

1 简介

恩智浦的可扩展控制器局域网 (MSCAN) 是博世规范中定义的通信控制器。为了充分了解 MSCAN 规范，建议用户先阅读博世规范。

KE16Z MCU 的 MSCAN 模块支持标准和扩展的数据帧，0 到 8 个字节数据长度，可编程的高达 1 Mbps 的比特率，支持远程帧等。有关更多特性，请参考 KE1xZP48M48SF0RM。

本应用笔记讲述了有关 MSCAN 的比特率计算和标识符仲裁。本文档还提供了两个 MSCAN 节点之间通信的应用案例，帮助用户理解如何使用 MSCAN 模块。用户可参考并修改 MSCAN 配置，按照自己的特定需求，以实现 MSCAN 节点之间的通信。

该示例的实现基于 IAR 8.30.1 开发环境，KE16Z SDK 2.4.0 软件，FRDM-KE16Z 开发板。

2 特性介绍

2.1 MSCAN 模块

2.1.1 概述

MSCAN 是实现了博世规范中定义的 CAN 2.0A/B 协议的一种通信控制器。MSCAN 使用先进的缓冲区排列，可实现实时行为的预测和应用软件的简化。

MSCAN 有三个具有内部优先级的发送缓冲区，使用本地优先级概念，允许预先设置多个报文并实现优化性能。它有五个带有 FIFO 存储方案的接收缓冲区来存储接收到的报文。

MSCAN 模块具有灵活的标识符滤波器，可通过标识符 (ID) 仲裁过程选择传入的报文。一次成功的传输或带有匹配标识符的报文被接收，都会产生标志位并生成一个向 CPU 发出的中断请求。

MSCAN 的基本特点如下：

- 支持 CAN 协议规范版本 2.0A/B。
- 有五个带 FIFO 存储的接收缓冲器。
- 三个具有本地优先级概念的传输缓冲区。
- 灵活的可掩蔽标识符接收滤波器。
- 支持内部时间戳。
- 具有内部低通滤波器的唤醒功能。
- 用于自检的循环模式。
- 监视总线的只听模式。

目录

1	简介.....	1
2	特性介绍.....	1
2.1	MSCAN 模块.....	1
2.2	MSCAN 比特率.....	3
2.3	MSCAN 标识符接收滤波器 and 屏蔽滤波器.....	6
3	应用细节.....	11
3.1	示例操作.....	11
3.2	MSCAN 节点配置.....	12
4	结论.....	14
5	参考资料.....	14



2.1.2 CAN 节点

CAN 是连接电子控制单元 (ECU) 的多主机串行总线标准，也称为节点。当 CAN 节点之间通信时，它们必须通过收发器设备连接到双线的 CAN 总线。连接是阻抗为 $120\ \Omega$ 双绞线。收发器能够驱动 CAN 总线所需的大电流，并可以对 CAN 节点进行电流保护。图 1 显示的是 CAN 节点与 CAN 总线的连接图。

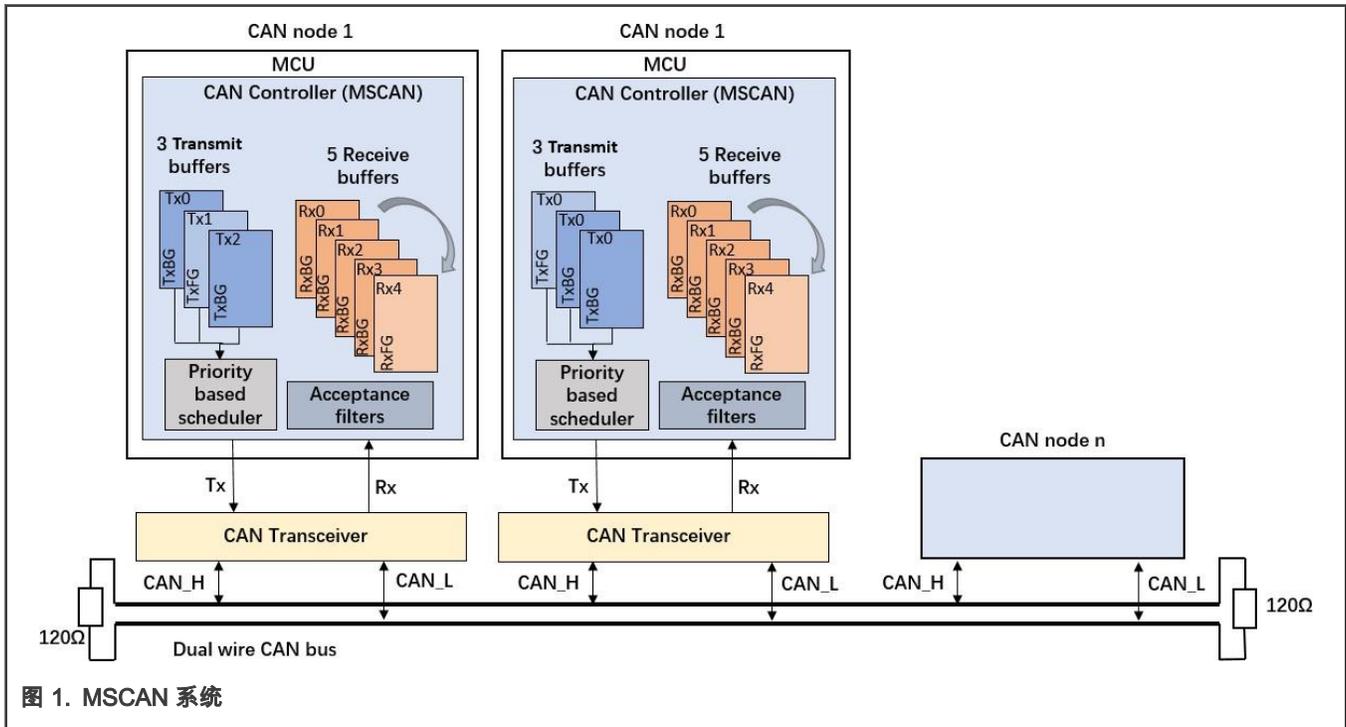


图 1. MSCAN 系统

2.1.3 CAN 帧结构

MSCAN 节点之间的报文传输由数据帧、远程帧、错误帧和过载帧四种不同的帧类型表示和控制。本应用笔记是基于数据帧的使用的。

报文传输有两种帧格式，它们的标识符字段的长度不一样。具有 11 位标识符的帧称为标准帧，具有 29 位标识符的帧称为扩展帧。数据帧包含标准帧格式和扩展帧格式两种帧格式。

