
PIC18 CAN FD 模块的基本配置

简介

控制器局域网灵活数据速率（Controller Area Network Flexible Datarate, CAN FD）是控制器局域网（Controller Area Network, CAN）规范的新版本，可在保持 CAN 总线优势的同时支持更高的数据吞吐量。PIC18F 器件上的 CAN FD 模块具有许多新功能，因此该模块的配置与先前的 CAN 模块截然不同。本文档将概述设置 CAN FD 模块以发送和接收 CAN FD 帧的基础知识。

目录

简介.....	1
1. CAN FD 和 CAN 2.0 之间的差异.....	3
2. CAN 模式和模式更改.....	4
2.1. 配置模式.....	4
2.2. 正常模式.....	4
3. 波特率设置.....	5
3.1. 时钟设置和要求.....	5
3.2. 标称比特率 (NBR)	5
3.3. 数据比特率 (DBR)	6
3.4. 使用 Microchip 代码配置器 (MCC) 设置比特率.....	6
4. FIFO 和存储区使用.....	8
4.1. 存储区设置和 FIFOBA 寄存器.....	8
4.2. 发送事件 FIFO.....	8
4.3. 发送队列 (TXQ)	8
4.4. 可配置 FIFO 1-3.....	8
4.5. 使用 MCC 设置存储区.....	8
5. 发送.....	10
5.1. 写入发送 FIFO.....	10
5.2. 请求发送.....	10
5.3. MCC 发送 API.....	11
6. 接收.....	13
6.1. 屏蔽器和过滤器设置.....	13
6.2. 从接收 FIFO 读取.....	13
6.3. CAN 接收中断.....	14
6.4. MCC 接收设置/API.....	14
7. 结论.....	16
Microchip 网站.....	17
产品变更通知服务.....	17
客户支持.....	17
Microchip 器件代码保护功能.....	17
法律声明.....	17
商标.....	18
质量管理体系.....	18
全球销售及服务网点.....	19

1. CAN FD 和 CAN 2.0 之间的差异

CAN FD 和 CAN 2.0 的差异主要体现在两方面。最明显的差异是 CAN FD 有两个波特率，而不是一个。第一个是用于 ID、开销和 CRC 的标称波特率，其数据速率限制与 CAN 2.0 相同。第二个是专门用于数据字节的数据波特率，其数据速率可以比标称波特率高出很多。此外，CAN FD 帧的数据段最多可包含 64 个字节，多于 CAN 2.0 的 8 个字节。

两个协议之间还有一些较小的差异，主要源于以下两项主要变化：

- 增加了 BRS 和 FDF 位以指示 CAN FD 帧/速度，并且删除了 RTR（远程发送请求）位。
- CRC 从 15 位增加到 17 位或 21 位。

自 CAN 2.0 模块起，PIC18 器件中的 CAN FD 模块在连接新的 CAN FD 硬件和处理 CAN FD 协议的更改方面存在多项主要更改。

2. CAN 模式和模式更改

CAN FD 模块有多种工作模式：

- 配置模式
- 正常（CAN FD）模式
- 正常（CAN 2.0）模式
- 禁止模式
- 仅监听模式
- 受限工作模式
- 内部环回模式
- 外部环回模式

其中大多数模式主要用于开发、错误或调试场景。用于基本配置和发送/接收的主要模式为配置模式和正常模式。

2.1 配置模式

通过将 C1CONT 寄存器的 REQOP[2:0]位设置为 0b100，可进入配置模式。当 C1CONU 寄存器的 OPMOD[2:0]位设置为 0b100 时，配置模式激活。

只能在配置模式下对下列寄存器和位域进行编程：

- C1CON: WAKFIL、CLKSEL、PXEDIS、ISOCRECEN、TXQEN、STEF、SERRLOM、ESIGM 和 RTXAT
- C1NBTCFG、C1DBTCFG 和 C1TDC
- C1TXQCON: PLSIZE[2:0]和 FSIZE[4:0]
- C1FIFOCON: TXEN、RXTSEN、PLSIZE[2:0]和 FSIZE[4:0]
- C1TEFCON: TEFTSEN 和 FSIZE[4:0]
- CxFIFOBA

此外，所有 FIFO 和 TXQ 在配置模式下将复位。

2.2 正常模式

有两种正常模式：

- 通过将 C1CONT 寄存器的 REQOP[2:0]位设置为 0b000，可进入 CAN FD 正常模式
- 通过将 C1CONT 寄存器的 REQOP[2:0]位设置为 0b110，可进入 CAN 2.0 正常模式

