
9 远程帧等待寄存器 (CANRFP)

每当 CAN 模块收到一个远程帧请求，远程帧等待寄存器中的相应位 RFP[M]被置位。如果一个远程帧被存储在一个接收邮箱 (AAM=0, CANMD=1)，则 RFPn位不会被置位。

为了防止自动应答邮箱回复一个远程帧请求，CPU 必须通过设置相应的传输的要求复位 TRR [n] 位来清除 RFP[n] 标志和 TRS[n]位。AAM 位也可以通过 CPU 清除，以便停止模块发送消息。

如果 CPU 试图在 CAN 模块对一个位置位的同时对其复位，则该位不会被置位。CPU 不能中断一个正在进行的传输。

图 13. 远程帧等待寄存器 (CANRFP)



图例：RC = 读取/写入；-n = 复位后的值

表 13. 远程帧等待寄存器 (CANRFP) 字段说明

位	字段	值	说明
31:0	RFP.31:0		远程-帧-等待寄存器。 对于一个接收邮箱，如果一个远程帧被接收并且 TRSn不受影响的话，RFTn被设置。 对于一个传送邮箱，如果一个远程帧被接收并且 TRSn被设置且邮箱的 AAM 是 1 的话，RFTn被设置。该邮箱 ID 必须和远程帧的 ID 相匹配。
		1	该模块接收远程帧请求。
		0	没有收到远程帧请求。该寄存器被 CPU 清除。

9.1 远程帧的处理

如果收到一个远程帧（传入的消息将 RTR(MSGCTRL.4)=1），使用适当的从最高邮箱编号开始的降序屏蔽，CAN 模块将此标识符与邮箱的所有标识符相比较。

如果标识符匹配（消息对象被配置为发送邮箱并且消息对象中的 AAM (MSGID.29) 被置位），这个标识符被标记为将被发送（TRS[n]被置位）。

如果与标识符匹配的邮箱被配置为发送邮箱并且该邮箱的 AAM 位未设置，则该邮箱不会接收到这条消息。在发送邮箱内找到匹配标识符后，将不再继续比较。

借助于匹配标识符和被配置为接收邮箱的消息对象，会像处理一个数据帧一样处理这条消息并且接收消息等待 (CANRMP) 寄存器中相应位也被设置。然后，CPU 必须决定如何处理这种情况。更多关于 CANRMP 寄存器信息，请参阅7节。

为了使 CPU 能够改变被配置为一个远程帧邮箱 (AAM 设置) 的邮箱内的数据，它必须先要设置邮箱编号并且改变 MCR 中的数据请求位 (CDR [MC.8])。然后，CPU 可以进行存取并清除 CDR 位以此告知 eCAN 存取已经完成。直到 CDR 位被清除后，该邮箱才能进行传输。从而，最新数据被发送。

要改变那个邮箱的标识符，必须先将邮箱禁用 (CANME_n=0)。

为了使 CPU 能从其他节点请求数据，它必须先将邮箱配置为接收邮箱并设置 TRS 位。这种情况下，该模块发送一个远程数据请求并在发出请求的同一个邮箱内接收数据帧。因此，远程请求只需一个邮箱。请注意，CPU 必须设置 RTR (MSGCTRL.4) 才能启动远程帧传输。一旦一个远程帧被发出，则邮箱的 TRS 位被 CAN 清除。在这种情况下，位 TA 位_n将为该邮箱进行设置。

消息对象_n的运行状态由 CANMD_n(CANMD.31-0), AAM (MSGID.29)及 RTR (MSGCTRL.4) 一起配置。这显示了如何根据所需运行状态来配置消息对象。

总的来说，一个消息对象可以被配置有四种不同运行状态：

1. 传输消息对象只能传输消息。
2. 接收消息对象只能接收消息。
3. 请求消息对象可以传输远程帧并且等待相应的数据帧。
4. 回复消息对象能在相应的标识符收到远程请求帧时传输数据帧。

注： 当一个被配置为请求模式的消息对象与一个传输远程数据请求一起成功发送时，CANTA 寄存器不再被设置并且不再产生中断。当收到远程回复消息时，该消息的运行状态将和被配置为接收模式的消息对象的运行状态相同。

10 全局接受屏蔽寄存器 (CANGAM)

全局-接受屏蔽由处于 SCC 模式下的 eCAN 使用。如果相应邮箱的 AME 位 (MSGID.30) 被设置, 则全局接受屏蔽被用于邮箱 6 到 15。接收到的消息仅储存在第一个具有匹配标识符的邮箱中。

全局-接受屏蔽由 SCC 的邮箱 6 到 15 使用。

图 14. 全局接受屏蔽寄存器 (CANGAM)

31	30	29	28	16
AMI	被保留			GAM[28:16]
RWI-0	R-0			RWI-0
15				0
GAM[28:16]				
RWI-0				

图例: RWI = 在任何时候读取, 只能在初始化模式期间写入; - n = 复位后的值

表 14. 全局接受屏蔽寄存器 (CANGAM) 字段说明

位	字段	值	说明
31	AMI	1	接受屏蔽标识符扩展位 可以接收标准和扩展帧。在扩展帧的情况下, 标识符的所有 29 位被存储在邮箱中, 全局接受屏蔽寄存器的所有 29 位被过滤器使用。在一个标准帧的情况下, 只有头 11 个位 (28 至 18 位) 的标识符和全局接受屏蔽被使用。 接收邮箱的 IDE 位是一个“无关”并且由被发送消息的 IDE 位写覆盖。必须满足筛选标准, 以便收到一条消息。要进行比较的位数是发送的消息的 IDE 位值的函数。
		0	存储在邮箱中的标识符扩展位确定了哪些消息应该被收到。接收邮箱的 IDE 位决定了要进行比较的位数。滤波是不适用的。为了接收消息, MSGID 必须位位匹配。
30:29	被保留		读取未定义, 写入无效。
28:0	GAM 28:0		全局接受屏蔽。这些位允许屏蔽一个传入消息的任何标识符位。接收到的标识符的相应位接受的一个 0 或一个 1 (无关)。收到的标识符位值必须与 MSGID 寄存器相应的标识符位相匹配。

11 主控制寄存器 (CANMC)

该寄存器用于控制 CAN 模块的设置。CANMC 寄存器的一些位受 EALLOW 保护。对于读取/写入操作，只支持 32 位的访问。

图 15. 主控制寄存器 (CANMC)

31															17		16
被保留															SUSP		
R-0															R/W-0		
15			14		13		12		11		10		9		8		
MBCC			TCC		SCB		CCR		PDR		DBO		WUBA		CDR		
R/WP-0			SP-x		R/WP-0		R/WP-1		R/WP-0		R/WP-0		R/WP-0		R/WP-0		
7			6		5		4								0		
ABO			STM		SRES						MBNR						
R/WP-0			R/WP-0		R/S-0						R/W-0						

图例：R = 读取，WP = 只在 EALLOW 模式时写入，S = 仅在 EALLOW 模式中设置；-n = 复位后的值；X = 不确定
 请注意：只适用于 eCAN，被保留在 SCC 中

表 15. 主控制寄存器 (CANMC) 字段说明

位	字段	值	说明
31:17	被保留		读取未定义，写入无效。
16	SUSP		SUSPEND（等待）。该位确定 SUSPEND 中 CAN 模块的运行（如断点或单步进仿真停止）。
		1	FREE（自由）模式。外设继续在 SUSPEND 模式中运行。在 SUSPEND 模式时，该节点将参与 CAN 正常通信（发送确认、产生错误帧、发送/接收数据）。
		0	SOFT（软）模式。在 SUSPEND 模式期间，当前传输完成后，外设关闭。
15	MBCC		邮箱时间戳计数器清除位。该位保留在 SCC 模式并受 EALLOW 保护。
		1	在 16 邮箱传输或接收成功后，时间戳计数器复位为 0。
		0	时间戳计数器不复位。
14	TCC		时间戳计数器 MSB 清除位。该位保留在 SCC 模式并受 EALLOW 保护。
		1	时间戳计数器的 MSB 复位为 0。在内在逻辑的一个时钟周期后，TCC 位复位。
		0	时间戳计数器没被更改。
13	SCB		SCC 兼容性位。该位保留在 SCC 模式并受 EALLOW 保护。
		1	选择 eCAN 模式。
		0	eCAN 处于 SCC 模式。只有邮箱 0 到 15 可以使用。
12	CCR		更改配置请求。此位受 EALLOW 保护。
		1	CPU 请求到配置寄存器 CANBTC 的写入访问和 SCC 的接收屏蔽寄存器（CANGAM，LAM[0] 和 LAM [3]）。该位置位后，在进入 CANBTC 寄存器之前，CPU 必须等待，直到 CANES 寄存器的 CCE 标志为 1。
		0	如果 ABO 位没被置位，在总线关闭状态时，CCR 位也将被置位。BO 状态可以通过清除该位（在总线上的 128 *11 连续隐性位之后）退出。 CPU 请求正常运作。这只有在配置寄存器 CANBTC 被设定到允许值后完成。在强制总线关闭恢复序列后，它还会退出总线关闭状态。
11	PDR		断电模式请求。在从低功耗模式唤醒后，该位被 eCAN 模块自动清除。该位受 EALLOW 保护。
		1	请求本地断电模式。
		0	没有请求局部断电模式（正常操作）。
			请注意： 如果一个应用程序为一个邮箱将 TRSn 位置位，然后立即设定 PDR 位，CAN 模块在不传送数据帧的情况下进入 LPM。这是因为从 RAM 邮箱被转移到发送缓冲区的数据大约需要 80 个 CPU 周期。因此，应用必须确保在写入 PDR 位之前就已经完成所有等待的传输。TAn 位可被轮询，以确保传输的完成。
10	DBO		数据字节顺序。该位选择消息数据字段的字节顺序。此位受 EALLOW 保护。
		1	首先接收或传输数据的最低有效位。
		0	首先接收或传输数据的最高有效位。

表 15. 主控制寄存器 (CANMC) 字段说明 (continued)

位	字段	值	说明
9	WUBA	1	在检测到任何总线活动时唤醒。此位受 EALLOW 保护。
		0	只有在 PDR 位上写入一个 0 之后, 该模块才离开断电模式。
8	CDR	1	更改数据字段请求。该位允许快速数据消息更新。 CPU 请求到 MBNR.4 字段 (MC.4-0) 指定的邮箱数据字段的写入访问。访问邮箱后, CPU 必须清除 CDR 位。在 CDR 置位时, 该模块不发送该邮箱的内容。在它从邮箱中读取数据的前后由状态机对它进行检查来将它存储在发送缓冲器中。 请注意: 一旦为一个邮箱设置 TRS 位, 之后通过使用 CDR 位更改邮箱中的数据, CAN 模块不能传输新数据而是只能传输旧数据。为了避免这种情况, 通过使用 TRRn 位来在该邮箱复位传输, 并且再次设置 TRSn 位。然后传送新数据。
		0	CPU 请求正常运行。
7	ABO	1	自动总线打开。此位受 EALLOW 保护。
		0	总线关闭状态后, 在监测到 128*11 个隐性位后, 该模块自动进入总线打开状态。 清除 CCR 位后, 只在总线上连续出现 128*11 个隐性位后退出总线关闭状态。
6	STM	1	自检模式。该位受 EALLOW 保护。 该模块处于自检模式。在这种模式下, CAN 模块产生自应答 (ACK) 信号, 从而在没有总线连接到模块的情况下启用运行。消息未发送, 但回读并存储在相应的邮箱中。在 STM 中, 接收帧的 MSGID 没有存储在 MBR。 请注意: 在 STM 中, 如果没有 MBX 被配置用来接收一个传输帧, 那么该帧将会存储在 MBX0 中, 尽管 MBX0 也没有被配置用来接收操作。如果 LAM 被配置, 这样一些邮箱能够接收和存储数据帧, 那么针对任一接收邮箱, 一个不满足接受屏蔽过滤标准的数据帧将丢失。
		0	该模块处于正常模式。
5	SRES	1	该位只能写入并一直读作零。 写入访问此寄存器会引起模块的软件复位 (除受保护寄存器之外的所有参数, 都被复位为其缺省值)。邮箱的内容和错误计数器不会被修改。在不干扰通信的情况下, 等待和正在进行的传输被取消。
		0	0 无影响
4:0	MBNR 4:0	1	邮箱数 MBNR.4 位仅用于 eCAN, 并被保留在 SCC 中。
		0	邮箱编号, CPU 请求一个到这个数据字段的写入权限。该字段与 CDR 位结合使用。

11.1 CAN 模块在 SUSPEND (中止) 时工作

1. 如果 CAN 总线空闲并且 SUSPEND 模式发出请求, 那么节点转入 SUSPEND 模式。
2. 如果 CAN 总线不空闲并且 SUSPEND 模式发出请求, 那么节点在正在进行的帧传输结束后转入 SUSPEND 状态。
3. 如果节点正在传输, SUSPEND 被请求时, 那么节点在得到应答后转入 SUSPEND 状态。如果节点没有得到应答或出现其他错误, 那么节点在发送一个错误帧后转入 SUSPEND 状态。对 TEC 做出相应地修改。第二种情况, 即, 节点在发送错误帧后中止, 则节点在解除中止状态后, 重新传输原来的帧。传输相应的帧后 TEC 被修改。
4. 如果节点正在接收时, 当 SUSPEND 被请求时, 它将在发出确认位后转入 SUSPEND 状态。如果出现任何错误, 节点发送一个错误帧后转入 SUSPEND 状态。进入 SUSPEND 状态前对 REC 进行相应修改。
5. 如果 CAN 总线空闲并且 SUSPEND 去除被请求, 那么节点脱离 SUSPEND 状态。
6. 如果 CAN 总线不空闲并且 SUSPEND 去除被请求, 那么节点在总线进入空闲状态后脱离 SUSPEND 状态。因此, 节点不接收任何会产生错误帧的“部分”帧。
7. 节点中止时, 它不参与传输或接收的任何数据。因此, 既不会有确认位发送也不会有错误帧发送。在 SUSPEND 状态期间, TEC 和 REC 都不会被修改。

12 位-时序配置寄存器 (CANBTC)