数据帧用于传输最大八个字节的用户数据。根据 CAN 通信协议可知，一个数据帧的结构是由七个不同的部分组成的，包括起始帧、仲裁场、控制场、数据场、CRC 场、应答场、帧结尾。图 2 显示了标准数据帧和扩展数据帧的结构。

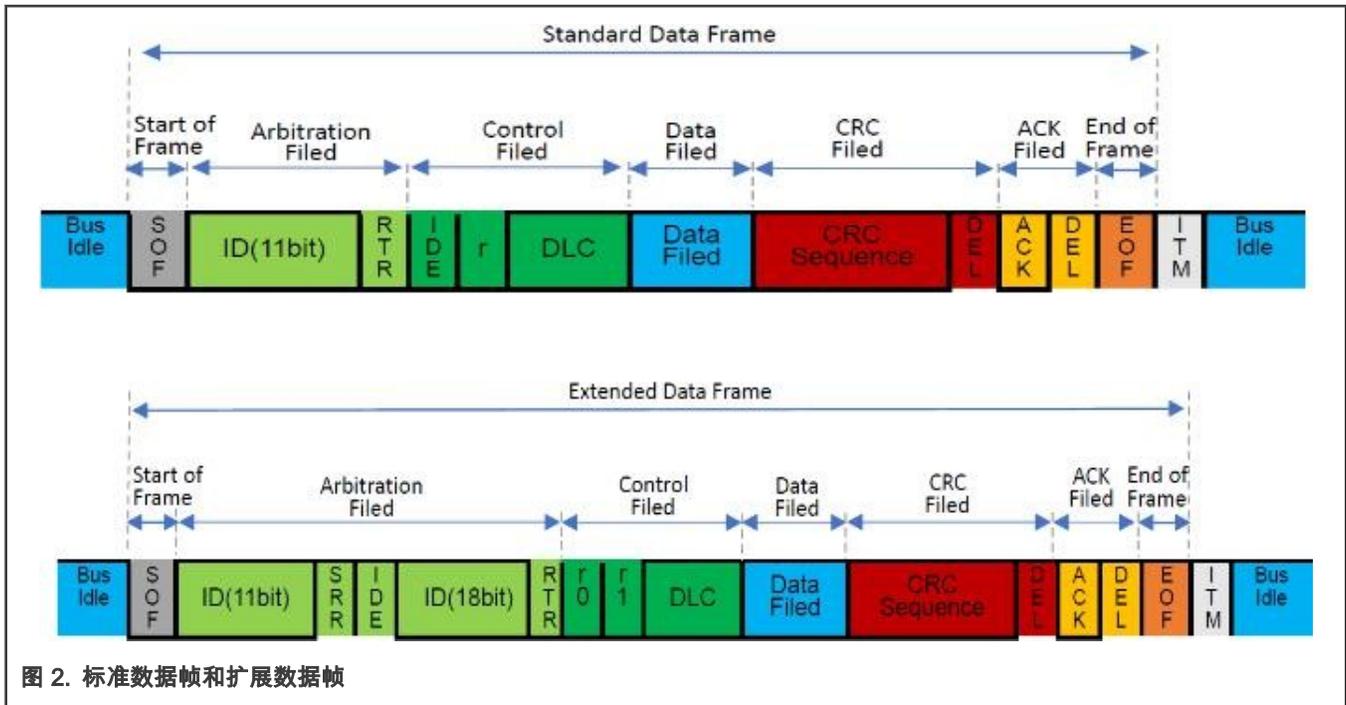


图 2. 标准数据帧和扩展数据帧

2.2 MSCAN 比特率

2.2.1 时钟选择

MSCAN 模块具有可编程时钟源。它可以是总线时钟或外部振荡器时钟 (SOSCDIV2_CLK)。图 3 显示了 MSCAN 框图，它可以作为计算比特率的参考过程。