CAN FD 必须退出配置模式并进入正常模式，才能执行任何发送或接收操作。CAN FD 正常模式可发送 CAN FD 或 CAN 2.0 帧，具体取决于 CAN 数据包的 FDF 位设置。相比之下，CAN 2.0 正常模式只能发送 CAN 2.0 帧，并且只能用于在专用 CAN 2.0 总线上使用器件的情况。

3. 波特率设置

设置 CAN FD 模块的第一步是设置通信波特率。CAN FD 有两个独立的波特率，因此使用两组独立的寄存器进行配置：

- 标称比特率 (NBR)，使用 C1NBTCFGT/U/H/L 寄存器
- 数据比特率 (DBR)，使用 C1DBTCFGT/U/H/L 寄存器

3.1 时钟设置和要求

CAN 时钟可从两个时钟源中选择：系统时钟或外部时钟。在很多情况下，两个时钟是相同的，只有一处明显的例外。外部时钟设置会旁路 4x PLL，允许 CAN 直接使用外部时钟，而器件由 4x PLL 电路的输出提供时钟。

建议仅通过特定的输入频率（尤其是 10 MHz、20 MHz 或 40 MHz）为 CAN FD 模块提供时钟。这种时钟限制是 CAN FD 硬件的固有限制，如果不遵守限制，可能会导致不可预测的行为和/或不良行为。也可以使用其他时钟频率进行调试和开发，但不建议将其用于实际应用。应用使用的外部晶振/时钟应采用上述推荐的频率之一。

3.2 标称比特率 (NBR)

标称比特率 (Nominal Bit Rate, NBR) 是数据前的仲裁段和数据后的 CRC 段的比特率。与 CAN 2.0 一样，该值受到物理 CAN 总线传播延时的限制，因此其数据速率的上限值较低。

3.2.1 标称比特率基础知识

标称比特率的设置需分步进行，第一步是选择标称时间份额 (Nominal Time Quanta, NTQ)。每个 CAN 位被分解为特定数量的时间份额 (Time Quanta, TQ)，然后进一步细分为更小的段，这些段也被类似地被分解为时间份额。利用以下公式，通过 C1NBTCFGT 寄存器的 BRP 位可确定标称时间份额：

公式 3-1. 标称时间份额

$$NTQ = NBRP \times T_{SYSCLK} = \frac{NBRP}{F_{SYSCLK}}$$

其中 NTQ 是标称时间份额，NBRP 是 C1NBTCFGT 寄存器 BRP 位的值， F_{SYSCLK} 是所选的 CAN 时钟。

使用得到的标称时间份额以及预期比特率，通过以下公式，即可确定每个位时间使用的时间份额数：

公式 3-2. 每位时间份额

$$\text{每位时间份额数} = \frac{1}{NTQ \times \text{比特率}}$$

在此处，每个位时间被分解为四个主要的段：

- 同步段
- 传播段
- 相位段 1
- 相位段 2

同步段是位时间中预计会出现边沿的部分。同步段是惟一个不可配置的段，始终硬编码为 1 个时间份额。传播段用于补偿物理 CAN 总线上的传播延时。该段的持续时间必须超过总线上信号的最长传播延时。相位段 1 和相位段 2 用于确定位的采样时间：采样点出现在从相位段 1 转换为相位段 2 的位置。两个相位段也可以延长或缩短，以调整位边沿的相移。

各个段通过 C1NBTCFGU 和 C1NBTCFGH 寄存器中的位进行配置。传播段和相位段 1 组合构成 TSEG1，由 C1NBTCFGU 寄存器的 TSEG1 位控制，相位段 2 由 C1NBTCFGH 寄存器的 TSEG2 位控制。如上文所述，同步段固定为 1 个时间份额，因此将这些配置与前面的公式结合，可以得到以下公式：

公式 3-3. 计算 TSEG1、TSEG2 和 BRP

$$\frac{F_{SYSCLK}}{(BRP + 1) \times \text{比特率}} = TSEG1 + TSEG2 + 1$$

无论 TSEG1、TSEG2 和 BRP 配置为何值，只要公式成立，配置即有效。例如，当输入时钟为 40 MHz，期望比特率为 500 kHz 时，公式可以简化为：