CANBTC 寄存器用于配置 CAN 节点适当的网络时序参数。在使用 CAN 模块前，必须编辑该寄存器。

该寄存器在用户模式中为写保护并且只能在初始化模式时写入（参见 3.6.1 节）。

注：为避免 CAN 模块不可预知的运行状态，CANBTC 寄存器绝不能使用 CAN 协议技术规范和 3.1.1 节中列出的位时序规则允许之外的值进行编辑。

图 16. 位-时序配置寄存器 (CANBTC)

31	24	23	16
被保留			BRP _{reg}
R-x			RWPI-0
15	10	9	8
被保留	SJW _{reg}	SAM	TSEG1 _{reg}
R-0	RWPI-0	RWPI-0	RWPI-0
		0	0

图例：RWPI = 在所有模式读取，只能在初始化模式期间的 EALLOW 模式中写入；-n=复位后的值

表 16. 位时序配置寄存器 (CANBTC) 字段描述

位	字段	值	说明
31:24	被保留		读取未定义，写入无效。
23:16	BRP _{reg}		<p>波特率分频器。该寄存器用于设置波特率设置的预分频器。一个 TQ 的长度被定义为：</p> $TQ = \frac{1}{SYSCLKOUT/2} \times (BRP_{reg} + 1)$ <p>在这里 SYSCLKOUT/2 是 CAN 模块时钟的频率。</p> <p>BRP_{reg} 表示预分频器“寄存器的值”；即写入 CANBTC 寄存器 23:16 位的值。CAN 模块访问它时，这个值会自动增加 1。增加值由符号 BRP(BRP=BRP_{reg}+1) 表示。BRP 可在 1 至 256 内增加。</p> <p>注意：对于 BRP=1 的特殊情况，信息处理时间 (IPT) 等于 3 时间份额 (TQ)。这不符合 ISO11898 标准，该标准规定 IPT 要小于或等于 2TQ。因此，这种模式的用法 (BRP_{reg}=0) 是不被允许的。</p>
15	被保留		读取未定义，写入无效。
9:8	SJW _{reg} 1:0		<p>同步跳转宽度。SJW 参数表明，进行重新同步时，一个位中允许的加长或者缩短的 TQ 单位数量。</p> <p>SJW_{reg} 表示“同步跳转宽度”的“寄存器的值”；即写入 CANBTC 寄存器 9:8 位的值。CAN 模块访问它时，这个值会自动增加 1。增加值由符号 SJW 表示。</p> $SJW = SJW_{reg} + 1$ <p>SJW 可在 1TQ 到 4TQ 内设定。SJW 的最大值由 TSEG2 和 4TQ 的最大值决定。</p> $SJW_{(最大值)} = \text{最小值} [4TQ, TSEG2]$
7	SAM		<p>此参数设定了 CAN 模块所使用的样本数量来确定 CAN 总线的实际电平。当 SAM 位设定后，根据最后三个值的多数决定结果，由 CAN 总线确定电平。采样点设在同一点和两倍于 1/2 TQ 的距离前的地方。</p> <p>1 CAN 模块三次取样后进行多数决定。只有当比特率预分频值大于 4 时 (BRP>4) 选择三重采样本模式。</p> <p>0 在采样点上 CAN 模块只能采样一次。</p>

表 16. 位时序配置寄存器 (CANBTC) 字段描述 (continued)

位	字段	值	说明
6:3	TSEG1 3:0		时段 1。CAN 总线上位长取决于 TSEG1, TSEG2 和 BRP 参数。CAN 总线上的所有控制器必须有相同的比特率和位长。对于单个控制器的不同的时钟频率, 波特率由上述参数进行调整。
			此参数指定了 TQ 单位中 TSEG1 段的长度。TSEG1 将 PROP_SEG 和 PHASE_SEG1 段组合在一起:
			$TSEG1 = PROP_SEG + PHASE_SEG1$
			在这里, PROP_SEG 和 PHASE_SEG1 是在 TQ 单位内这两个区段的长度。
			TSEG1 _{reg} 表示时段 1 的“寄存器的值”, 即写入 CANBTC 寄存器 6:3 位的值。CAN 模块访问它时, 这个值会自动增加 1。增加值由符号 TSEG1 表示。
			$TSEG1 = TSEG1_{reg} + 1$
2:0	TSEG2 _{reg}		时段 2。TSEG2 在 TQ 单位内定义了 PHASE_SEG2 的长度: TSEG2 可在 1TQ 至 8TQ 内设定, 同时遵循下列时序规则: TSEG2 必须小于或等于 TSEG1 同时必须大于或等于 IPT。
			TSEG2 _{reg} 表示“时段 2”的“寄存器的值”, 即写入 CANBTC 寄存器 2:0 位的值。当 CAN 模块访问它时, 这个值会自动增加 1。增加值由符号 TSEG2 表示。 $TSEG2 = TSEG2_{reg} + 1$

13 错误和状态寄存器 (CANES)

CAN 模块的状态由错误和状态寄存器 (CANES) 和错误计数器寄存器显示, 这些在本节中进行描述。

错误和状态寄存器包含 CAN 模块的实际状态的信息并且显示总线错误标志以及错误状态标志。如果这些错误标志的一个被设定, 那么所有其他的错误标志的当前状态被冻结, 只有第一个错误被存储。为了更新随后的 CANES 寄存器, 被设定的错误标志必须通过将 1 写入到寄存器中进行确认。这个动作也会清除标志位。

图 17. 错误和状态寄存器 (CANES)

31		25	24	23	22	21	20	19	18	17	16
	被保留		FE	BE	SA1	CRCE	SE	ACKE	BO	EP	EW
	R-0		RC-0	RC-0	R-1	RC-0	RC-0	RC-0	RC-0	RC-0	RC-0
15					6	5	4	3	2	1	0
	被保留					SMA	CCE	PDA	Rsvd	RM	TM
	R-0					R-0	R-1	R-0	R-0	R-0	R-0

图例: R = 读取; C = 清除; -n = 复位后的值

表 17. 错误和状态寄存器 (CANES) 字段说明

位	字段	值	说明
31:25	被保留		读取未定义, 写入无效。
24	FE	1 0	格式错误标志 发生在总线上的一个格式错误。这意味着, 总线上的一个或多个固定格式的位字段有错误电平。 没有检测到任何格式错误; CAN 模块能够正确发送和接收。
23	BE	1 0	位错误标志 接收到的位与仲裁域以外被发送为不匹配或者在仲裁域传输期间, 发送的是一个显性位, 却收到了一个隐性位。 没有检测到位错误。
22	SA1	1 0	停留在显性错误。在硬件复位、软件复位、或总线关闭状态后, SA1 位一直为 1。当在总线上检测到隐性位时, 该位被清除。 CAN 模块从未检测到一个隐性位。 CAN 模块检测到一个隐性位。
21	循环冗余码校验错误 (CRCE)	1 0	CRC 错误。 CAN 模块收到一个 CRC 错误。 CAN 模块从未收到一个 CRC 错误。
20	SE	1 0	填充错误。 发生一个填充位错误。 未发生填充位错误。
19	ACKE	1 0	确认错误。 CAN 模块没有收到确认。 所有消息已收到正确确认。
18	BO	1 0	总线关闭状态。CAN 模块处于总线关闭状态。 CAN 总线上存在一个异常错误率。当发送错误计数器 (CANTEC) 已达到 256 个的限值时, 会出现这种情况。总线关闭期间, 不能接收或发送任何消息。总线关闭状态通过清除 CANMC 寄存器中 CCR 位或者如果自动总线打开 (ABO) (CANMC.7) 位被设定, 在收到 128*11 接收位后退出。离开总线关闭状态后, 错误计数器都被清除。 正常运行
17	EP	1 0	错误被动状态 CAN 模块处于错误被动模式。CANTEC 已达到 128。 CAN 模块处于错误主动模式。

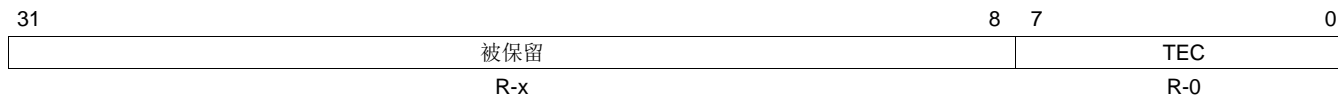
表 17. 错误和状态寄存器 (CANES) 字段说明 (continued)

位	字段	值	说明
16	EW	1	警告状态 两个错误计数器 (CANREC 或 CANTEC) 之一已达到 96 级警告级别。
		0	两个错误计数器 (CANREC 或 CANTEC) 值均低于 96。
15:6	被保留		读取未定义, 写入无效。
5	SMA	1	等待模式确认。在等待模式被激活后, 一个时钟周期延迟达到一个帧的长度后, 该位被置位。当电路不在运行模式时, 用调试工具激活等待模式。在等待模式期间, CAN 模块将被冻结, 无法接收或发送任何帧。然而, 如果在等待模式被激活时, CAN 模块正在传输或接收一个帧, 则模块只在帧的末尾进入等待模式, 运行模式就是当 SOFT (软模式) 被激活(CANMC.16=1) 时的模式。
		0	该模块已进入等待模式。 该模块不在等待模式。
4	CCE	1	更改配置使能。该位显示了配置的访问权。经过一个时钟周期的延迟, 该位被置位。 CPU 具有到配置寄存器的写入访问权限。
		0	CPU 被拒绝到配置寄存器的写入访问。 请注意: CCE 位的复位状态为 1。也就是说, 复位时, 您可以写入位时序寄存器。然而, 一旦 CCE 位被清除 (作为模块初始化的一部分), 在您可以再次把 CCE 位置位为 1 前, CANRX 引脚必须检测到高电平。
3	PDA	1	断电模式确认 CAN 模块已经进入断电模式。
		0	正常运行
2	被保留		读取未定义, 写入无效。
1	RM	1	接收模式。CAN 模块处于接收模式。不管邮箱配置如何, 该位反映了 CAN 模块实际运行状态。 CAN 模块正在接收消息。
		0	CAN 模块没有接收消息。
0	TM	1	发送模式。CAN 模块处于发送模式。不管邮箱配置如何, 该位反映了 CAN 模块实际运行状态。 CAN 模块正在发送消息。
		0	CAN 模块没有发送消息。

14 CAN 错误计数寄存器 (CANTEC/CANREC)

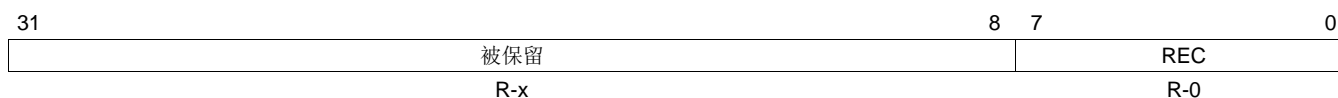
CAN 模块包含两个错误计数器：接收错误计数器(CANREC) 和发送错误计数器 (CANTEC)。可以通过 CPU 接口读取这两个计数器的值。根据 CAN 协议技术规范 2.0 版本，这些计数器是递增或递减的。

图 18. 发送错误计数器寄存器 (CANTEC)



图例：R = 只读取；-n=复位后的值

图 19. 接收错误计数器寄存器 (CANREC)



图例：R = 只读取；-n=复位后的值

接收错误计数器达到或超过了错误被动的上限值 (128) 后不会再增加。当一个消息被正确接收时，计数器被重新置位到 119 和 127 (与 CAN 技术规范相比较) 之间的值。

在达到总线关闭状态后，发送错误计数器未定义，而接收错误计数器则改变其功能。在达到总线关闭状态后，接收错误计数器被清除。在总线上的每 11 个是连续隐性位后，随后它被递增。这些 11 个位与在总线上的两帧之间的差相对应。如果计数器达到 128 且此功能被使能 (自动总线打开位 (ABO) (MC.7) 置位)，则模块会自动变回在总线开状态。所有内部标志被复位，并且错误计数器被清除。离开初始化模式后，错误计数器都被清除。

15 中断寄存器

中断由中断标志寄存器、中断屏蔽寄存器和邮箱中断级别寄存器控制。在以下小节中对这些寄存器进行了描述。

15.1 全局中断标志寄存器 (CANGIF0/CANGIF1)

这些寄存器允许 CPU 识别中断源。

如果确实发生了相应的中断状况，则中断标志位被置位。全局中断标志的置位取决于 GIL 位在 CANGIM 寄存器中的设置。如果该位被置位，全局中断对在 CANGIF1 寄存器中的位进行置位；否则，在 CANGIF0 寄存器中进行置位。这也适用于 AAIF 和 RMLIF 中断标志。这些位将按照在 CANGIM 寄存器位中适当 GIL 位的设置被置位。

不管 CANGIM 寄存器中相应的中断屏蔽位是否被设置，以下位被置

位：MTOFn, WDIFn, BOIFn, TCOFn, WUIFn, EPIFn, AAIFn, RMLIFn, 和 WLIFn。

对于任何邮箱来说，只有当相应的邮箱中断屏蔽位（在 CANMIM 寄存器）被置位时，GMIFn 位才会被置位。

如果所有的中断标志位清零，并且一个新的中断标志被置位，当相应的中断屏蔽位被置位时，中断输出线被激活。中断线持续保持活跃直至 CPU 通过把 1 写入适当位或清除引起中断条件来清除中断标志。

必须通过在 CANTA 寄存器或 CANRMP 寄存器（取决于邮箱配置）的适当位上写入 1 来清除 GMIFx 标志，且不能在 CANGIFx 寄存器中清除。在清除一个或多个中断标志且一个或多个中断标志仍然被设定，会产生新的中断。通过写 1 到适当位的位置来清除中断标志。如果 GMIFx 被设定，邮箱中断向量 MIVx 表示引起 GMIFx 设置的邮箱数量。在多个邮箱中断被等待的情况下，它总是显示分配给该中断线路的最高邮箱中断矢量。

图 20. 全局中断标志寄存器 0 (CANGIF0)

31								24									
被保留																	
R-x																	
23												18		17		16	
被保留												MTOF0		TCOF0			
R-x												R-0		RC-0			
15		14		13		12		11		10		9		8			
GMIF0		AAIF0		WDIF0		WUIF0		RMLIF0		BOIF0		EPIF0		WLIF0			
R/W-0		R-0		RC-0		RC-0		R-0		RC-0		RC-0		RC-0			
7				5		4		3		2		1		0			
被保留				MIV0.4		MIV0.3		MIV0.2		MIV0.1		MIV0.0					
R/W-0				R-0		R-0		R-0		R-0		R-0					