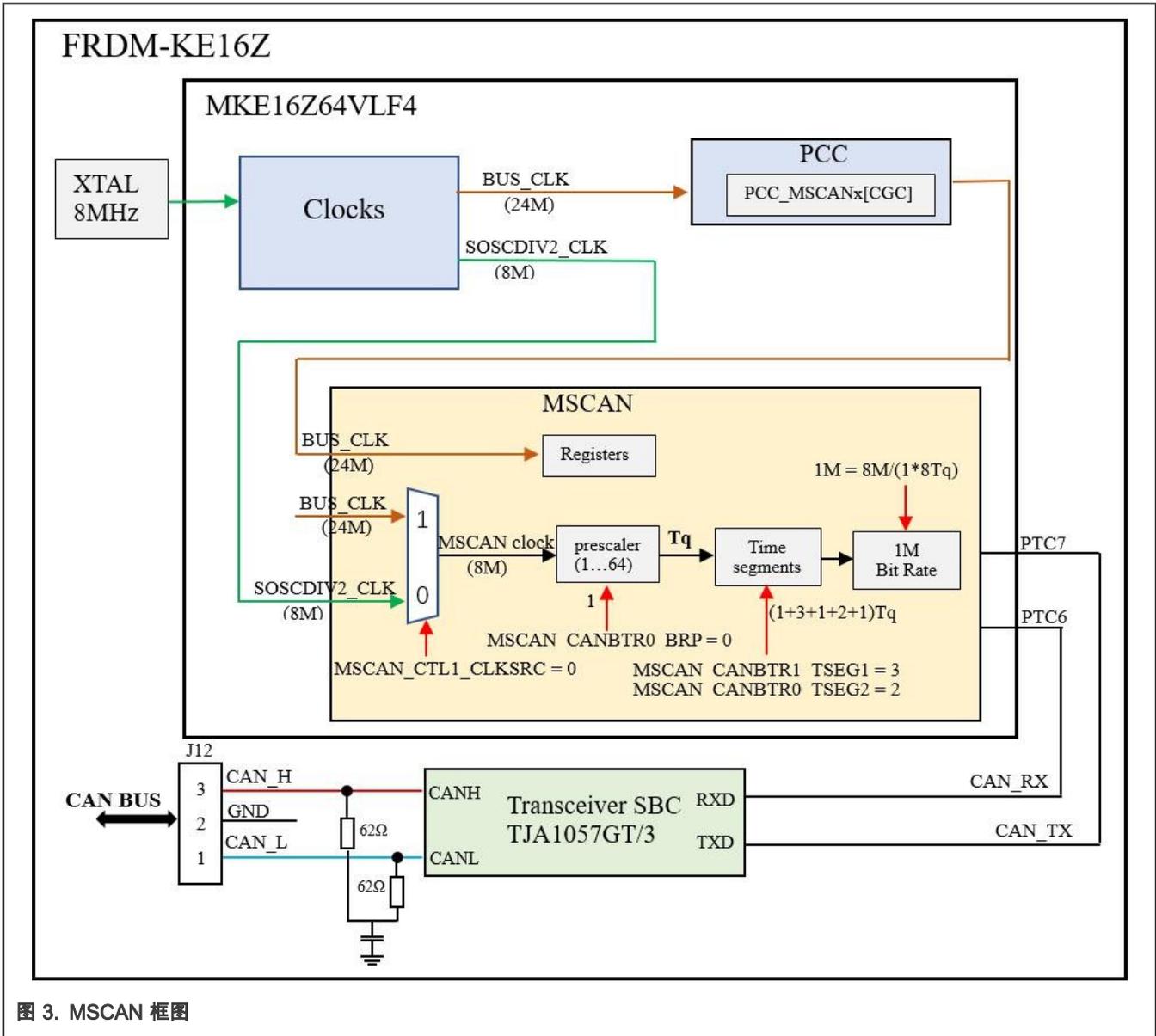


图 3. MSCAN 框图

如 图 3 所示，用户可以配置 MSCAN_CANCTL1_CLKSRC 位来选择 MSCAN 时钟源 (f_{CANCLK})。选择好时钟源后，配置 MSCAN_CANBTR0_BRP 位来选择比特率预分频器。时间份额 (T_q) 是 MSCAN 使用的固定时间单元。它可以通过选定的时钟和预分频器值来计算，如 Equation 1 所示。

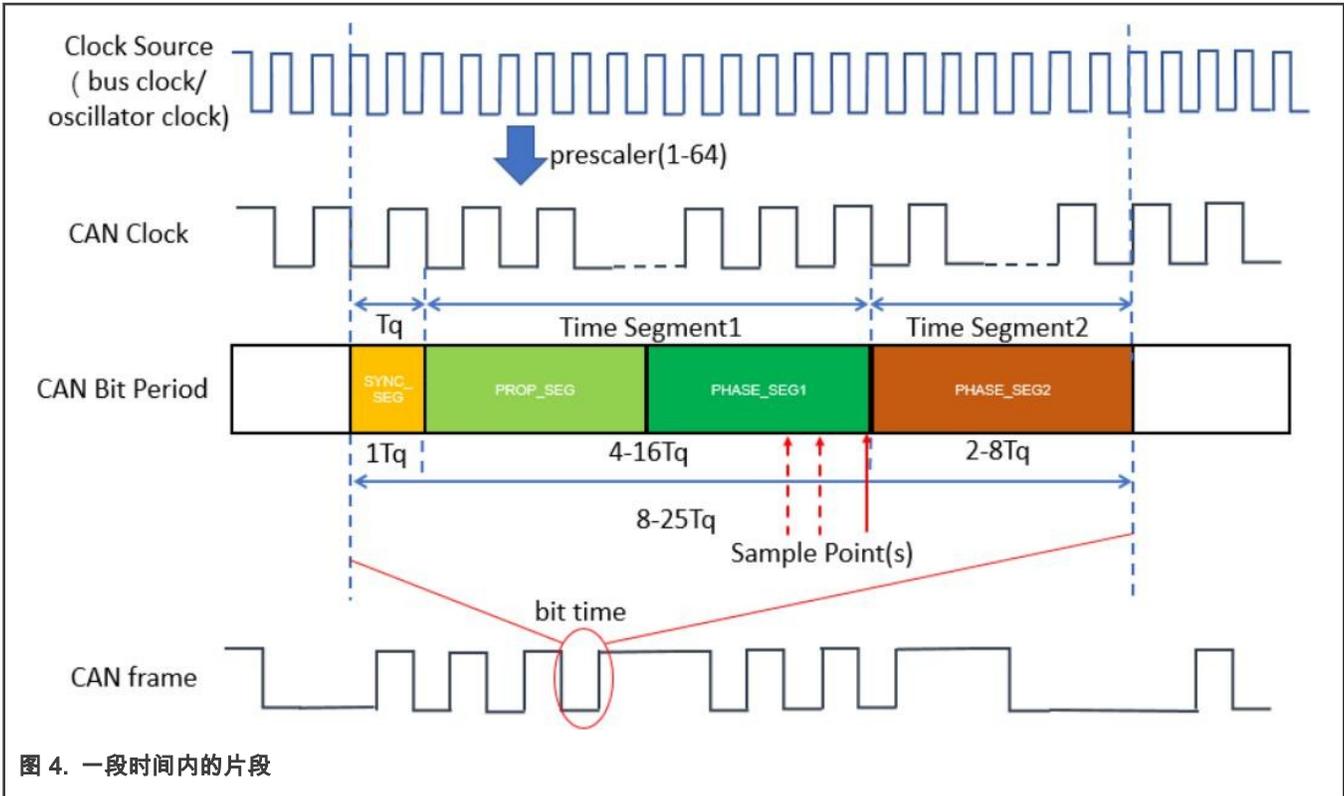
$$T_q = \frac{f_{CANCLK}}{(\text{prescaler value})}$$

Equation 1. T_q 计算

2.2.2 位时间

通过配置总线时序寄存器 MSCAN_CANBTR0 和 MSCAN_CANBTR1，获得 MSCAN 比特时间。位时间被细分为三个段：SYNC_SEG、时间段 1 和时间段 2。这些段可以用时间份额 (T_q) 表示。通过配置每个段中的 T_q 数来确定和更改位时间。

图 4 显示了一个位时间内的时间段组成信息。



Equation 2 显示了计算位时间的方法。

$$bit\ time = \frac{(SYNC_SEG + Time\ Segment\ 1 + Time\ Segment\ 2) * (prescaler\ value)}{f_{CANCLK}}$$

Equation 2. 位时间计算

式中：

- SYNC_SEG 为 1 个时间份额。根据 CAN 标准协议，时间段 1 包括 PROP_SEG (传播时间段) 和 PHASE_SEG1 (相位缓冲段 1)。可以通过配置 MSCAN_CANBTR1_TSEG1 寄存器将时间段 1 设置为 4 到 16 个时间份额。
- 根据 CAN 标准协议，时间段 2 代表 PHASE_SEG2 (相位缓冲段 2)。可以通过配置 MSCAN_CANBTR1_TSEG2 寄存器将时间段 2 设置为 2 到 8 个时间份额。

Equation 3 为比特率的计算方法。

$$bit\ rate = \frac{1}{bit\ time} = \frac{f_{CANCLK}}{((prescaler\ value) * (1 + Time\ Segment\ 1 + Time\ Segment\ 2))}$$