公式 3-4. TSEG1、TSEG2 和 BRP 示例

$$80 = (BRP + 1)(TSEG1 + TSEG2 + 1)$$

在确定 BRP、TSEG1 和 TSEG2 时有几个因素需要考虑。

3.2.2 BRP 考虑因素

对于 BRP，考虑到 TSEG1 和 TSEG2 的限制，通常建议选择使公式成立的最小值。这样可以更精细地控制采样点（时间份额更小）。

3.2.3 TSEG1 和 TSEG2 考虑因素

TSEG1 和 TSEG2 的两个考虑因素为：

1. 传播时间
2. 采样点

传播时间是 TSEG1 的一部分，因此 TSEG1 至少需要大于总线的最长传播延时，该延时由以下公式给出。

公式 3-5. 最长传播延时

$$T_{PROP} = 2 \times (t_{TXD-RXD} + T_{BUS})$$

其中 $t_{TXD-RXD}$ 为收发器的传播延时， T_{BUS} 为 CAN 总线的延时。

采样点是以百分比表示的位的采样位置，位于 TSEG1 和 TSEG2 之间。

3.2.4 同步跳转宽度 (SJW)

比特率设置的最后一步为同步跳转宽度 (Synchronization Jump Width, SJW)。当输入位的起始边沿不在同步段内时，此操作用于重新同步。在这种情况下，可以调整 PHSEG1 或 PHSEG2，使位时间与实际信号再次同步。SJW 是调整 PHSEG1 或 PHSEG2 以进行重新同步时所允许的最大值。该值应设置为可能的最大值（同时不超出 PHSEG1 或 PHSEG2 的范围），因为这会增大 CAN 节点的振荡器容差。

3.3 数据比特率 (DBR)

数据比特率 (Data Bit Rate, DBR) 是 CAN FD 帧期间发送的数据字节的比特率。由于该帧段可保证总线上只发送一个节点，因此其要求并非特别严苛，允许使用较高的数据速率。

数据比特率的设置方式与标称比特率相同，使用的是 C1DBTCFGT、C1DBTCFGU、C1DBTCFGH 和 C1DBTCFGL 寄存器，而非 C1NBTCFG 寄存器。所有公式和位的设置相似。主要差异在于，对于数据比特率，TSEG1、TSEG2 和 SJW 的最大值要小得多，因为考虑到 CAN 帧数据段中使用的比特率更高，所以每比特时间的时间份额自然会更低。

为数据比特率设置 BRP、TSEG1、TSEG2 和 SJW 时的考虑因素与标称比特率类似，但有以下三项主要差异。

1. 强烈建议将数据比特率预分频比 (Data Bit Rate Prescaler, DBRP) 设置为与标称比特率预分频比 (Nominal Bit Rate Prescaler, NBRP) 相同的值，这样可以防止比特率切换期间出现错误。
2. 如前文所述，TSEG1、TSEG2 和 SJW 的最大值更小，并且当 BRP 相同时，由于比特率更快，这些数据速率段中的时间份额通常更低。
3. TSEG1 值不再需要考虑传播延时，因为传播仅需要在仲裁期间关注，并且数据比特率仅限于只发送一个节点的情况。

3.4 使用 Microchip 代码配置器 (MCC) 设置比特率

MCC 配有简单的设置工具，允许用户输入时钟频率、所需的标称和数据比特率、时间份额数和采样点百分比，并会对所有相关寄存器进行写操作。下图给出了一个设置示例，生成的 MCC 代码通过 40 MHz 源时钟产生 500 Kb 的标称比特率和 2 Mb 的数据比特率。

例 3-1. MCC 比特率设置代码示例

```
static void CAN1_BitRateConfiguration(void)
{
    // SJW 14;
    C1NBTCFGL = 0x0E;

    // TSEG2 14;
    C1NBTCFGH = 0x0E;

    // TSEG1 63;
    C1NBTCFGU = 0x3F;

    // BRP 0;
    C1NBTCFGT = 0x00;

    // SJW 4;
    C1DBTCFGL = 0x04;

    // TSEG2 4;
    C1DBTCFGH = 0x04;

    // TSEG1 13;
    C1DBTCFGU = 0x0D;

    // BRP 0;
    C1DBTCFGT = 0x00;

    // TDCO 14;
    C1TDCH = 0x0E;

    // TDCMOD Auto;
    C1TDCU = 0x02;
}
```

4. FIFO 和存储区使用

CAN FD 模块中包含数据 FIFO 的实现，用于存储接收和发送的报文。所有这些 FIFO 都包含在一段用于 CAN FIFO 的存储区中，位于通用 RAM 的末端。