图例: R/W = 读取/写入; R = 只读; C = 清除; -n = 复位后的值

图 21. 全局中断标志寄存器 1 (CANGIF1)

31								24									
被保留																	
R-x																	
23												18		17		16	
被保留												MTOF1		TCOF1			
R-x												R-0		RC-0			
15		14		13		12		11		10		9		8			
GMIF1		AAIF1		WDIF1		WUIF1		RMLIF1		BOIF1		EPIF1		WLIF1			
R/W-0		R-0		RC-0		RC-0		R-0		RC-0		RC-0		RC-0			
7				5		4		3		2		1		0			
被保留				MIV0.4		MIV0.3		MIV0.2		MIV0.1		MIV0.0					
R/W-0				R-0		R-0		R-0		R-0		R-0					

图例: R/W=读取/写入; R = 读取; C = 清除; -n=复位后的值

注意: 只用于 eCAN, 被保留在 SCC 中。

注: 以下位说明同时适用于 CANGIF0 和 CANGIF1 寄存器。以下中断标志, 由在 CANGIM 寄存器位中的 GIL 位值来确定它们是否在 CANGIF0 或 CANGIF1 寄存器内被设置: TCOFn, AAIFn, WDIFn, WUIFn, RMLIFn, BOIFn, EPIFn, 和 WLIFn。

如果 GIL=0, 这些标志设置在 CANGIF0 寄存器中; 如果 GIL=1, 这些标志设置在 CANGIF1 寄存器中。

相似地, CANGIF0 和 CANGIF1 寄存器对于 MTOFn 和 GMIFn 位的选择取决于 CANMIL 寄存器的 MILn 位。

表 18. 全局中断标志寄存器 (CANGIF0/CANGIF1) 字段说明

位	字段	值	说明
31:18	被保留		被保留。读取未定义, 写入无效。
17	MTOF1/1	1	邮箱超时标志。在 SCC 模式下此位不可用。
		0	其中的一个邮箱没有在指定的时间帧内传输或接收消息。 邮箱无超时。
			注意: MTOFn 位是否被设置在 CANGIF0 或 CANGIF1 中取决于 MIL 值 n。TOSn 被清除时, MTOFn 也被清除。TOSn 位在成功发送/接收后 (最终) 将被清除。

表 18. 全局中断标志寄存器 (CANGIF0/CANGIF1) 字段说明 (continued)

位	字段	值	说明
16	TCOF1/1	1 0	时间戳计数器溢出标志位。 时间戳计数器的 MSB 从 0 变为 1。 时间戳计数器的 MSB 为 0。即，它没有从 0 变为 1。
15	GMIF1/1	1 0	全局邮箱中断标志。只有当在 CANMIM 寄存器中，相应的邮箱中断屏蔽位被设置时，才设置此位。 其中一个邮箱成功发送或接收到一个消息。 没有成功发送或接收到消息。
14	AAIF1/1	1 0	中止确认的中断标志 发送传输请求已中止。 无传输已经被中止。 请注意： 通过清除设置 AAN 位，AAIFn 位被清零。
13	WDIF0/WDIF1	1 0	写入被拒绝的中断标志 CPU 的写入访问邮箱不成功。当邮箱标识符字段被写入时，WDIF 中断被置为有效的同时被启用。在写入 MBX 的 MSGID 字段前，它应该被禁用。如果您在 MBX 仍然是启用时尝试此操作，WDIF 位将被设置并且一个 CAN 中断被置为有效。 CPU 到邮箱的写入访问成功。
12	WUIF0/WUIF1	1 0	唤醒中断标志 局部断电期间，该标志表明该模块已脱离睡眠模式。 该模块仍处于睡眠模式或正常运行
11	RMLIF1/1	1 0	接收-消息-丢失中断标志 至少一个接收邮箱，一个溢出条件已经发生并且 MILn 寄存器的相应位已清零。 无消息丢失。 请注意： 通过清除设置 RMPn 位，RMLIFn 位被清零。
10	BOIF0/BOIF1	1 0	总线关闭中断标志 CAN 模块已经进入总线关闭模式。 CAN 模块仍然是在总线打开模式上。
9	EPIF0/EPIF1	1 0	错误被动中断标志 CAN 模块已经进入错误被动模式。 CAN 模块不在错误被动模式。
8	WLIF0/WLIF1	1 0	警告级别中断标志 至少有一个错误计数器达到错误警告限值。 没有一个错误计数器达到错误警告限值。
7:5	被保留		读取未定义且写入无效。
4:0	MIV0.4:0/MIV1.4:0		邮箱中断矢量。在 SCC 模式中只有 3:0 位可用。 这个向量表示设定全局邮箱中断标志中的邮箱编号。它保持该矢量直到适当 MIFn 位被清除或更高优先级的邮箱中断发生。然后显示最高的中断矢量，邮箱 31 具有最高优先级。在 SCC 模式中，邮箱 15 具有最高优先级。邮箱 16 至 31 未使用。 如果在 TA/RMP 寄存器中没有标志设置，GMIF1 或 GMIF0 也被清除，此值是未定义的。

15.2 全局中断屏蔽寄存器 (CANGIM)

中断屏蔽寄存器的设置和中断标志寄存器相同。如果一个位被设定，相应的中断被启用。此寄存器受 EALLOW 保护。

图 22. 全局中断屏蔽寄存器 (CANGIM)

31											18		17	16		
被保留											MTOM		TCOM			
R-0											R/WP-0		R/WP-0			
											0		0			
15			14		13		12		11		10		9		8	
被保留			AAIM		WDIM		WUIM		RMLIM		BOIM		EPIM		WLIM	
R-0			R/WP-0		R/WP-0		R/WP-0		R/WP-0		R/WP-0		R/WP-0		R/WP-0	
7							3			2		1		0		
保留							GIL			I1EN		IOEN				
R-0							R/WP-0			R/WP-0		R/WP-0				

图例：R = 读取；W = 写入；WP = 只在 EALLOW 模式时写入；-n= 复位后的值

表 19. 全局中断屏蔽寄存器 (CANGIM) 字段说明

位	字段	值	说明
31:18	保留		读取未定义，写入无效。
17	MTOM	1 0	邮箱超时中断屏蔽 被启用 被禁用
16	TCOM	1 0	时间戳计数器溢出屏蔽 被启用 被禁用
15	被保留		读取未定义，写入无效。
14	AAIM	1 0	中止确认中断屏蔽。 被启用 被禁用
13	WDIM	1 0	拒绝写入中断屏蔽 被启用 被禁用
12	WUIM	1 0	唤醒中断屏蔽 被启用 被禁用
11	RMLIM	1 0	接收-消息-丢失中断屏蔽 被启用 被禁用
10	BOIM	1 0	总线关闭中断屏蔽 被启用 被禁用
9	EPIM	1 0	被动错误中断屏蔽 被启用 被禁用
8	WLIM	1 0	警告级别中断屏蔽 被启用 被禁用
7:3	被保留		读取未定义，写入无效。
2	GIL	1 0	TCOF, WDIF, WUIF, BOIF, EPIF, RMLIF, AAIF 和 WLIF 中断的全局中断级别。 1 所有的全局中断被映射到 ECAN1INT 中断线路。 0 所有的全局中断被映射到 ECAN0INT 中断线。

表 19. 全局中断屏蔽寄存器 (CANGIM) 字段说明 (continued)

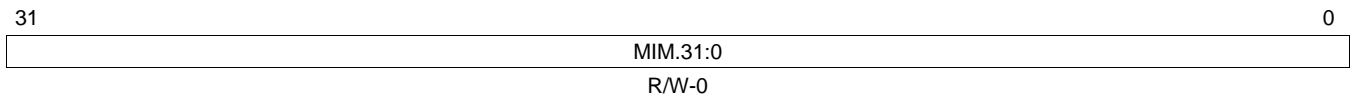
位	字段	值	说明
1	I1EN	1	中断 1 使能 如果设定了相应的屏蔽, 则该位在全局范围内启用针对 ECAN1INT 线路的所有中断。
		0	ECAN1INT 中断线路被禁用。
0	I0EN	1	中断 0 启用 如果设定了相应的屏蔽, 则该位在全局范围内启用针对 ECAN0INT 线路的所有中断。
		0	ECAN0INT 中断线路被禁用。

因为各个邮箱在 CANMIM 寄存器都有独立的屏蔽位, 所以在 CANGIM 中该 GMIF 没有相应的位。

15.3 邮箱中断屏蔽寄存器(CANMIM)

每个邮箱都有一个中断标志。根据邮箱配置，这可以是一个接收或发送中断。此寄存器受 EALLOW 保护。

图 23. 邮箱中断屏蔽寄存器(CANMIM)



图例：R/W = 读取/写入；-n= 复位后的值

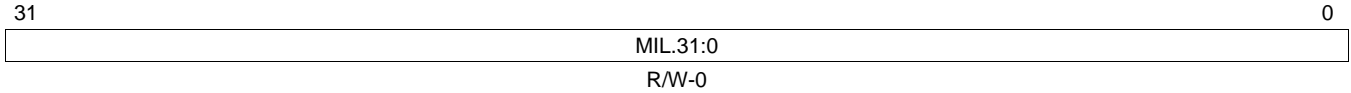
表 20. 邮箱中断屏蔽寄存器（CANMIM）字段说明

位	字段	值	说明
31:0	MIM.31:0	1	邮箱中断屏蔽。加电后，所有中断屏蔽位被清除，且中断被禁用。这些位允许任何邮箱中断被单独屏蔽。
		0	邮箱中断被启用。如果消息已成功发送（在一个发送邮箱的情况下），或者如果在没有收到任何错误的情况下消息已经被接收（在一个接收邮箱的情况下），会产生中断。
		0	邮箱中断被启用。

15.4 邮箱中断级别寄存器 (CANMIL)

32 个邮箱中的任一个都可能在两个中断线路之一引发中断。根据邮箱中断级别寄存器 (CANMIL) 的设置, 中断一般会发生在 ECAN0INT (MIL $n=0$) 或 ECAN1INT 线路上 (MIL $[n]=1$)。

图 24. 邮箱中断级别寄存器 (CANMIL)



图例: R/W = 读取/写入; -n = 复位后的值

表 21. 邮箱中断级别寄存器 (CANMIL) 字段说明

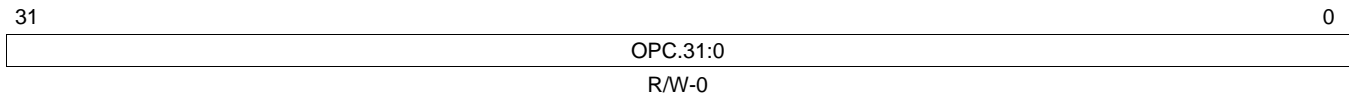
位	字段	值	说明
31:0	MIL.31:0		邮箱中断级别。这些位允许单独选择任何邮箱的中断级别。
		1	邮箱中断产生在中断线路 1 上。
		0	邮箱中断产生在中断线路 0 上。

16 写覆盖保护控制寄存器 (CANOPC)

如果邮箱 n (RMP[n]被置位为 1, 且一个新的接收消息适合 n 邮箱) 出现溢出情况, 则新的信息存储取决于 CANOPC 寄存器中的设置。如果相应的 OPC 位[n]被设置为 1, 旧消息就被保护不被新消息写覆盖; 这样一来, 下个邮箱就会被检查与之相匹配的 ID。如果没有发现其他邮箱, 在没有进一步通知的情况下消息丢失。如果 OPC 位 n 被清除为 0, 旧消息就会被新消息写覆盖。这通过设置接收消息丢失位 RML[n]来通知。

对于读取/写入操作, 只支持 32 位访问。

图 25. 写覆盖保护控制寄存器 (CANOPC)



说明: R/W = 读取/写入; - n = 复位后的值

表 22. 写覆盖保护控制寄存器 (CANOPC) 字段说明

位	字段	值	说明
31:0	OPC.31:0	1	写覆盖保护控制位
		0	1 如果 OPC 位[n]被设置为 1, 存储在该邮箱中的旧消息就被保护不被新消息写覆盖。 0 如果 OPC 位 n 未被设定, 旧消息就会被新消息写覆盖。

17 eCAN I/O 控制寄存器 (CANTIOC, CANRIOC)

CANTX 和 CANRX 引脚应被配置以使用于 CAN。这是通过使用 CANTIOC 和 CANRIOC 寄存器来完成的。

图 26. TX I/O 控制寄存器 (CANTIOC)



图例：RWP = 在所有模式中读取，只在 EALLOW 模式时写入；R= 只读取；-n = 复位后的值

表 23. TX I/O 控制寄存器 (CANTIOC) 字段说明

位	字段	值	说明
31:4	被保留		读取未定义，写入无效。
3	TXFUNC	1 0	该位必须为使用 CAN 模块功能而设定。 CANTX 引脚被用于 CAN 传输功能。 被保留
2:0	被保留		被保留

图 27. RX I/O 控制寄存器 (CANRIOC)



图例: RWP = 在所有模式中读取, 只在 EALLOW 模式时写入; R= 只读取; -n = 复位后的值; X = 不确定

表 24. RX I/O 控制寄存器 (CANRIOC) 字段说明

位	字段	值	说明
31:4	被保留		读取未定义, 写入无效。
3	RXFUNC	1 0	该位必须为使用 CAN 模块功能而设定。 CANTX 引脚被用于 CAN 接收功能。 被保留
2:0	被保留		被保留

18 定时器管理单元

当消息时发送/接收时，几个功能在 eCAN 中被执行以监测时间。eCAN 包含一个独立的状态机来处理时间控制功能。当访问寄存器时，这个状态机的优先级低于 CAN 状态机。因此，其他正在进行的活动可能会使时间控制功能延迟。

18.1 时间戳功能

为了得到一个接收或传送消息的时间指示，一个自由运行的 32 位定时器 (TSC) 在模块中被执行。当存储一个收到的消息或一个消息已被发出时，其内容就被写入到相应的邮箱中的时间戳寄存器中（消息对象的时间戳 [MOTS]）。