Equation 3. 比特率计算

根据 Equation 3，图 4 中有一个示例展示了如何通过选择外部振荡器时钟作为 MSCAN 时钟来获得 1 M 比特率。因此，选择合适的参数配置 MSCAN_CANCTL1_CLKSRC，MSCAN_CANBTR0_BRP，MSCAN_CANBTR1_TSEG1，和 MSCAN_CANBTR1_TSEG2 寄存器可以自定义用户所需的比特率。

2.2.3 同步跳转宽度

同步跳跃宽度 (SJW) 定义了可以缩短或延长的最大 T_q 时钟周期数，以实现 CAN 总线上数据转换的重新同步。

由于每个单元都在自己的时钟上运行，时钟误差是可以积累的，PHASE_SEG1 和 PHASE_SEG2 可以用来吸收它们。通过延长 PHASE_SEG1 或缩短 PHASE_SEG2，吸收误差来调整同步，这些时间段可以调整的最大值为 SJW 值。增加 SJW 后，允许误差增加，但降低了通信速度。

通过设置 SJW 参数，同步跳转宽度（详细信息请参考博世 CAN2.0A/B 规范）可以被配置成 1 到 4 范围内的时间份额。表 1 显示了和博世 CAN2.0A/B 规范兼容的时间段设置和相关参数取值。

表 1. 博世 CAN2.0A/B 兼容的位时间段设置

时间段 1	TSEG1	时间段 2	TSEG2	同步跳转宽度	SJW
5..10	4..9	2	1	1..2	0..1
4..11	3..10	3	2	1..3	0..2
5..12	4..11	4	3	1..4	0..3
6..13	5..12	5	4	1..4	0..3
7..14	6..13	6	5	1..4	0..3
8..15	7..14	7	6	1..4	0..3
9..16	8..15	8	7	1..4	0..3

2.3 MSCAN 标识符接收滤波器和屏蔽滤波器

当一个 MSCAN 节点接收来自其它节点的报文时，将接收到的每个报文都写入后台接收缓冲区（RxBG）。如果报文位移到前台接收缓冲区（RxFG），能通过标识符仲裁（设置的标识符接收寄存器和标识符掩码寄存器实现），则被 CPU 读取并接收。否则，该报文被下一条报文覆盖。

2.3.1 可扩展的 MSCAN 滤波器

在 MSCAN 模块中，灵活的可编程标识符滤波器可以减少接收到的报文的评估时间，防止 CPU 处理某些不需要的报文。

MSCAN 标识符接收寄存器定义标准或扩展标识符（ID[10 : 0] 或 ID[28 : 0]）的可接收模式。标识符掩码寄存器定义标识符接收寄存器中的哪些位需要关心或者不关心。

如果在标识符掩码寄存器（MSCAN_CANIDMRn）中，将接收掩码位设置为 0，意味着在对应的位中，期望的标识符必须与标识符接收寄存器（MSCAN_CANIDARn）中的接收码的此位相同。如果在标识符掩码寄存器（MSCAN_CANIDMRn）中，将接收掩码位设置为 1，则期望的标识符可与标识符接收寄存器（MSCAN_CANIDARn）中的接收码的此位不匹配。

MSCAN 模块有八对滤波器寄存器。每对由 8 位掩码寄存器和 8 位接收寄存器组成。在 ID 仲裁过程之前，可以配置 MSCAN_CANIDAC_IDAM 位的值来更改滤波器寄存器的数量。

滤波器寄存器可被配置成下面四种操作模式：

- 两个 32 位标识符滤波器。
- 四个 16 位标识符滤波器。
- 八个 8 位标识符滤波器。
- 关闭滤波器。

2.3.2 仲裁过程

本节讲述了在不同的滤波器操作模式下，对 11 位 ID 或 29 位 ID 的报文仲裁的过程。

MSCAN 模块设置标识符命中标志（IDHIT[2 : 0]）以指示标识符接收命中。IDHIT 指示器总是与前台缓冲区（Rx FG）中的报文相关，当报文被位移到接收器 FIFO 的前台缓冲区时，指示器也会被更新。

2.3.2.1 两个 32 位标识符滤波模式模式

使用两个 32 位滤波器检查扩展报文报文的 29 位 ID、RTR 位、IDE 位和 SRR 位。这些位中的任何一位都可以先在 CANIDMR0-CANIDMR3 寄存器中被标记为“无关”位，然后与 CANIDAR0-CANIDAR3 寄存器的内容进行比较，如果匹配，则会产生滤波器 0

命中。如果不匹配，这些位继续与 CANIDMR4-CANIDMR7 和 CANIDAR4-CANIDAR7 进行比较，如果匹配，则会产生滤波器 1 命中。至此，如果没有产生命中，则接收到的扩展报文将被丢弃。

对于接收到的标准报文，会检查 11 位 ID、RTR 位、IDE 位。类似地，这些位与 CANIDMR0-CANIDMR1 寄存器和 CANIDAR0-CANIDAR1 寄存器的内容进行比较，如果匹配，则发生滤波器 0 命中。否则，这些位将继续与 CANIDMR4-CANIDMR5 和 CANIDAR4-CANIDAR5 进行比较，如果它们匹配，则会发生滤波器 1 命中。如果没有命中，接收到的标准报文将被丢弃。

图 5 显示了第一群组标识符滤波器仲裁过程。第二群组滤波器的仲裁流程类似。

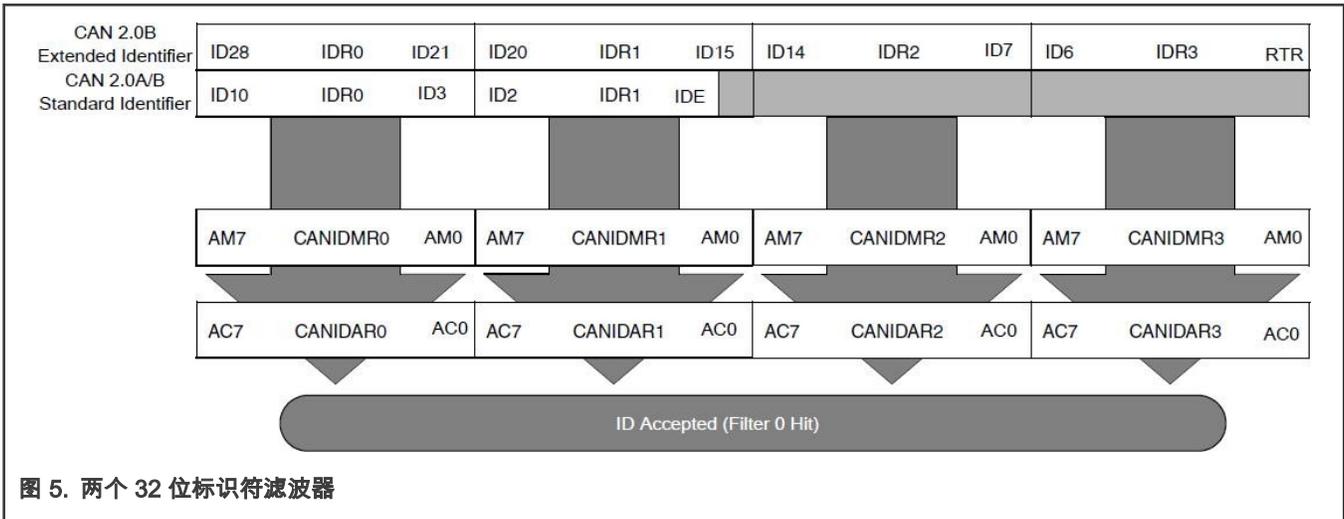


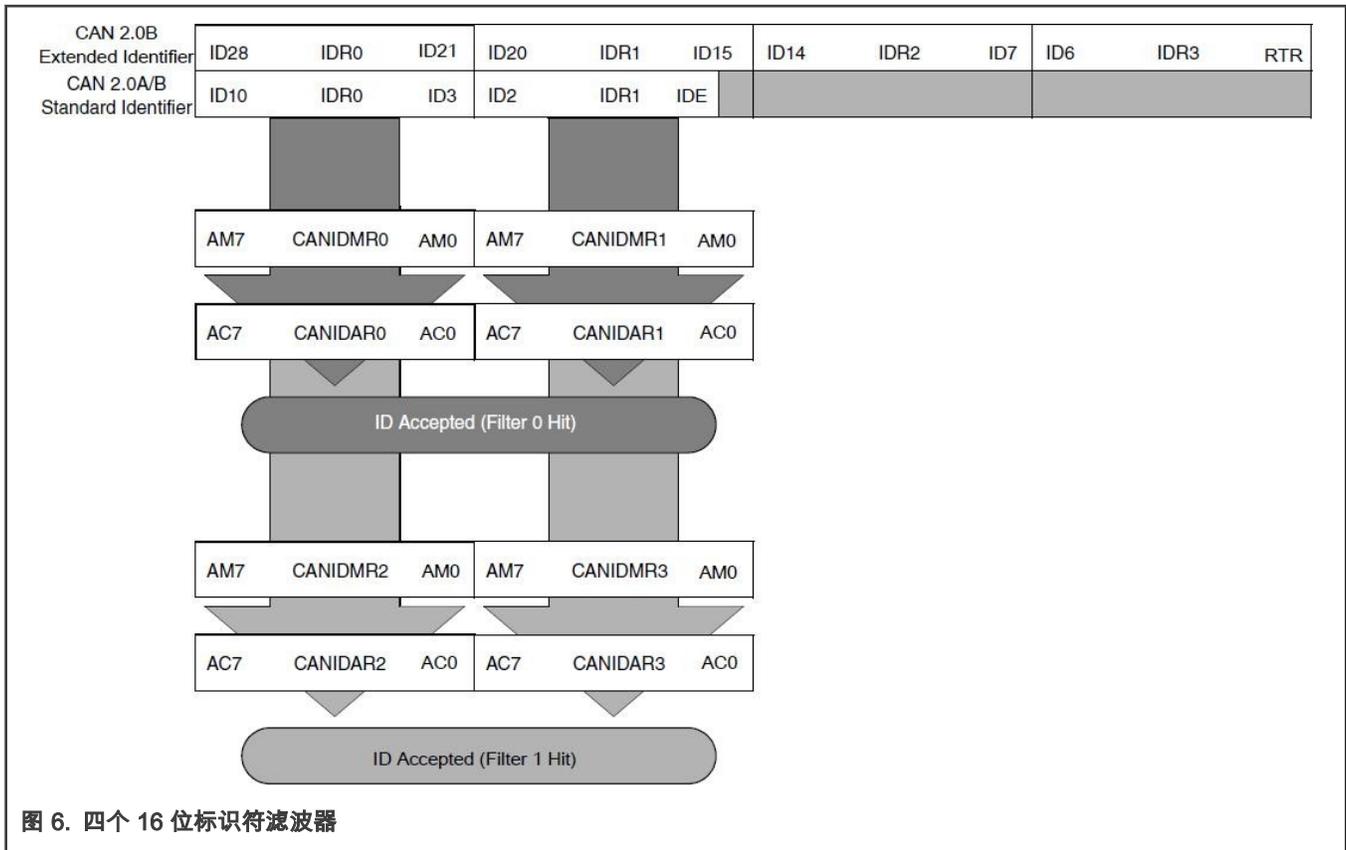
图 5. 两个 32 位标识符滤波器

2.3.2.2 四个 16 位标识符滤波器模式

使用四个 16 位滤波器检查扩展报文的高 14 位报文 ID、IDE 位和 SRR 位。这些位中的任何一位都可以先通过 CANIDMR0-CANIDMR1 寄存器被标记为“无关”位。然后再与 CANIDAR0-CANIDAR1 寄存器的内容进行比较，如果匹配，则会产生滤波器 0 命中。如果不匹配，扩展报文的这些位继续与 CANIDMR2-CANIDMR3 和 CANIDAR2-CANIDAR3 进行比较，滤波器如果匹配，滤波器 1 将发生命中。如果滤波器 1 没有命中，这些位将继续与 CANIDMR4-CANIDMR5 和 CANIDAR4-CANIDAR5 进行比较，如果匹配，则将发生滤波器 2 命中。如果滤波器 2 没有命中，这些位将继续与 CANIDMR6-CANIDMR7 和 CANIDAR6-CANIDAR7 进行比较，如果它们匹配，则将发生滤波器 3 命中。至此，如果仍然没有滤波器命中，接收到的扩展报文将被丢弃。

如果接收到标准报文，则检查标准报文的 11 位 ID、RTR 位和 IDE 位。标准消息与扩展报文在四种 16 位标识符滤波器模式下具有相同的仲裁过程。

图 6 显示了第一群组标识符滤波器仲裁的过程。第二群组仲裁流程类似。



2.3.2.3 八个 8 位标识符滤波器的模式

八个 8 位滤波器检查标准报文和扩展报文的 ID 的高 8 位。标准报文和扩展报文在八个 8 位标识符模式下具有相同的仲裁过程。将 ID 的高 8 位与八个滤波器组（掩码寄存器：CANIDMRn，接收寄存器：CANIDARn）进行比较，以产生滤波器 0 到滤波器 7 的命中。只要有一次滤波器命中产生，报文都会被接收。没有滤波器命中，消息怎会被丢弃。否则数据丢失。