计数器由 CAN 总线线路的位时钟驱动。在初始化模式期间，或者如果模块处于睡眠或等待模式时定时器将停止。加电复位后，自由运行的计数器清零。

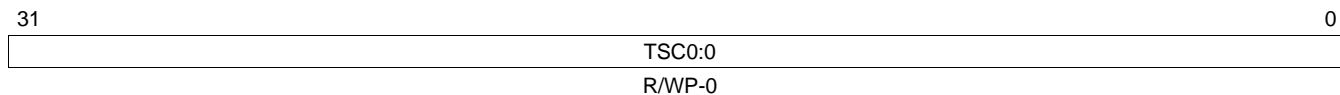
通过将 1 写入到 TCC (CANMC.14) 中，使 TSC 寄存器最高有效位被清零。邮箱 16 成功的发送或收到（取决于 CANMD.16 位的设置）消息时，TSC 寄存器也可以被清零。这通过设置 MBCC 位 (CANMC.15) 来启用。因此，它有可能使用邮箱 16 进行全局网络时间同步。CPU 可以读取和写入计数器。

计数器的溢出由 TSC-计数器-溢出-中断 标志进行检测 (TCOF n -CANGIF n .16)。当 TSC 计数器的最高位变为 1 时一个溢出发生。因此，CPU 有足够的时间来处理这种情况。

18.1.1 时间戳计数器寄存器 (CANTSC)

该寄存器保存在任何时刻的时间戳计数器的值。这是一个自由运行的 32 位定时器，它由 CAN 总线的位时钟计时。例如，在 1Mbps 的比特率时，CANTSC 会每 1 μ s 累加一次。

图 28. 时间戳计数器寄存器 (CANTSC)



图例：R = 读取；WP = 只在 EALLOW 启用模式下写入；-=复位后的值
 请注意：只适用于 eCAN 模式，被保留在 SCC 中

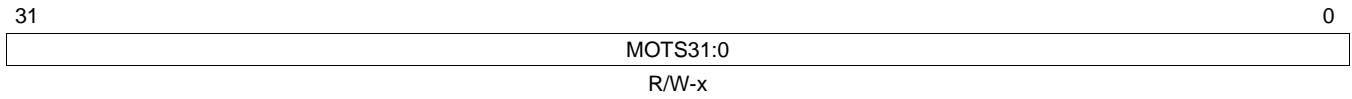
表 25. 时间戳计数器寄存器 (CANTSC) 字段说明

位	字段	值	说明
31:0	TSC0:0		时间戳计数器寄存器。将本地网络时间计数器的值用于时间戳和超时功能。

18.1.2 消息对象时间戳寄存器 (MOTS)

在相应的邮箱数据被成功发送或接收时，该寄存器保存 TSC 的值。每个邮箱都有其自己的 MOTS 寄存器。

图 29. 消息对象时间戳寄存器 (MOTS)



图例：R/W = 读取/写入；-n = 复位后的值

表 26. 消息对象时间戳寄存器 (MOTS) 字段说明

位	字段	值	说明
31:0	MOTS31:0		消息对象时间戳寄存器。当消息已被实际接收到或发出时的时间戳计数器 (TSC) 的值。

18.2 超时功能

为确保在预定义时间内所有消息都能发出或收到，所以每个邮箱都设置了自己的超时寄存器。如果消息在超时寄存器表明的时间内，没有发出或收到一个消息并且 TOC 寄存器中相应的 TOC 位[n] 被设定的话，那么在超时状态寄存器（TOS）中将设定一个标志。

对于发送邮箱，不论是成功发送或者发送请求中止的原因，当 TOC[n]位或相应的 TRS [n]位被清除时，TOS 标志被清除，。对于接收邮箱，当相应的 TOC[n]位被清除时，TOS[n]标志被清除。

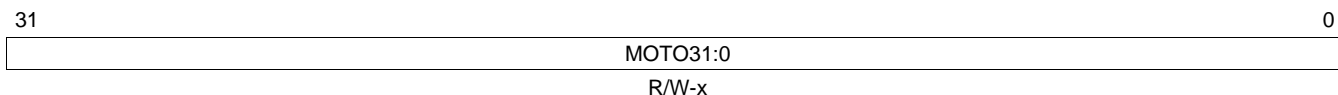
CPU 也可以通过往超时状态寄存器中写入一个 1 来清除超时状态寄存器的标志。

消息对象的超时寄存器（MOTO）被执行为一个 RAM。状态机扫描所有的 MOTO 寄存器并将它们与 TSC 计数器的值进行比较。如果 TSC 寄存器的值等于或大于超时寄存器中的值，以及相应的 TRS 位（仅适用于发送邮箱）被设置，并且 TOC[n]位被设置，则相应的 TOS 位[n]被设置。由于所有的超时寄存器都要按顺序进行扫描，所以在 TOS[n]位设置之前会出现延迟。

18.2.1 消息对象超时寄存器（MOTO）

在相应的邮箱数据被成功发送或接收时，该寄存器保存 TSC 的超时值。每个邮箱都有其自己的 MOTO 寄存器。

图 30. 消息对象超时寄存器（MOTO）



图例：R/W = 读取/写入；-n = 复位后的值；x = 不确定

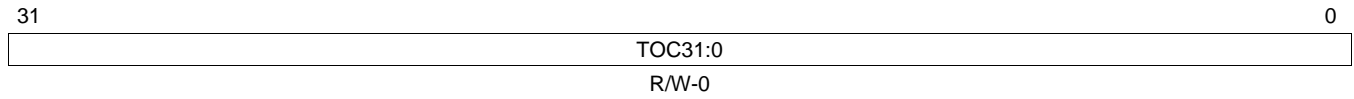
表 27. 消息对象超时寄存器（MOTO）字段说明

位	字段	值	说明
31:0	MOTO31:0		消息对象超时寄存器。实际发送或接收消息的时间戳计数器（TSC）限值。

18.2.2 超时控制寄存器 (CANTOC)

该寄存器控制指定的邮箱是否启用制超时功能。

图 31. 超时控制寄存器 (CANTOC)



图例: R/W = 读取/写入; -n = 复位后的值

表 28. 超时控制寄存器 (CANTOC) 字段说明

位	字段	值	说明
31:0	TOC31:0	1	超时控制寄存器
		0	TOCn位必须由 CPU 置位来启动邮箱的超时功能n。在设定 TOCn位之前, 相应的 MOTO 寄存器应当载入与 TSC 相关的超时值。
		0	超时功能被禁用。从不设定 TOS[n]标志。

18.2.3 超时状态寄存器 (CANTOS)

该寄存器保存已超时邮箱的状态信息。

图 32. 超时状态寄存器 (CANTOS)

31	TOS31:0	0
R/C-0		

图例: R/C = 读取/清除; -n= 复位后的值

表 29. 超时状态寄存器 (CANTOS) 字段说明

位	字段	值	说明
31:0	TOS 31:0	1	超时状态寄存器 邮箱 n 已经超时。TSC 寄存器中的值大于或等于与邮箱 n 相对应的超时寄存器中的值并且 TOC[n] 位被设定。
		0	没有发生超时或该邮箱的此功能被禁用。

当全部满足以下三个条件时, TOS n 位被设定:

1. TSC 值大于或等于超时寄存器 (MOTOn) 中的值。
2. TOC n 位被置位。
3. TRS n 位被置位。

超时寄存器被作为一个 RAM 执行。状态机扫描所有超时寄存器并把它们与时间戳计数器的值进行比较。由于所有超时寄存器依次被扫描, 即使发送邮箱已超时, TOS n 位也有可能未被置位。在状态机扫描超时该邮箱的超时寄存器前, 当邮箱已成功发送并清除了 TRS n 位时这个情况会发生。对接收邮箱也是如此。在这种情况下, 在状态机扫描该邮箱的超时寄存器时, RMP n 位可被设置为 1。然而, 接收邮箱可能在超时寄存器指定的时间前没有收到的消息。

18.3 用户应用程序中 MTOF 0/1 位的运行状态/用法。

在邮箱接收/发送时, 首先将这个标志置为有效, MTOF 0/1 位被 CPK (和 TOS n 位一起) 自动清除。它也可以由用户清除 (通过 CPU)。在超时的状态下, MTOF0/1 位 (以及 TOS n 位) 被置位。在一次 (最终) 成功通信后, 这些位由 CPK 自动清除。以下是 MTOF0/1 位可能的运行状态/用法:

1. 发生超时状况。MTOF0/1 位和 TOS n 位都被置位。通信从未成功过; 即, 帧从未被传送 (或被接收) 一个中断被置为有效。应用程序来处理该问题, 并最终把 MTOF0/1 位和 TOS n 位都清除
2. 出现超时状况。MTOF0/1 位和 TOS n 位都被置位。然而, 通信最终成功; 即, 帧被传送 (或被接收)。MTOF0/1 位和 TOS n 位都被 CPK 自动清除。因为发生中断被记录在 PIE 模块中, 所以中断仍然被置为有效。当 ISR 扫描 GIF 寄存器时, 它不会查看 MTOF0/1 位的设置情况。这就是虚中断情况。应用程序仅仅返回到主代码。
3. 出现超时状况。MTOF 0/1 位和 TOS n 位都被置位。在执行与超时有关的 ISR 时, 通讯是成功的。这种状况必须谨慎处理。在中断被置为有效的时间和 ISR 正尝试采取纠错操作时间之间应用不应该重传一个邮箱。这样做的的方式是轮询 GSR 寄存器中的 TM/RM 位。这些位表示了 CPK 是否正在发送/接收。如果在这样的情况下, 应用程序应该一直等到通信结束再检查 TOS. n 位开始。如果通讯仍然没有成功, 那么该应用程序应采取纠错操作。

19 邮箱布局

各个邮箱都由以下四个 32 位寄存器组成：

- MSGID - 存储消息 ID
- MSGCTRL - 定义字节，传输优先级和远程帧的数目
- CANMDL - 4 字节数据
- CANMDH - 4 字节数据

19.1 消息标识符寄存器 (MSGID)

对于一个给定的邮箱，该寄存器包含信息 ID 和其他控制位。

图 33. 消息标识符寄存器 (MSGID) 寄存器

31	30	29	28	0
IDE	AME	AAM	ID[28:0]	
R/W-x	R/W-x	R/W-x	R/W-x	

图例：R/W=读取/写入；R=只读；-n=复位后的值；x=不确定

请注意：当邮箱禁用时 (CANME[n] (CANME.31-0)=0)，该寄存器只可写入。

表 30. 消息标识符寄存器 (MSGID) 字段说明

位	字段	值	说明
31	IDE		标识符扩展位。IDE 位的特征根据 AMI 的位值变化。 当 AMI=1 时： 1. 接收邮箱的 IDE 位是“无关”。接收邮箱的 IDE 位被已传输消息的 IDE 位写覆盖。 2. 必须满足过滤标准以接收一条消息。 3. 要进行比较的位数是被发送消息的 IDE 位值的函数。 当 AMI=0 时： 1. 接收邮箱的 IDE 位决定了要进行比较的位数。 2. 过滤不适用的。为了接收消息，MSGID 必须位与位匹配。 当 AMI=1 时： IDE = 1: 收到的消息有扩展标识符。 IDE = 0: 收到的消息有一个标准标识符 当 AMI=0 时： IDE=1: 要被接收的消息必须有一个扩展标识符 IDE=0: 要被接收的消息必须有一个标准标识符。
30	AME	1 0	接受屏蔽使能位。AME 只用于接收邮箱。它不能被设置为自动回复 (AAM[n]=1, CANMD[n]=0) 邮箱，否则邮箱的运行状态未定义。该位不能通过消息接收被修改。 1 使用相应的接受屏蔽。 0 没有使用接受屏蔽，所有标识符位必须与接收消息相匹配。
29	AAM	1 0	自动应答模式位。该位唯有对配置为传输的消息邮箱有效。对于接收邮箱，该位无效：邮箱总是按正常接收操作配置。 该位不能由一个消息接收修改。 1 自动应答模式。如果接收到匹配的远程请求，CAN 模块通过发送邮箱内容来应答远程请求。 0 正常传输模式。邮箱不会回复远程请求。远程请求帧的接收对消息邮箱没有影响。
28:0	ID[28:0]	1 0	消息标识符 1 在标准标识符模式中，如果 IDE 位 (MSGID.31) = 0，消息标识符存储在 ID.28: 18 位中。在这种情况下，ID.17: 0 位没有任何意义。 0 在扩展标识符模式中，如果 IDE 位 (MSGID.31) = 1，消息标识符存储在 ID.28: 0 位中。

19.2 CPU 邮箱访问

只有邮箱被禁用时（CANME[n]（CANME.31-0）= 0）才能完成到标识符的写入访问。访问数据字段期间的关键是在 CAN 模块读取数据时，数据不会改变。因此，对于一个接收邮箱，到数据字段的写入访问被禁用。

对于发送邮箱，如果 TRS (TRS.31-0) 或 TRR(TRR.31-0) 标志被设置，访问通常被拒绝。在这些情况下，可将一个中断置为有效。有一种访问这些邮箱方法就是，在访问邮箱数据之前设定 CDR (MC.8)。

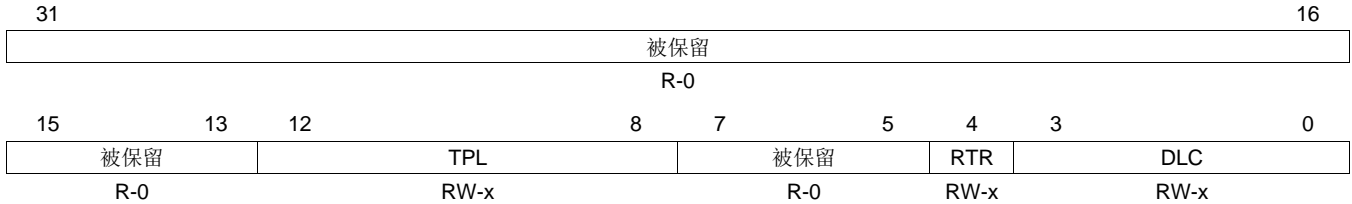
CPU 访问结束后，CPU 必须通过将 0 写入到它来清除 CDR 标志。CAN 模块在读取邮箱前后均会检查此标志。如果在这些检查中 CDR 标志被设置，则 CAN 模块不发送消息，但继续寻找其他的发送请求。CDR 标志的设置也将阻止拒绝写入中断（WDI）被置为有效。