图 7 显示了第一群组标识符滤波器仲裁的过程。第二群组仲裁的过程类似。

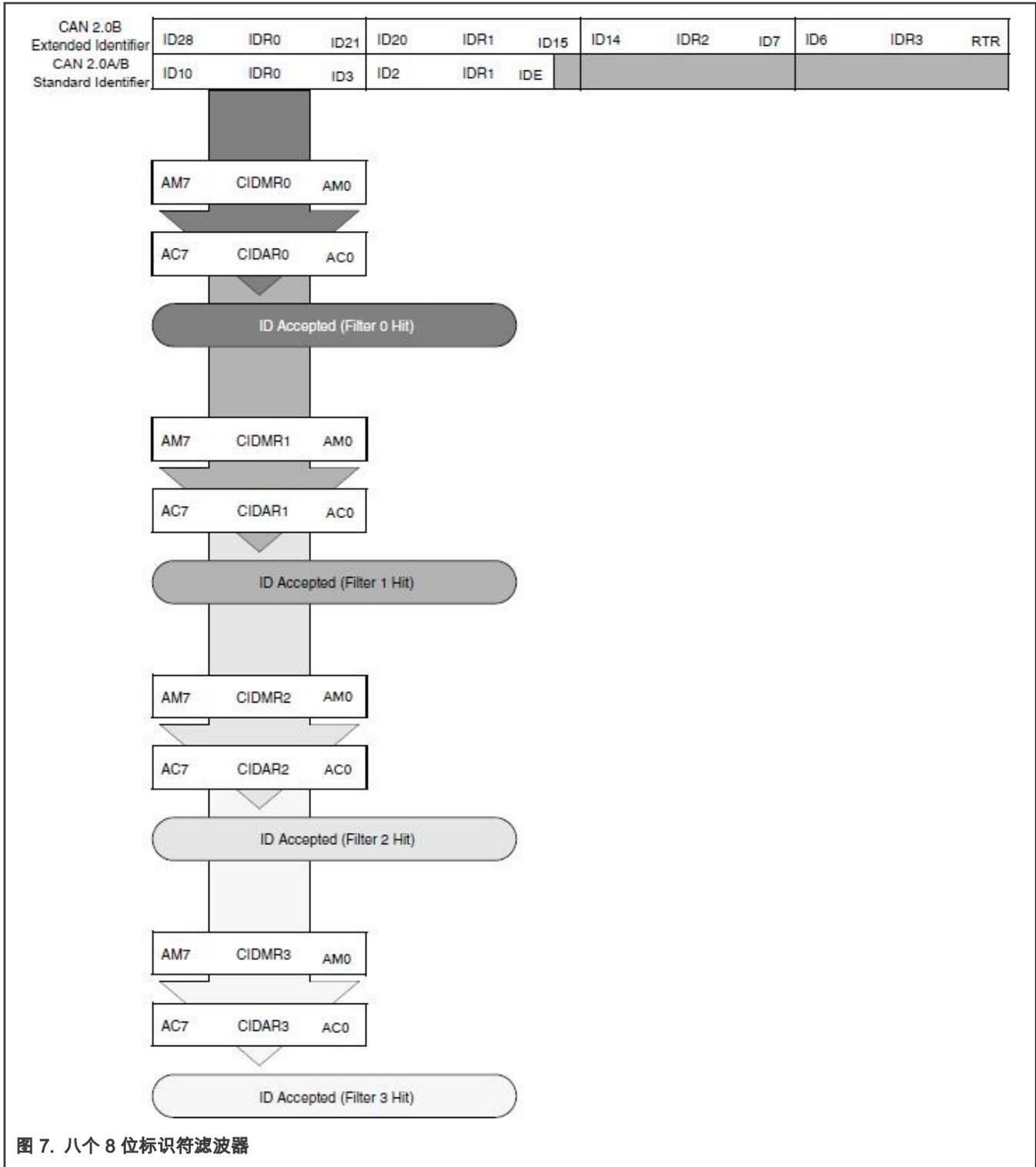


图 7. 八个 8 位标识符滤波器

2.3.3 标识符寄存器的配置

在 MSCAN 节点通信过程中，必须为每个节点设置适当的标识符。MSCAN 模块有四个扩展标识符寄存器和两个标准标识符寄存器，用于来写入接收到的标识符。下面描述了如何在两个 32 位标识符滤波器模式下为 MSCAN 节点通信配置适当的标识符。

2.3.3.1 具有特定标识符的标准数据帧

如果 MSCAN 节点想要从具有标识符 0x320 到 0x323 的任意标准数据帧接收报文，那么现在它可以通过设置第一组标识符滤波器来实现该功能。图 8 显示了标准标识符寄存器结构和标识符滤波器配置的过程。

	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RSRTR	RSIDE	Reserved			
0x321	1	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	
0x322	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	
0x323	1	1	0	0	0	1	0	0	0	1	1	0	0	0	0	0	
Filter Value	1	1	0	0	0	1	0	0	0	X	X	0	0	0	0	0	
CANIDMR (0-1)	AM7				CANIDMR0				AM0				CANIDMR1				AM0
	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
CANIDAR (0-1)	AC7				CANIDAR0				AC0				CANIDAR1				AC0
	1	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0

图 8. 标准数据帧的 MSCAN 标识符滤波器寄存器配置

如图 8 所示，为了获得期望的标准数据标识符，将 CANIDMR0-CANIDMR1 寄存器中的特定位设置为 0，这表明待接收标识符相应位必须和 CANIDAR0-CANIDAR1 寄存器中的相应位相同。在本节中，CANIDAR0-CANIDAR1 寄存器可以设置为 0x321、0x322 或 0x323。将 CANIDMR0-CANIDMR1 寄存器中的位设置为 1，可以用于定义哪个位不用于比较。

当 CANIDMR0-CANIDMR1 寄存器和 CANIDAR0-CANIDAR1 寄存器的配置完成后，图 8 中，Filter Value 显示的是可以接收的 ID 值。在 Filter Value 中，X 表示此位可以是 0 或 1。那么，CANIDAR1 寄存器中的第 6 位 (ID0) 和第 7 位 (ID1) 可以是 1 或 0，因为这两个位已经被设置为“无关”的。当接收到的报文具有标识符 0x321、0x322 或 0x323 时，则产生滤波器 0 击中，该报文将被接收。

2.3.3.2 具有特定标识符的扩展数据帧

如果 MSCAN 节点想要从具有标识符 0x801 或 0x802 的任意扩展数据帧接收报文，则可以通过设置第一组标识符滤波器来实现该功能。图 9 显示了扩展标识符寄存器结构和标识符滤波器配置的过程。

0x801	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18	RSRR	REIDE	ID17	ID16	ID15	
	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	
	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RERTR	
	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	
0x802	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18	RSRR	REIDE	ID17	ID16	ID15	
	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	
	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RERTR	
	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	
Filter Value	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18	RSRR	REIDE	ID17	ID16	ID15	
	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	
	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RERTR	
	0	0	0	1	0	0	0	0	0	0	0	0	0	X	X	0	
CANIDMR (0-3)	AM7				CANIDMR0				AM0				CANIDMR1				AM0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	AM7				CANIDMR2				AM0				CANIDMR3				AM0
	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	
CANIDAR (0-3)	AC7				CANIDAR0				AC0				CANIDAR1				AC0
	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	
	AC7				CANIDAR2				AC0				CANIDAR3				AC0
	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	

图 9. 扩展数据帧的 MSCAN 标识符滤波器寄存器配置

如 [图 9](#) 所示，为了获得期望扩展数据标识符(0x801, 0x802)，将 CANIDMR0-CANIDMR3 寄存器中的特定位设置为 0，这表明待接收标识符的相应位必须和 CANIDAR0-CANIDAR3 寄存器中的相应位相同。在本节中，CANIDAR0-CANIDAR3 寄存器可以设置为 0x801 或 0x802。将 CANIDMR0-CANIDMR3 寄存器中的特定位设置为 1，可以用于定义哪个位不用于比较。

当 CANIDMR0-CANIDMR3 寄存器和 CANIDAR0-CANIDAR3 寄存器的配置完成后，[图 9](#) 中，Filter Value 显示的是可以接收的 ID 值。在 Filter Value 中，X 表示此位可以是 0 或 1。那么，在 CANIDAR3 寄存器中的第 1 位 (ID0) 和第 2 位 (ID1) 可以是 1 或 0，因为这两个位已经被设置为“无关”。当接收到的报文具有标识符 0x801 或 0x802 时，则产生滤波器 0 命中，该报文将被接收。

3 应用细节

下面是两个 MSCAN 节点之间通信的一个使用案例。在本例中，MSCAN 节点 1 向 MSCAN 节点 2 发送报文，并接收来自 MSCAN 节点 2 的报文。同时，MSCAN 节点 2 向 MSCAN 节点 1 发送报文，并接收来自 MSCAN 节点 1 的报文。

3.1 示例操作

软件执行之前务必确认硬件连接。准备两块 KE16Z 开发板，按照 [表 2](#) 连接板子之间的 MSCAN 引脚。

表 2. 连接两个开发板上的 MSCAN 引脚

BOARD_1	Connects to		BOARD_2
Pin Name	Board location	Board location	Pin name
CAN_L	J12-1	J12-1	CAN_L
CAN_H	J12-3	J12-3	CAN_H
GND	J12-2	J12-2	GND

执行以下步骤可以实现两块板子之间的通信。

1. 如 [图 10](#) 所示，通过 USB 线将两块 KE16Z 开发板 (节点 1, 节点 2) 和 PC 连接供电。

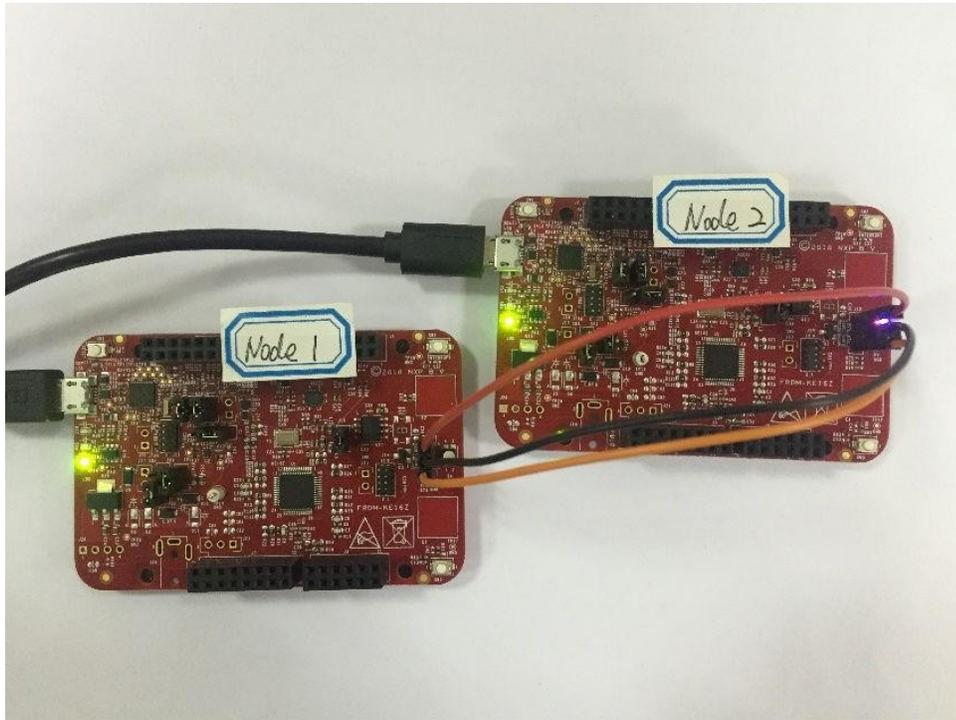


图 10. 两个 FRDM-KE16Z 板的连接

2. 打开两个串口终端工具，将波特率设置为 115200，以显示 MSCAN 节点代码的输出。
3. 将两个程序分别下载到两个开发板上。
4. 按下板上的复位按键，或者在 IDE 中启动调试器来运行演示代码。

请参考 MSCAN 节点配置来获取两个 MSCAN 节点代码的详细配置信息。

3.2 MSCAN 节点配置

配置 MSCAN 节点 1 接收 MSCAN 节点 2 发送过来的数据，然后再将数据传输到 MSCAN 节点 2。同样的，配置 MSCAN 节点 2 以接收 MSCAN 节点 1 发送过来的数据，然后再将数据传输到 MSCAN 节点 1。本节讲述 MSCAN 节点 1 发送和接收报文的详细过程。

3.2.1 ID 接受和掩码滤波器的定义

在本应用示例中，节点 1 被配置为 ID 为 0x801 的扩展帧。它从节点 2 接收 ID 为 0x802 的报文。在两个 32 位标识符滤波器的模式下，使用宏定义填充掩码和接收寄存器 (MSCAN_CANIDARn, MSCAN_CANIDMRn) 的值。