19.3 消息控制寄存器 (MSGCTRL)

对于一个发送邮箱，该寄存器指定要发送的字节数和传输优先级。它还指定的远程帧的操作。

注：作为 CAN 模块的初始化过程的一部分，MSGCTRL n 寄存器的所有位都必须先初始化为零，然后再将不同的位字段初始化为所需的值。

图 34. 消息控制寄存器 (MSGCTRL)



图例：RW = 随时读取，邮箱禁用或被配置为传输时写入；- n =复位后的值； x =不确定的
 请注意：如果邮箱 n 被配置为传输 (CANMD[n] (CANMD.31-0)=0) 或邮箱被禁用 (CANME[n] (CANME.31-0)=0)，MSGCTRL 寄存器 (n) 只能被写入。

表 31. 消息控制寄存器 (MSGCTRL) 字段说明

位	字段	值	说明
31:13	被保留		被保留
12:8	TPL 0:0		传输优先级。这 5 位字段定义了这个邮箱相对于其它 31 个邮箱的优先级。数字越高拥有越高的优先级。当两个邮箱的优先级相同时，邮箱编号大的那个先进行传输。TPL 只用于传输邮箱。TPL 不用于 SCC 模式。
7:5	被保留		被保留
4	RTR	1 0	远程传输请求位 对于接收邮箱：如果 TRS 标志被设置，那么远程帧被传送，并在同一邮箱中收到相应的数据帧。一旦一个远程帧被发送，则邮箱的 TRS 位被 CAN 模块清除。 对于传输邮箱：如果 TRS 标志被设置，那么远程帧被传送，但在另一邮箱中收到相应的数据帧。 无远程帧请求
3:0	DLC 3:0		数据长度代码。这些位中的数量决定发送或接收了多少字节的数据。有效值范围从 0 到 8。从 9 到 15 的值是不允许的。

19.4 消息数据寄存器 (CANMDL, CANMDH)

邮箱的 8 个字节用来存储 CAN 消息的数据字段。DBO (MC.10) 的设置决定了存储数据的顺序。从 CAN 总线的数据传输或接收，从 0 字节开始。

- DBO (MC.10)=1 时，数据存储或读取开始于 CANMDL 寄存器最低有效字节并结束于 CANMDH 寄存器最高有效字节。
- DBO (MC.10)= 0 时，数据存储或读取开始于 CANMDL 寄存器最高有效字节并结束于 CANMDH 寄存器最低有效字节。

如果邮箱 n 被配置为传输 (CANMD[n] (CANMD.31-0)=0) 或者邮箱被禁用 (CANME[n] (CANME.31-0)=0)，CANMDL 寄存器 (n) 和 CANMDH(n) 只能能被写入。如果 TRS[n] (TRS.31-0)=1，寄存器 CANMDL (n) 和 CANMDH (n) 不能写入，除非 CDR(MC.8)=1，且 MBNR(MC.4-0) 被设置为 n 。这些设置也适用于一个配置为答复模式 (AAM(MSGID.29)= 1) 的消息对象。

图 35. DBO=0(CANMDL) 时的消息数据低寄存器

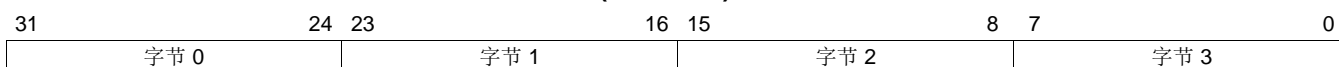


图 36. DBO=0 (CANMDH) 时的消息数据高寄存器

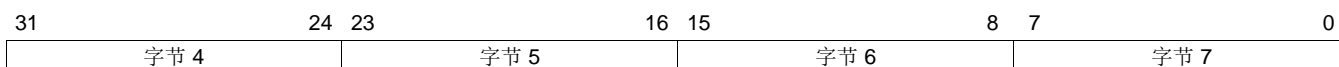


图 37. DBO=1 (CANMDL) 时的消息数据低寄存器

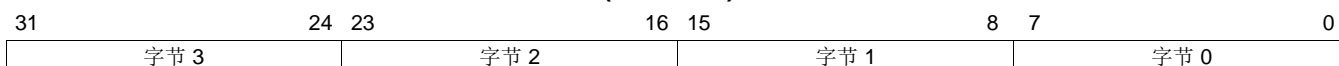
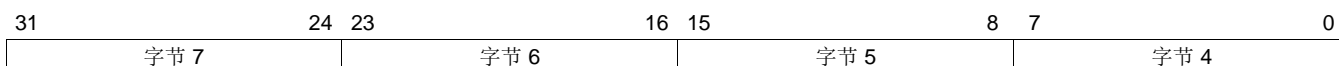


图 38. DBO=1 (CANMDH) 时的消息数据高寄存器



注：超出有效接收数据的数据字段被任何消息接收修改，且是不确定的。

20 接受滤波器

首先将传入消息的标识符与该邮箱的消息标识符（被存储在该邮箱的消息）相比较。之后，适当的接受屏蔽被用于屏蔽掉不应进行比较的标识符的位。

在 SCC 的兼容模式中，全局接受屏蔽（GAM）用于 6 至 15 邮箱。进入的消息被存储在有匹配标识符的编号最高的邮箱。如果在 15 至 6 邮箱中没有匹配的标识符，则把进入的消息与存储在邮箱 5 到 3 的标识符相比较，之后是邮箱 2 至 0 的标识符。

邮箱 5 到 3 使用 SCC 寄存器的局部接受屏蔽 LAM(3)。邮箱 2 到 0 使用 SCC 寄存器的局部接受屏蔽 LAM(0)。用于特定用途，请参阅图 39。

要修改全局接受屏蔽寄存器（CANGAM）和 SCC 的两个本地接受屏蔽寄存器，CAN 模块必须设置为初始化模式（请见 3.1 节）。

eCAN 的 32 个邮箱的任一个都有其各自的局部接受屏蔽 LAM(0) 至 LAM(31)。在 eCAN 中没有全局接受屏蔽。

用于比较的屏蔽选择取决于所用的模式（SCC 或 eCAN）。

20.1 本地接受屏蔽 (CANLAM)

本地接受过滤允许用户在本地屏蔽（无关）进入消息的任何标识符位。

在 SCC 中，本地接受屏蔽寄存器 LAM(0) 用于 邮箱 2 至 0。本地接受屏蔽寄存器 LAM(3) 用于 邮箱 5 至 3。对于邮箱 6 至 15，使用全局接受屏蔽 (CANGAM) 寄存器。

在 SCC 模块的硬件或软件复位后，CANGAM 被复位至零。在 eCAN 复位后，LAM 寄存器不会被修改。

在 eCAN 中，每个信箱（0 到 31）都有其各自的屏蔽寄存器，LAM(0) 至 LAM(31)。进入的消息被存储在具有匹配标识符的编号最高的邮箱中。

图 39. 本地接受屏蔽寄存器 (LAM n)

31	30	29	28	16
LAMI	被保留	LAM n [28:16]		
R/W-0	R/W-0	R/W-0		
15	LAM n [15:0]			0
				R/W-0

图例: R/W = 读取/写入; - n = 复位后的值

表 32. 本地接受屏蔽寄存器 (LAM n) 字段说明

位	字段	值	说明
31	LAMI	1	本地接受屏蔽标识符扩展位 可以接收标准和扩展帧。在扩展帧的情况下，标识符的所有 29 位被存储在邮箱中，本地接受屏蔽寄存器的所有 29 位被过滤器使用。在一个标准帧的情况下，只有标识符的头 11 个位 (28 至 18 位) 和本地接受屏蔽被使用。
		0	存储在邮箱中的标识符扩展位决定了哪些消息应该被接收到。
30:29	被保留		读取未定义，写入无效。
28:0	LAM[28:0]	1	这些位启用一个进入消息的任意标识符位的屏蔽。 针对接受到的标识符的相应位，接受一个 0 或 1 (无关)。
		0	接收到的标识符位值必须与 MSGID 寄存器的相应标识符位相匹配。

你可以本地屏蔽传入消息的任何标识符位。一个 1 值代表“无关”或者针对那个位位置接收一个 0 或 1。一个 0 值意味着进入的位值必须与消息标识符的相应位完全相匹配。

如果本地接受屏蔽标识符扩展位被置位 (LAMI= 1 => 无关)，则可以接收标准和扩展帧。一个扩展帧使用存储在邮箱中标识符的所有 29 位，并且本地接受屏蔽寄存器的所有 29 位被用于过滤器。对于一个标准帧，只有标识符的头 11 个位 (28 至 18 位) 和局部接收屏蔽被使用。

如果本地接受屏蔽标识符扩展位被复位 (LAM=0)，存储在邮箱中的标识符扩展位确定收到的消息。

本节解释了初始化过程，并描述了配置 eCAN 模块的过程。

Topic	Page

1 CAN 模块初始化

在使用 CAN 模块前必须将其初始化。如果模块在初始化模式，初始化是唯一可行的。图 40 是一个显示进程的流程图。

编辑 CCR (CANMC.12)=1 设定初始化模式。只有在 CCE(CANES.4)=1 时，才可以进行初始化。此后，可以写入配置寄存器。

SCC 模式只：

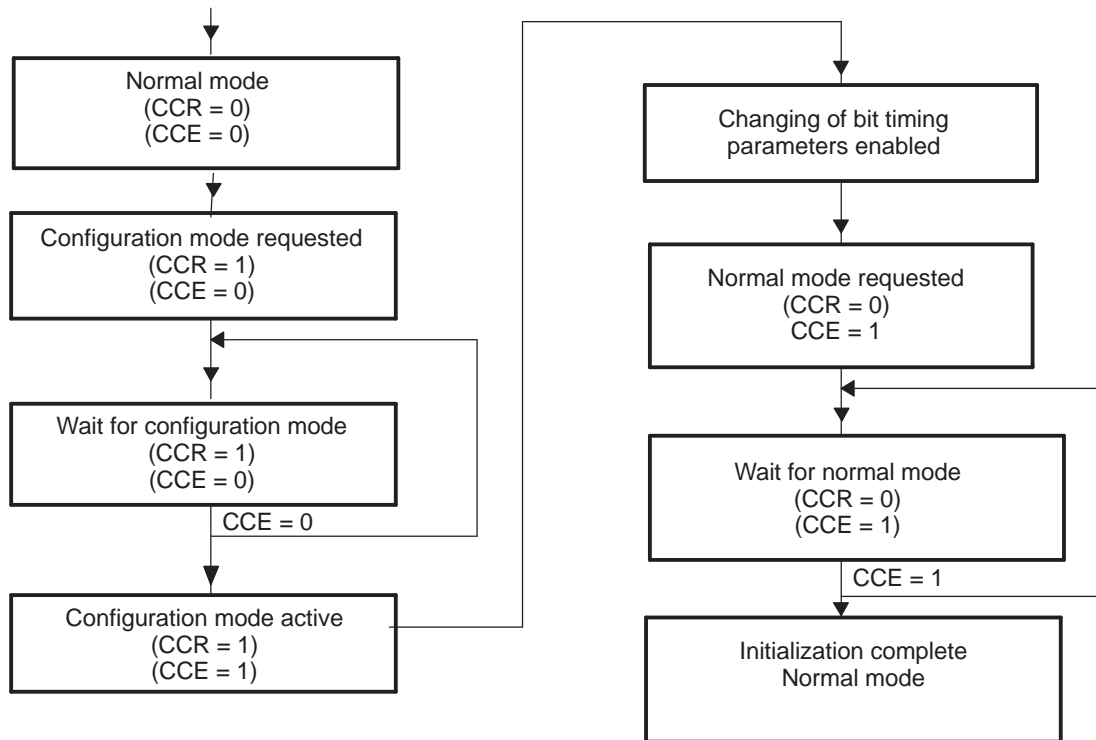
为了更改全局接受屏蔽寄存器 (CANGAM) 和两个本地接受屏蔽寄存器[LAM(0) 和 LAM(3)]，CAN 模块也必须在初始化模式中设定。

通过设定 (CCR CANMC.12)=0，模块再次被激活。

硬件复位后，初始化模式被激活。

注： 如果 CANBTC 寄存器被编程为 0 值，或保留为初始值，CAN 模块从未离开初始化模式，当清除 CCR 位时，CCE (CANES.4) 位仍为 1。

图 40. 初始化序列



注：初始化模式和正常模式（反之亦然）之间的转换与 CAN 网络进行同步。也就是说，在 CAN 控制器更改模式前，它一直等到检测到总线空闲序列（=11 个隐性位）。在一个停留在显性总线错误事件中，CAN 控制器无法检测一个总线空闲状态，因此无法执行的模式转换。

1.1 CAN 位时序配置

CAN 协议技术规范把标称位时间分为四个不同的时间段：

SYNC_SEG：这部分位时间用于同步总线上的不同节点。预计这一段内有一个边沿。这个部分始终是 1 个时间定额 (TQ)。

PROP_SEG：这部分位时间被用来补偿网络内的物理延时时间。它是总线线路上信号传播时间、输入比较器延迟、和输出驱动器延迟总和的二倍。这部分被编程为 1 到 8 个时间定额 (TQ)。

PHASE_SEG1：这个相位被用于补偿正边沿相位误差。这部分可在 1 到 8 个时间定额 (TQ) 之间编辑，并可通过重新同步加长。

PHASE_SEG2：这个相位被用于补偿负边沿相位误差。这部分可在 2 到 8 个时间定额 (TQ) 之间编辑，并可通过重新同步来缩短。

在 eCAN 模块中，CAN 总线上一个位的长度由参数 TSEG1 (BTC.6-3)，TSEG2 (BTC.20) 和 BRP (BTC.23.16) 确定。

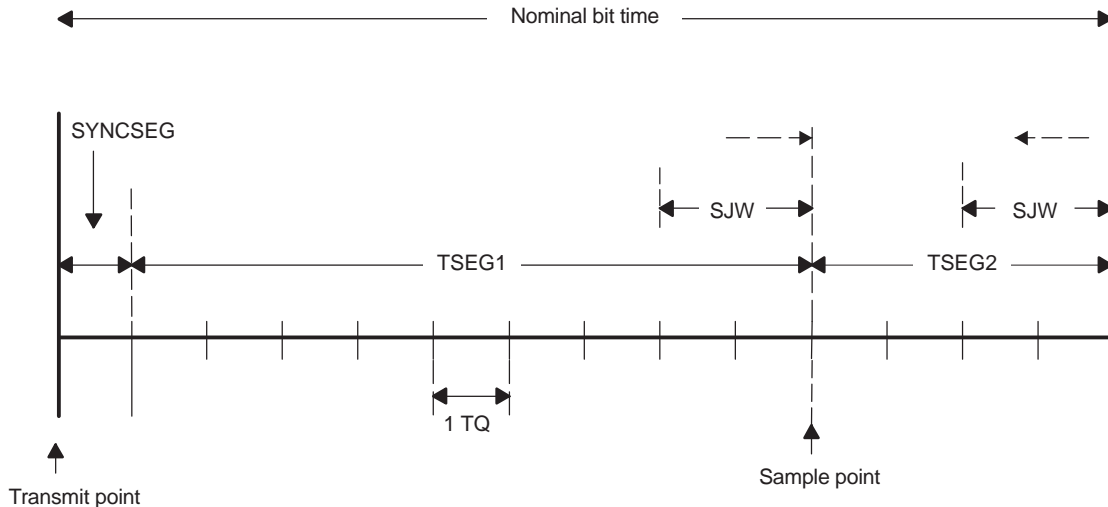
就如由 CAN 协议定义的一样，TSEG1 结合了 PROP_SEG 和 PHASE_SEG1 两个时间段。TSEG2 定义了 PHASE_SEG2 时间段的长度。

IPT（信息处理时间）对应位读取处理所需的时间。IPT 对应 TQ 的两个单位。

确定位段值时必须满足以下的位定时规则：

- TSEG1 (最小) ≥ TSEG2
- IPT ≤ TSEG1 ≤ 16TQ
- IPT ≤ TSEG2 ≤ 8TQ
- IPT = 3/BRP (由此产生 IPT 必须四舍五入到下一个整数值)
- 1TQ ≤ SJW 的最小值 [4 TQ, TSEG2] (SJW = 同步跳转宽度)
- 为了使用三次采样模式, 必须选定 BRP ≥ 5

图 41. CAN 位时序



A 可以由 SJW 对 TSEG1 进行延长或对 TSEG2 进行缩短

1.2 CAN 比特率计算

比特率乃按每秒比特计算, 如下:

$$\text{Bit rate} = \frac{\text{SYSCLKOUT} / 2}{\text{BRP} \times \text{Bit Time}}$$

在这里, 位时间是每比特时间定额 (TQ) 数量。SYSCLKOUT 是 CAN 模块系统的时钟频率, 与 CPU 的时钟频率一样。BRP 是 BRP_{reg}+1 (CANBTC.23-16) 的值。

位时间定义如下:

$$\text{Bit-time} = (\text{TSEG1}_{\text{reg}} + 1) + (\text{TSEG2}_{\text{reg}} + 1) + 1$$

在上述方程式中, TSEG1_{reg}和 TSEG2_{reg}表示实际写入到 CANBTC寄存器中相应字段的值。当 CAN 模块访问 TSEG1_{reg}, TSEG2_{reg}, SJW_{reg}以及 BRP_{reg}的这些参数时, 它们会自动增加 1。按照图 41, TSEG1, TSEG2 和 SJW 表示适用的值。

$$\text{Bit-time} = \text{TSEG1} + \text{TSEG2} + 1$$

1.3 针对30MHz 系统时钟的的位配置参数

针对不同 CAN 模块时钟、比特率和采样点的 CANBTC 位字段, 这部分 提供了示例值。请注意, 这些值仅用于说明。在实际应用中, 在选择时序参数前, 必须把各种实体的传播延迟, 如网络电缆和收发器/隔离器考虑在内。

表 33显示了如何使用一个针对 80% SP 的 15BT 来改变 BRP_{reg}字段来实现不同的比特率。

表 33. 比特率 (BT = 15, TSEG1_{reg}=10, TSEG2_{reg}=2, 采样点 = 80%) 的 BRP 字段

CAN 总线速度	BRP	CAN 模块时钟
1Mbps	BRP _{reg} +1=2	15MHz
500kbps	BRP _{reg} +1=4	7.5MHz
250kbps	BRP _{reg} +1=8	3.75MHz
125kbps	BRP _{reg} +1=16	1.875MHz
100kbps	BRP _{reg} +1=20	1.5MHz
50kbps	BRP _{reg} +1=40	0.75MHz

表 34 显示了如何实现与一个 15 BT 的不同采样点。

表 34. 实现与一个 15 BT 的不同采样点

TSEG1 _{reg}	TSEG2 _{reg}	SP
10	2	80%
9	3	73%
8	4	66%
7	5	60%

表 35 显示了如何使用一个针对 80% 采样点的 10BT 来改变 BRP_{reg} 字段来实现不同的比特率。

表 35. 比特率 (BT=10, TSEG1_{reg}=6, TSEG2_{reg}=1, 采样点 = 80%) 的 BRP 字段

CAN 总线速度	BRP	CAN 模块时钟
1Mbps	BRP _{reg} +1=3	10MHz
500kbps	BRP _{reg} +1=6	5MHz
250kbps	BRP _{reg} +1=12	2.5MHz
125kbps	BRP _{reg} +1=24	1.25MHz
100kbps	BRP _{reg} +1=30	1MHz
50kbps	BRP _{reg} +1=60	0.5MHz

注: 对于一个 60MHz 的 SYSCLKOUT, 最低的比特率能达到 4.687kbps。

1.4 EALLOW 保护

为了防止意外修改，一些关键的寄存器 / eCAN 模块的位都受 EALLOW 保护。只有受 EALLOW 保护已被禁用才可以改变这些寄存器/位。以下是在 eCAN 模块中受 EALLOW 保护的寄存器/位：

- CANMC[15..9] & MCR[7..6]
- CANBTC
- CANGIM
- MIM[31..0]
- TSC[31..0]
- IOCONT1[3]
- IOCONT2[3]

2 配置 eCAN 的步骤

注：这个序列必须在 EALLOW 被启用时完成。

配置 eCAN 的操作，必须执行以下步骤：

- Step 1. CAN 模块的使能时钟。
- Step 2. 将 CANTX 和 CANRX 引脚设定为 CAN 功能：
 - (a) 写入 CANTIOC.3:0=0x08
 - (b) 写入 CANTIOC.3:0=0x08
- Step 3. 复位后，CCR 位 (CANMC.12) 和 CCE 位(CANES.4) 设置为 1。这使用户能够配置位时序配置寄存器 (CANBTC)。

如果 CCE 位被设置 (CANES.4=1)，进行下一步；否则，设置 CCR 位(CANMC.12=1)，并等待，直到 CCE 位设置(CANES.4=1)。
- Step 4. 使用适当的时序值对 CANBTC 寄存器进行配置。确保 TSEG1 和 TSEG2 不等于 0。如果两个值等于 0，则模块不能退出初始化模式。
- Step 5. 对于 SCC 模式，现在对接受屏蔽寄存器编程。例如：

写入 LAM(3)=0x3C0000
- Step 6. 对主控制寄存器 (CANMC) 进行如下编程：
 - (a) 清除 CCR (CANMC.12) = 0
 - (b) 清除 PDR (CANMC.11) = 0
 - (c) 清除 DBO (CANMC.10) = 0
 - (d) 清除 WUBA (CANMC.9)= 0
 - (e) 清除 CDR (CANMC.8) = 0
 - (f) 清除 ABO (CANMC.7) = 0
 - (g) 清除 STM (CANMC.6) = 0
 - (h) 清除 SRES (CANMC.5) = 0
 - (i) 清除 MBNR (CANMC.4-0) = 0
- Step 7. 将 MSGCTRLn 寄存器的所有位初始化为零。
- Step 8. 验证 CCE 位是否被清零 (CANES.4=0)，如果被清零则表明 CAN 模块已被配置。

这就完成了基本功能的设置。

2.1 配置邮箱用于传输

为了传输一个消息，必须执行以下步骤（本例中以邮箱 1 为例）：

1. 将 CANTRS 寄存器中适当位清零：

清除 CANTRS.1=0（向 TRS 写入 0 无效；相反设置 TRR.1 并等待直到 TRS.1 清除。）如果 RTR 位被置位，TRS 可以发送一个远程帧。一旦一个远程帧被发送，则邮箱的 TRS 位被 CAN 模块清除。因此，同一个邮箱可被用来从其它节点中请求一个数据帧。

2. 通过清除邮箱使能（CANME）寄存器的相应位来禁用邮箱。

清除 CANME.filter1=0

3. 加载邮箱的消息标识符（MSGID）寄存器。对于正常的发送邮箱（MSGID.30=0 和 MSGID.29=0），清除 AME（MSGID.30）和 AAM（MSGID.29）位。在操作过程中，该寄存器通常不会被修改。只有在邮箱被禁用时，才能把它修改。例如：

(a) 写入 MSGID(1)=0x15AC0000

(b) 消息控制字段寄存器（MSGCTRL.3:0）DLC 字段写入数据的长度。RTR 标志通常都会被清除（MSGCTRL.4=0）。

(c) 通过清除 CANMD 寄存器的相应位来设置邮箱方向。

(d) 清除 CANMD.1=0

4. 通过设置 CANME 寄存器的相应位设置邮箱启用

设置 CANME.1=1

这将邮箱 1 配置为传输模式。

2.2 传输一个消息

为了开始一个传输（在这个例子中，对于邮箱：

1. 将消息数据写入邮箱数据字段。

(a) 由于在配置时，DBO（MC.10）设置为零，MSGCTRL(1) 设置为 2，数据存储在 CANMDL(1) 2 个最有效字节中。

(b) 写入 CANMDL(1) = xxxx0000h

2. 在发送请求寄存器中设定相应的标志（CANTRS.1=1）来启动的消息传输。现在 CAN 模块处理 CAN 消息的完整传输。

3. 等待，直到相应邮箱的发送确认标志被设置（TA.1=1）。成功传输后，CAN 模块设置此标志。

4. 传输成功或中止传输后，模块将 TRS 标志复位为 0（TRS.1=0）。

5. 为了下一个传输（从同一邮箱）发送确认必须清除。

(a) 设置 TA.1=1

(b) 等到 TA 读取为 0

6. 要在同一邮箱发送另一条消息，邮箱 RAM 中的数据必须更新。设置 TRS.1 标志来启动下一个传输。写入邮箱 RAM 中的可以是半字（16 位）或整字（32 位），但模块总是从偶数边界返回 32 位。CPU 必须接受所有 32 位或其中的一部分。

2.3 配置邮箱用于接收

为了配置接收邮箱，必须执行以下步骤（本例中以邮箱 3 为例）：

1. 清除邮箱使能（CANME）寄存器的相应位来禁用邮箱。

清除 CANME.3=0

2. 将所选的标识符写入相应的 MSGID 寄存器。标识符扩展位必须被配置为适合预计的标识符。如果使用接受屏蔽，接受屏蔽使能（AME）位必须被设定（MSGID.30=1）。例如：

写入 MSGID(3) = 0x4f780000

3. AME 位如果设置为 1，必须设定相应的接受屏蔽。

写入 LAM(3) = 0x03c0000。

4. 通过设置邮箱方向寄存器 (CANMD.3=1) 的相应标志, 将邮箱配置为一个接收邮箱。确保此操作不会影响该寄存器中的其它位。
5. 如果邮箱中的数据受到保护, 现在需要对写覆盖保护控制寄存器 (CANOPC) 进行编程。如果没有消息必须被丢弃, 这种保护是非常有用的。如果设置了 OPC, 该软件必须确保一个额外的邮箱 (缓冲邮箱) 被配置用来存储“溢出”的消息。否则, 消息可能会在没有任何通知的情况下丢失。
写入 OPC.3=1
6. 通过设置邮箱中的使能寄存器 (CANME) 相应的标志来启用邮箱。这必须通过读取 CANME, 并写回 (CANME|=0x0008) 的方法来完成以确保没有其他标志被意外更改。

对象现在被配置为接收模式。该对象的任何传入消息都将自动处理。

2.4 接收一个消息

以邮箱 3 为例。当接收到一个消息时, 接收消息等待寄存器 (CANRMP) 的相应标志被设为 1, 并产生一个中断。然后, CPU 可以从邮箱 RAM 中读取消息。在 CPU 从邮箱读取消息之前, 应先清除的 RMP 位 (RMP.3=1)。CPU 还应该检查接收消息丢失标志 RML.3=1。根据不同的应用, CPU 来决定如何处理这种情况。

读取数据后, CPU 需要检查 RMP 位有没有被模块再次设置。如果 RMP 位已被设置为 1, 说明数据有可能已损坏。因为当 CPU 读取旧消息的同时收到一个新消息, 所以 CPU 需要再次读取数据。

2.5 过载情况下的处理

如果 CPU 无法快速的处理重要的信息, 那么为该标识符配置多个邮箱的方法应该是可取的。下面是一个例子, 其中对象 3, 4 和 5 有相同的标识符并且共用相同的屏蔽。在 SCC 模式中, 屏蔽就是 LAM(3)。对于 eCAN, 每个对象都有自己的 LAM: LAM(3), LAM(4), 和 LAM(5), 它们都需要以相同值进行编程。

为了确保无消息丢失, 针对对象 4 和 5 设置了 OPC 标志, 这样就防止了未读消息被写覆盖。如果 CAN 模块一定要存储接收到的消息, 它首先要检查邮箱 5。如果该邮箱是空的, 那么消息就被储存在该邮箱。如果对象 5 的 RMP 标志被设置 (邮箱被占用), 那么 CAN 模块将检查邮箱 4 的状态。如果邮箱 4 也被占用, 那么模块将检查邮箱 3 并将消息存储在该邮箱, 这是由于并未针对邮箱 3 设置 OPC 标志。如果邮箱 3 内容之前尚未读取, 那么它设置对象 3 的 RML 标志, 该标志可以启动一个中断。

对象 4 产生一个中断使 CPU 立刻读取邮箱 4 和邮箱 5 的方法也是可行的。该技术同样适用于请求超过 8 字节数据 (即, 不止一条消息) 的消息。这种情况下, 消息的所有数据都能在邮箱中收集到并立刻读取。