类似地，节点 2 被配置为 ID 为 0x802 的扩展帧。它从节点 1 接收来自 ID 为 0x801 的报文。以节点 1 为例，下面显示了宏定义接收和掩码滤波器。

```

/* Definitions for node1 ID Acceptance and Mask filter */
#define NODE_ID1 0x801 /* Transmit Extend ID */
#define NODE_ID2 0x802 /* Receive Extend ID */
#define MSCAN_IDAR0 ((NODE_ID2>>21)<<24)|(((NODE_ID2>>18)&0x7)<<21)\
|(((NODE_ID2>>15)&0x7)<<16)|(((NODE_ID2>>7)&0xFF)<<8)\
|(((NODE_ID2>>0)&0x7F)<<1)|(1<<19) /* ID Acceptance Registers of First Bank
*/
#define MSCAN_IDAR1 ((NODE_ID2>>21)<<24)|(((NODE_ID2>>18)&0x7)<<21)\
|(((NODE_ID2>>15)&0x7)<<16)|(((NODE_ID2>>7)&0xFF)<<8)\
|(((NODE_ID2>>0)&0x7F)<<1)|(1<<19) /* ID Acceptance Registers of second Bank
*/

```

```
#define MSCAN_IDMR0 0x1|(uint32_t)0x18<<16 /* ID Mask Registers of First Bank */
#define MSCAN_IDMR1 0x1|(uint32_t)0x18<<16 /* ID Mask Registers of Second Bank */
```

3.2.2 MSCAN 初始化配置

选择时钟源，配置比特率，以保证通信速度。选择标识符的筛选模式，配置接收滤波器和掩码滤波器，以确定待接收标识符。

下面以节点 1 为例，来展示初始配置的方法。

```
/* Initialize MSCAN Module config struct with default value. */
MSCAN_GetDefaultConfig(&mscanConfig); //bit rate:1M; clock source:Oscillator clock; loopback
mode:disabled

//Identifier acceptance mode: two 32-bit filters

/* Acceptance filter and Mask filter configuration. */
mscanConfig.filterConfig.u32IDAR0 = MSCAN_IDAR0; // Configure the first bank of Acceptance filter
mscanConfig.filterConfig.u32IDAR1 = MSCAN_IDAR1; // Configure the second bank of Acceptance filter
mscanConfig.filterConfig.u32IDMR0 = MSCAN_IDMR0; // Configure the first bank of Mask filter
mscanConfig.filterConfig.u32IDMR1 = MSCAN_IDMR1; // Configure the second bank of Mask filter
/* Initialize MSCAN module.*/
MSCAN_Init(EXAMPLE_MSCAN, &mscanConfig, EXAMPLE_MSCAN_CLK_FREQ);

/* Enable Rx Buffer interrupt.*/
MSCAN_EnableRxInterrupts(EXAMPLE_MSCAN, kMSCAN_RxFullInterruptEnable);
MSCAN_EnableRxInterrupts(EXAMPLE_MSCAN, kMSCAN_OverrunInterruptEnable);
MSCAN_EnableRxInterrupts(EXAMPLE_MSCAN, kMSCAN_StatusChangeInterruptEnable);
EnableIRQ(EXAMPLE_MSCAN_IRQn);
```

3.2.3 发送数据

当向其它节点发送报文时，将 MSCAN 的帧结构配置为数据帧，并选择标准格式或扩展格式到数据帧。报文可以在完成数据场，仲裁场以及其它配置后发出。

以节点 1 为例，下面介绍了发送数据的准备工作。

```
/*Prepare Tx Frame for sending. */
txFrame.ID_Type.ID = NODE_ID1; // Configure the transfer message identifier to NODE_ID1
txFrame.format = kMSCAN_FrameFormatExtend; // Configure the transfer message format to extended ID
txFrame.type = kMSCAN_FrameTypeData; // Configure the MSCAN frame to data frame
txFrame.DLR = 8; // Configure the data length to 8
txFrame.dataWord0 = 0x44332211; // Configure the transfer data
txFrame.dataWord1 = 0x88776655;

/* Send data through Tx Buffer using polling function. */
MSCAN_TransferSendBlocking(EXAMPLE_MSCAN, &txFrame);
```

3.2.4 接收数据

接收新报文时，读取数据缓冲区，在接收器中断函数中清除接收标志位，以保持模块可用于接收更多的报文。

以节点 1 为例，下面展示接收数据中断处理程序的过程。

```
/* Receive Data interrupt handler */
void EXAMPLE_MSCAN_IRQHandler(void)
{
    /* If new data arrived.*/
    if (MSCAN_GetRxBufferFullFlag(EXAMPLE_MSCAN))
    {
        MSCAN_ReadRxMb(EXAMPLE_MSCAN, &rxFrame); // Read message received from acceptance nodes
    }
}
```

```
MSCAN_ClearRxBufferFullFlag(EXAMPLE_MSCAN); // Clear Rx flag
}
/* Add for ARM errata 838869, affects Cortex-M4, Cortex-M4F Store immediate overlapping
exception return operation might vector to incorrect interrupt */
#if defined __CORTEX_M && (__CORTEX_M == 4U)
  __DSB();
#endif
}
```

4 结论

本应用笔记讲述了 MSCAN 模块的重要特性、比特率计算和标识符仲裁。还提供了一个关于两个节点通信的应用示例，以帮助用户了解如何使用 MSCAN 模块。用户可参考本文的配置轻松实现 MSCAN 的功能，并可根据设计需要修改相应配置。

5 参考资料

您可以在 NXP 官方网站上查阅以下文档以获得更多信息:

- *Kinetis KE1xZ64 Sub-Family Reference Manual* (文档 [KE1xZP48M48SF0RM](#))
- *Using MSCAN on the HCS12 Family* (文档 [AN3034](#))
- *XGATE Library: CAN Driver Providing a Full CAN Mailbox System* (文档 [AN2726](#))
- *Bosch Controller Area Network (CAN) Version 2.0 Protocol Standard* (文档 [BCANPSV2](#))
- *S32K1xx Series Cookbook* (文档 [AN5413](#))

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

Limited warranty and liability — Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

Right to make changes - NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer’s applications and products. Customer’s responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer’s applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, ICODE, JCOP, LIFE, VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, CodeWarrior, ColdFire, ColdFire+, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeScale, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org. M, M Mobileye and other Mobileye trademarks or logos appearing herein are trademarks of Mobileye Vision Technologies Ltd. in the United States, the EU and/or other jurisdictions.

© NXP B.V. 2019-2021.

All rights reserved.

For more information, please visit: <http://www.nxp.com>

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 2019 年 2 月 25 日

Document identifier: AN12355