3 远程帧邮箱的处理

远程帧处理有两个功能。一个功能是模块针对来自另一个节点的数据数据, 另一个功能是针对需要应答的模块的数据请求。

3.1 从另一个节点请求数据

为了从另一个节点请求数据, 该对象被配置为接收邮箱。这个例子使用对象 3, CPU 需要做到以下几点:

1. 把消息控制字段寄存器 (CANMSGCTRL) 的 RTR 位设置为 1。
写入 MSGCTRL(3)=0X12
2. 向消息标识符寄存器 (MSGID) 中写入正确的标识符。
写入 MSGID(3)= 0x4F780000
3. 为该邮箱设置 CANTRS 标志。由于邮箱被配置为接收, 因此它只是发送一个远程请求消息到其他节点。
设置 CANTRS.3=1
4. 该模块把应答存储在邮箱, 并在接收 RMP 位时对其进行设置。这个操作可以启动一个中断。此外, 请确保没有其他邮箱具有相同的 ID。
等待 RMP.3=1

5. 阅读收到的消息。

3.2 应答一个远程请求

1. 把该对象配置作为一个发送邮箱。
2. 启用邮箱前，设定 MSGID 寄存器中的自动应答模式 (AA)(MSGID.29) 位。
MSGID(1)=0x35AC0000
3. 更新数据字段。
MDL, MDH(1)=xxxxxxxh
4. 通过把 CANME 标志设置为 1 来启用邮箱。
CANME.1=1
当从另一个节点收到远程请求时，TRS 标志自动置位，数据传送到那个节点。收到的消息和发送的消息的标识符是相同的。
数据传输后，TA 标志被置位。之后 CPU 可以更新数据。
等待 TA.1=1

3.3 更新数据字段

要更新被配置为自动应答模式的对象的数据，需要执行以下步骤。这个顺序也可以用来更新被 TRS 标志设置配置为正常传输的对象的数据。

1. 设置更改数据请求 (CDR)(MC.8) 位和主控制寄存器 (CANMC) 中那个对象的邮箱号码 (MBNR)。这就告诉 CAN 模块，CPU 想要更改的数据字段。例如，对于对象 1：
写入 MC=0x0000101
2. 把信息数据写入邮箱数据寄存器。例如：
写入 CANMDL(1) = xxxx0000h
3. 清除 CDR 位 (MC.8) 来启用对象。
写入 MC=0x00000000

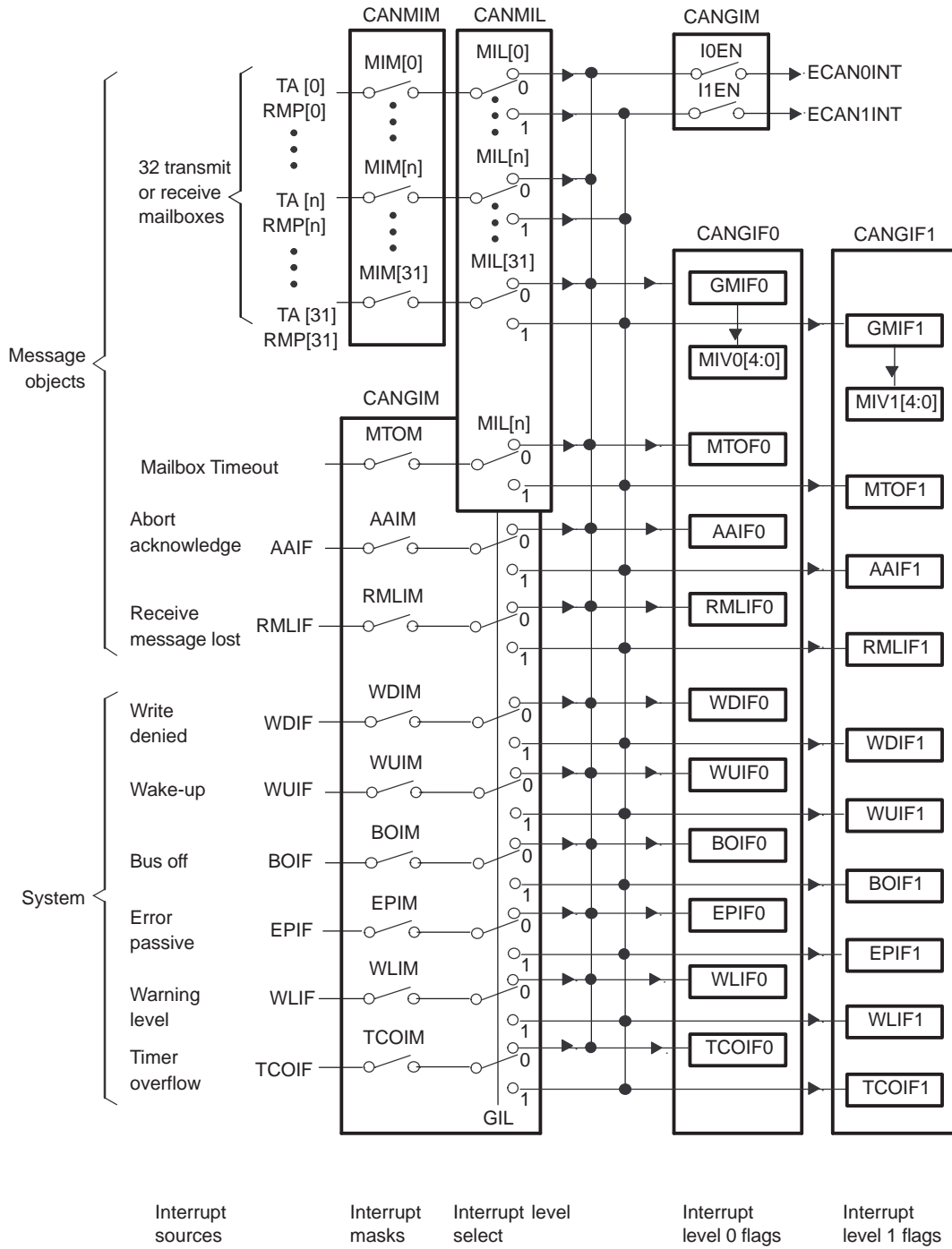
4 中断

有两种不同类型的中断。一种中断类型是邮箱相关的中断，例如，接收消息等待中断或中止确认中断。另一中断类型是处理错误或系统相关中断源的系统中断，例如，被动错误中断或唤醒中断。请参考图 42。

下列事件可以启动两个中断中的一个：

- 邮箱中断
 - 消息接收中断：收到一个消息
 - 信息传输中断：成功一个发送消息
 - 中止确认中断：等待传输被中止
 - 接收消息丢失中断：旧消息被一个新消息（在旧邮件被读取前）写覆盖
 - 邮箱超时中断（只在 eCAN 模式）：在预定的时间帧内其中一条消息没有被发送或未收到
- 系统中断
 - 写入拒绝中断：CPU 试图写入一个邮箱，但不被允许
 - 唤醒中断：在一个唤醒后产生中断
 - 总线关闭中断：CAN 模块进入总线关闭状态
 - 被动错误中断：CAN 模块进入被动错误模式
 - 警告级别中断：一个或两个错误计数器都大于或等于 96
 - 时间戳计数器溢出中断（只适用于 eCAN）：时间戳计数器发生溢出

图 42. 中断方案



4.1 中断方案

如果发生相应的中断状况，则中断标志被置位。系统中中断标志的置位取决于 GIL (CANGIM.2) 的设置。如果被置位，全局中断对在 CANGIF1 寄存器中的位进行置位，否则，在 CANGIF0 寄存器中进行置位。

根据与产生那个中断的邮箱相对应的 MIL[n] 位的设置，对 GMIF0/GMIF1(CANGIF0.15/CANGIF1.15) 位进行置位。如果 MIL[n] 位被置位，相应邮箱中断标志 MIF[N] 设置 CANGIF1 寄存器中的 GMIF1 标志，否则，它设置 GMIF0 标志。

如果所有的中断标志被清零并且一个新的中断标志就被置位，如果相应的中断屏蔽位被置位，则 CAN 模块中断输出线路 (ECAN0INT 或 ECAN1INT) 被激活。中断线路持续保持有效直到由 CPU 在相应位写 1 来清除中断标志。

必须通过在 CANTA 寄存器或 CANRMP 寄存器（取决于邮箱配置）的相应位上写入 1 来清除 GMIF0 (CANGIF0.15) 或者 GMIF1 (CANGIF1.15) 位，且不能在 CANGIF0/CANGIF1 寄存器中清除。

在清除一个或多个中断标志后，一个或多个中断标志仍然等待，就会产生新的中断。通过写 1 到相应位的位置来清除中断标志。如果 GMIF0 或者 GMIF1 位被置位，邮箱中断矢量 MIV0 (CANGIF0.4-0) 或 MIV1 (CANGIF1.4-0) 会表明引起 GMIF 0/1 设置的邮箱号码。它总是显示分配给那个中断线路的最高邮箱中断矢量。

4.2 邮箱中断

eCAN 的 32 个邮箱或 SCC 的 16 个邮箱中的每个邮箱都可以在两个中断输出线路 1 或 0 中的一个上启动一个中断。这些中断可以根据邮箱配置接收或传输中断。

有一个中断屏蔽位 (MIM[n]) 和一个每个邮箱专用的中断级别 (MIL[n])。若要在接收/发送事件上产生邮箱中断，必须设置 MIM 位。如果在接收邮箱收到一个 CAN 消息 (RMP[n]=1) 或由发送邮箱发送一个 CAN 消息 (TA[n]=1)，则将一个中断置为有效。如果一个邮箱被配置为远程请求邮箱 (CANMD[n]=1, MSGCTRL.RTR=1)，在接收到应答帧后，发生中断。应答帧传输成功后 (CANMD[N]=0, MSGID.AAM=1)，一个远程应答邮箱产生中断。

如果相应的中断屏蔽位被置位，RMP[n] 位或 TA[n] 位的设置也将设定 GMIF0/GMIF1 寄存器内的 GMIF0/GMIF1 (GIF0.15/GIF1.15) 标志。然后 GMIF0/GMIF1 标志产生一个中断，相应的邮箱矢量 (=信箱号码)，就可以从 GIF0/GIF1 寄存器中的 MIV0/MIV1 位字段中读取。如果多个邮箱中断等待，MIV0/MIV1 的实际值反映最高优先级中断矢量。中断的产生取决于邮箱中断级别 (MIL) 寄存器的设置。

在一个发送邮件被 TRR[n] 位中止时，GIF0/GIF1 寄存器的中止确认标志 (AA[n]) 和中止确认中断标志 (AAIF) 被置位。如果 GIM 寄存器中的屏蔽位 AAIM 被置位，一个中断在传输中止时被置为有效。清除 AA[n] 标志会清除的 AAIF0/AAIF1 标志。

通过设置 GIF0/GIF1 寄存器中的接收消息丢失标志 RML[n] 和接收消息丢失中断标志 RMLIF0/RMLIF1 来通知一个丢失接收消息。如果一个中断应在丢失接收消息事件时发生，GIM 寄存器中的接收消息丢失中断屏蔽位 (RMLIM) 必须被置位。清除 RML[n] 标志不会复位 RMLIF0/RMLIF1 标志。必须单独清除中断标志。

eCAN 的每个信箱（只在 eCAN 模式）被链接到一个消息对象，超时寄存器 (MOTO)。如果超时事件发生 (TOS[n]=1)，如果 CANGIM 寄存器中的邮箱超时中断屏蔽位 (MTOM) 被置位，则信箱超时中断被置为两个中断线路中的一个。按照有关邮箱的邮箱中断级别 (MIL[n]) 选择邮箱超时中断的中断线路。

4.3 中断处理

通过将两个中断线路中的一个置为有效，CPU 被中断。在中断处理后，一般也应清除中断源，中断标志必须由 CPU 清除。要做到这一点，中断标志必须清除寄存器 CANGIF0 或 CANGIF1 中的中断标志。这通常通过向中断标志中写如一个 1 来完成。在表 36 中也列出了这种情况的一些例外情况。如果没有其他中断等待，这也释放了中断线路。

表 36. eCAN 的中断置位/清除⁽¹⁾

中断 标记	中断条件	GIF0/GIF1	
		确定	清除机制
WLIFn	一个或两个错误计数器都 >= 96	GIL 位	通过写入一个 1 来清除
EPIFn	CAN 模块已进入“错误被动”模式	GIL 位	通过写入一个 1 来清除
BOIFn	CAN 模块已进入“总线关闭”模式	GIL 位	通过写入一个 1 来清除
RMLIFn	一个溢出状况已经发生在 接收邮箱中的一个。	GIL 位	通过清除 RMPn 设置清除 位
WUIFn	CAN 模块已经离开了本地断电模式	GIL 位	通过写入一个 1 来清除
WDIFn	一个到邮箱的写入访问被拒绝	GIL 位	通过写入一个 1 来清除
AAIFn	一个传输请求被中止	GIL 位	通过清除 AAn 位设置来清除。
GMIFn	成功进行数据传输的邮箱中的一个 导致中断的情况的已发送/已接收的一个消息	MILn 位	通过适当的处理 来清除。通过在 CANTA 或 CANRMP 寄存 器中的适当位写入一个 1 来清除
TCOFn	TSC 的最高有效位 (MSB) 从 0 改为到 1	GIL 位	通过写入一个 1 来清除
MTOFn	其中一个邮箱不是如此 发送/接收设定来清除。	MILn 位	通过清除指定时间帧内的 TOSn 位

⁽¹⁾ 解释上述表的关键:

- 1) 中断标志: 这是适用于 CANGIF0/CANGIF1 寄存器的中断标志位的名称。
- 2) 中断条件: 此栏说明了引起中断被置为有效的条件。
- 3) GIF0/GIF1 确定: 可以在 CANGIF0 或 CANGIF1 寄存器中设置中断标志位。根据正在考虑的中断, 这是由 CANGIM 寄存器的 GIL 位或 CANMIL 寄存器的 MILn 位确定的。此栏说明了一个特定的中断是否取决于 GIL 位或 MILn 位。
- 4) 清除机制: 此栏说明了如何可以清除一个标志位。一些位通过写入一个 1 来清除。通过控制 CAN 控制寄存器中的一些其它位来清除其他位。

4.3.1 中断处理配置

为了配置中断处理, 邮箱中断优先级寄存器(CANMIL), 邮箱中断屏蔽寄存器 (CANMIM), 和全局中断屏蔽寄存器 (CANGIM) 都需要进行配置。配置步骤如下所述:

1. 写入 CANMIL 寄存器。这定义了将一个成功传输置为中断线路 0 还是 1。例如, CANMIL = 0xFFFFFFFF 设置所有邮箱中断为 1 级。
2. 配置邮箱中断屏蔽寄存器 (CANMIM) 来屏蔽掉不会引起中断的邮箱。该寄存器可以设置为 0xFFFFFFFF, 这将启动所有邮箱中断。未使用的邮箱无论如何不会引起任何中断。
3. 现在配置 CANGIM 寄存器。标志 AAIM, WDIM, WUIM, BOIM, EPIM, 和 WLIM (GIM.14-9) 应一直被设置 (启用这些中断)。此外, GIL (GIM.2) 位可被设置为具有邮箱中断级别之外的全局中断。I1EN (GIM.1) 和 I0EN (GIM.0) 标志都应被设置以启动两条中断线路。标志 RMLIM (GIM.11) 也可以根据 CPU 的负载设置。

这种配置将所有邮箱中断放置在线路 1 上, 将所有系统中断放置在线路 0。因此, CPU 可以处理所有具有高优先级的系统中断 (一直为重要中断), 和具有较低优先级的邮箱中断 (在其他线路上)。所有具有高优先级的消息也可以定向到中断线路 0。

4.3.2 邮箱中断处理

有三个针对邮箱中断的中断标志。在下面列出了这些标志:

GMIF0/GMIF1: 已经接收或传输一个消息的对象中的一个。邮箱编号在 MIV0/MIV1(GIF0.4-0/GIF1.4-0) 中。正常的处理例程如下:

1. 在引起中断的 GIF 寄存器上进行半字读取。如果该值为负，一个邮箱引起中断。否则，检查 AAIF0/AAIF1 (GIF0.14/GIF1.14)位（中止确认中断标志）或 RMLIF0/RMLIF1 (GIF0.11/GIF1.11) 位（接收消息丢失中断标志）。否则，发生一个系统中断。在这个情况下，每个系统中断标志都必须被检查。
2. 如果 RMLIF (GIF0.11) 标志引起了中断，则其中一个邮箱里的消息已经被新消息所写覆盖。这不应发生在正常操作中。CPU 通过将 1 写入来清除该标志。想要找到发生中断的具体邮箱，CPU 还应检查接收消息丢失寄存器 (RML)。根据应用情况，CPU 决定下一步操作。该中断和一个 GMIF0/GMIF1 中断一起发生。
3. 如果是 AAIF (GIF.14) 标志引起中断，CPU 将中止发送传输操作。CPU 应检查中止确认寄存器 (AA.31-0) 来找到造成中断的邮箱并在请求发生时再次发送那个消息。通过将 1 写入来清除该标志。
4. 如果是 GMIF0/GMIF1 (GIF0.15/GIF1.15) 标志造成中断，那么造成中断的邮箱编号可以从 MIV0/MIV1 (GIF0.4-0/GIF1.4-0) 字段读取。此矢量可被用于跳转到被处理的邮箱的位置。如果这是一个接收邮箱，CPU 应该如上所述读取数据并通过将 1 写入来清除 RMP.31-0 标志。如果这是一个发送邮箱，除非 CPU 需要发送更多数据，不会采取进一步操作。在这种情况下，如上所述的正常发送过程是必须的。CPU 需要通过写入一个 1 来清除发送确认位(TA.31-0)。

4.3.3 中断处理序列

为了 CPU 内核识别和处理 CAN 中断，在任何一个 CAN ISR 中，都必须进行以下操作：

1. 在 CANGIF0/CANGIF1 寄存器中，最初造成中断的标志位必须清除。在这些寄存器中存在两种标志位：
 - (a) 非常相同位需要被清除。以下位这些属于这一类：TCOFn, WDFn, WUIFn, BOIFn, EPIFn, WLIFn
 - (b) 第二组的位需要通过写入相关的寄存器的相应位来清除。以下这些属于这一类：MTOFn, GMIFn, AAIFn, RMLIFn
 - (i) 通过清除 TOS 寄存器的相应位来清除 MTOFn 位。例如，如果由于 MTOFn 位被设置而使邮箱 27 造成了超时状态，为了清除 MTOFn 位，ISR（为超时状态做出合适的操作后）就需要清除 TOS27 位。
 - (ii) 通过清除 TA 或 RMP 寄存器的相应位来清除 GMIFn 位。例如，如果邮箱 19 已经被配置为传输邮箱并完成了传输，TA19 被设定，转而设定 GMIFn 位。为了清除 GMIFn 位，ISR（采取适当操作后）需要清除 TA19 位。如果邮箱 8 已被配置为一个接收邮箱并完成了接收，RMP8 被设定，转而设定 GMIFn 位。为了清除 GMIFn 位，ISR（采取适当操作后）需要清除 RMP8 位。
 - (iii) 通过清除 AA 寄存器的相应位来清除 AAIFn 位。例如，如果由于 AAIFn 位被设置而使邮箱 13 的传输中止，ISR 就需要清除 AA13 位来清除 AAIFn 位。
 - (iv) 通过清除 RMP 寄存器的相应位来清除 RMLIFn 位。例如，如果由于 RMLIFn 位被设置而使邮箱 13 的消息被写覆盖，ISR 就需要清除 RMP13 位来清除 RMLIFn 位。
2. PIEACK 位相应的 CAN 模块必须写入一个 1，可以用下面的 C 语言语句完成：


```
PieCtrlRegs.PIEACK.bit.ACK9 = 1; // Enables PIE to drive a pulse into the CPU
```
3. 与 CAN 模块相对应的 CPU 中断线路必须被启用，它可以用下面的 C 语言的语句完成：


```
IER |= 0x0100; // Enable INT9
```
4. 必须通过清除 INTM 位来全局启用 CPU 中断。

5 CAN 断电模式

已执行本地断电模式，在这里，CAN 模块的内置时钟由 CAN 模块本身取消激活。

5.1 进入和退出本地断电模式

在本地断电模式期间，CAN 模块的时钟被关闭（由 CAN 模块本身关闭），并且只有唤醒逻辑仍然有效。其他外设继续正常运行。

通过写一个 1 到 PDR (CANMC.11) 位来请求本地断电模式，以完成正在进行中的任何数据包的传输。传输完成后，状态位 PDA (CANES.3) 被设置。这证实了 CAN 模块已经进入了断电模式。

CANES 寄存器读取的值是 0x08（PDA 位被设定）。所有其它寄存器的读访问传送值 0x00。

当 PDR 位被清除或如果在 CAN 总线线路上检测到任何总线活动（如果已启用总线活动唤醒），该模块离开本地断电模式。

可用 CANMC 寄存器的配置位 WUBA 启用或禁用总线活动上的自动唤醒。如果在 CAN 总线上有任何操作，模块就开始其加电序列。模块一直等待直到它在 CANRX 引脚上检测到 11 个连续隐性位，之后它进入总线-有效状态。

注： 启动总线活动的第一个 CAN 消息不能被接收。这就意味着，在断电模式和自动唤醒模式收到的第一条消息丢失。

离开睡眠模式后，PDR 和 PDA 位被清除。CAN 错误计数器保持不变。

如果在 PDR 位被设定时模块正在传输消息，该传输继续，直至成功传输、丢失仲裁或在 CAN 总线线路上发生错误情况。然后，PDA 位被激活，这样该模块不会在 CAN 总线线路上引起错误状态。

为了执行本地断电模式，在 CAN 模块中使用了两个独立的时钟。一个时钟始终保持有效来确保断电操作（也就是说，唤醒逻辑和到 PDA (CANES.3) 位的写和读取访问）。其他时钟的启用取决于 PDR 位的设置。

5.2 进入和退出器件低功耗模式的预防措施 (LPM)

在外设时钟被关闭时，28x DSP 具有两种低功耗模式，STANDBY（待机）和 HALT（暂停）。由于 CAN 模块是通过网络连接到多个节点上，因此在进入和退出器件的低功耗模式前，如 STANDBY 和 HALT，您必须要小心。一个 CAN 数据包必须通过所有节点全部接收；因此，如果传输在该过程的中途被中止，被中止的数据包就会违反 CAN 协议，致使所有节点产生错误帧。退出 LPM 的节点应该这样安全操作。例如，如果一个节点退出 LPM 时在 CAN 总线上有数据流量，它可以“看到”一个被截断的数据包并用错误帧干扰总线。

在进入器件的低功耗模式前，必须考虑以下几点：

1. CAN 模块已经完成了最后一个请求数据包的传输。
2. CAN 模块已经向准备好进入 LPM 的 CPU 发送信号。

换言之，只有使 CAN 模块进入本地断电模式，器件才能进入低功耗模式。

5.3 启用或禁用到 CAN 模块的时钟

CAN 模块不能使用，除非到该模块的时钟处于启用状态。通过使用 PCLKCR 寄存器的为 14 来启用或禁用。此位对从不使用 CAN 模块的应用程序有用。在此类应用中，CAN 模块的时钟可以永久关闭，从而节省一些电能。这个位的用途不是为了把 CAN 模块置于低功耗模式，并且也不应被用于此目的。与其他所有外设相似，到 CAN 模块的时钟在复位时被禁用。

5.4 CAN 控制器模块的外部可能故障模式

本节列出了一些在基于 CAN 的系统中的潜在故障模式。列出的故障模式在 CAN 控制器之外，因此需要在系统级别上对其进行评估。

- CAN_H 和 CAN_L 短接
- CAN_H 和/或 CAN_L 短接至地
- CAN_H 和/或 CAN_L 短接至电源
- 故障 CAN 收发器
- 对 CAN 总线的电气干扰

重要声明

德州仪器(TI) 及其下属子公司有权在不事先通知的情况下, 随时对所提供的产品和服务进行更正、修改、增强、改进或其它更改, 并有权随时中止提供任何产品和服务。客户在下订单前应获取最新的相关信息, 并验证这些信息是否完整且是最新的。所有产品的销售都遵循在订单确认时所提供的TI 销售条款与条件。

TI 保证其所销售的硬件产品的性能符合TI 标准保修的适用规范。仅在TI 保证的范围内, 且TI 认为有必要时才会使用测试或其它质量控制技术。除非政府做出了硬性规定, 否则没有必要对每种产品的所有参数进行测试。

TI 对应用帮助或客户产品设计不承担任何义务。客户应对其使用TI 组件的产品和应用自行负责。为尽量减小与客户产品和应用相关的风险, 客户应提供充分的设计与操作安全措施。

TI 不对任何TI 专利权、版权、屏蔽作品权或其它与使用了TI 产品或服务的组合设备、机器、流程相关的TI 知识产权中授予的直接或隐含权限作出任何保证或解释。TI 所发布的与第三方产品或服务有关的信息, 不能构成从TI 获得使用这些产品或服务的许可、授权、或认可。使用此类信息可能需要获得第三方的专利权或其它知识产权方面的许可, 或是TI 的专利权或其它知识产权方面的许可。

对于TI 的产品手册或数据表, 仅在没有对内容进行任何篡改且带有相关授权、条件、限制和声明的情况下才允许进行复制。在复制信息的过程中对内容的篡改属于非法的、欺诈性商业行为。TI 对此类篡改过的文件不承担任何责任。

在转售TI 产品或服务时, 如果存在对产品或服务参数的虚假陈述, 则会失去相关TI 产品或服务的明示或暗示授权, 且这是非法的、欺诈性商业行为。TI 对此类虚假陈述不承担任何责任。

TI 产品未获得用于关键的安全应用中的授权, 例如生命支持应用(在该类应用中一旦TI 产品故障将预计造成重大的人员伤亡), 除非各方官员已经达成了专门管控此类使用的协议。购买者的购买行为即表示, 他们具备有关其应用安全以及规章衍生所需的所有专业技术和知识, 并且认可和同意, 尽管任何应用相关信息或支持仍可能由TI 提供, 但他们将独力负责满足在关键安全应用中使用其产品及TI 产品所需的所有法律、法规和安全相关要求。此外, 购买者必须全额赔偿因在此类关键安全应用中使用TI 产品而对TI 及其代表造成的损失。

TI 产品并非设计或专门用于军事/航空应用, 以及环境方面的产品, 除非TI 特别注明该产品属于“军用”或“增强型塑料”产品。只有TI 指定的军用产品才满足军用规格。购买者认可并同意, 对TI 未指定军用的产品进行军事方面的应用, 风险由购买者单独承担, 并且独力负责在此类相关使用中满足所有法律和法规要求。

TI 产品并非设计或专门用于汽车应用以及环境方面的产品, 除非TI 特别注明该产品符合ISO/TS 16949 要求。购买者认可并同意, 如果他们在汽车应用中使用任何未被指定的产品, TI 对未能满足应用所需要求不承担任何责任。

可访问以下URL 地址以获取有关其它TI 产品和应用解决方案的信息:

	产品		应用
数字音频	www.ti.com.cn/audio	通信与电信	www.ti.com.cn/telecom
放大器和线性器件	www.ti.com.cn/amplifiers	计算机及周边	www.ti.com.cn/computer
数据转换器	www.ti.com.cn/dataconverters	消费电子	www.ti.com/consumer-apps
DLP® 产品	www.dlp.com	能源	www.ti.com/energy
DSP - 数字信号处理器	www.ti.com.cn/dsp	工业应用	www.ti.com.cn/industrial
时钟和计时器	www.ti.com.cn/clockandtimers	医疗电子	www.ti.com.cn/medical
接口	www.ti.com.cn/interface	安防应用	www.ti.com.cn/security
逻辑	www.ti.com.cn/logic	汽车电子	www.ti.com.cn/automotive
电源管理	www.ti.com.cn/power	视频和影像	www.ti.com.cn/video
微控制器 (MCU)	www.ti.com.cn/microcontrollers		
RFID 系统	www.ti.com.cn/rfidsys		
OMAP 机动性处理器	www.ti.com/omap		
无线连通性	www.ti.com.cn/wirelessconnectivity		
	德州仪器在线技术支持社区		www.deyisupport.com

邮寄地址: 上海市浦东新区世纪大道 1568 号, 中建大厦 32 楼 邮政编码: 200122
Copyright © 2012 德州仪器 半导体技术 (上海) 有限公司