4.1 存储区设置和 FIFOBA 寄存器

寄存器的 C1FIFOBA 设置决定了 FIFO RAM 在存储区中的起始位置。在大多数情况下，建议在 8 位 CAN FD 器件上将 FIFOBA 设置在 CAN FIFO 指定存储区的开头，这样可以为 CAN FIFO 提供最大的空间，但在某些情况下，这段存储区可能需要用于其他用途，此时可以使用 C1FIFOBA 将 CAN FIFO 的开头重定位到相应存储段。

在 PIC18-Q84 器件上，CAN RAM 段的 RAM 大小被限制为 2 KB，因此在确定应用程序中使用的 FIFO 的数量和大小时，请记住这一点。

FIFO 段始于 C1FIFOBA 定义的位置，存储区按以下顺序分配：

1. TEF（发送事件 FIFO）
2. TXQ（发送队列）
3. FIFO 1
4. FIFO 2
5. FIFO 3

未使能或未使用的 FIFO 将被跳过，不占用任何 CAN RAM 段。

4.2 发送事件 FIFO

发送事件 FIFO（Transmit Event FIFO，TEF）用于记录发送报文，并附带序列号和可选时间戳。TEF 不存储发送报文的有效负载。最基本的发送事件 FIFO 配置包括：

- 通过将 C1CONU 寄存器的 STEF 位置 1 来使能 TEF。
- 通过更改 C1TEFCONT 寄存器的 FSIZE 位来配置 TEF 中存储的报文对象的数量。
- 通过 C1TEFCONL 寄存器的 TEFTSEN 位使能或禁止 TEF 中报文对象的时间戳。

4.3 发送队列（TXQ）

发送队列（Transmit Queue，TXQ）属于仅发送 FIFO。通过将 C1CONU 的 TXQEN 位置 1 可使能 TXQ，C1TXQCONT 的 FSIZE 位用于配置可存储在 TXQ 中的报文对象的数量。C1TXQCONT 的 PLSIZE 位用于配置报文对象的有效负载大小（同一 FIFO 中所有对象的有效负载大小相同）。

4.4 可配置 FIFO 1-3

其他三个 FIFO（1-3）可配置为发送或接收 FIFO。通过将 C1FIFOCONxL 寄存器中的 TXEN 位置 1，可以将每个 FIFO 配置为发送 FIFO，其中 x 是被控的 FIFO 编号。通过将 TXEN 位清零，可将 FIFO 配置为接收 FIFO。与发送队列一样，C1FIFOCONxT 寄存器的 FSIZE 位用于配置 FIFO 中可存储的报文对象数量，C1FIFOCONxT 寄存器的 PLSIZE 位用于配置报文对象的有效负载大小。当 FIFO 处于接收模式时，可将 C1FIFOCONxL 寄存器的 RXTSEN 位置 1 以使用接收报文的时间戳。

4.5 使用 MCC 设置存储区

MCC Easy Setup 可用于设置 FIFO、使能 TXQ 和全部三个可重新编程的 FIFO，以及配置以下各项：

- RX FIFO 或 TX FIFO（TXQ 除外）
- FIFO 深度（报文对象数量）

- 有效负载（每个报文对象中的数据字节数）
- 自定义名称（适用于 TX FIFO）
- 发送优先级（适用于 TX FIFO）

FIFO 设置还会汇总 FIFO 使用的数据量，显示 CAN 存储区中是否有足够的空间供 FIFO 使用。下面给出了 MCC 生成的示例代码，其中 FIFO 1 设置为 RX FIFO、FIFO 2 设置为 TX FIFO，两个 FIFO 以及 TXQ 的深度均为 6、有效负载大小均为 32 字节。

例 4-1. 用于设置 FIFO 的 MCC 代码

```
static void CAN1_RX_FIFO_Configuration(void)
{
    // TXEN disabled; RTREN disabled; RXTSEN disabled; TXATIE disabled; RXOVIE
    disabled; TFERFFIE disabled; TFHRFHIE disabled; TFNRFNIE disabled;
    C1FIFOCON1L = 0x00;

    // FRESET enabled; TXREQ disabled; UINC disabled;
    C1FIFOCON1H = 0x04;

    // TXAT Unlimited number of retransmission attempts; TXPRI 1;
    C1FIFOCON1U = 0x60;

    // PLSIZE 32; FSIZE 6;
    C1FIFOCON1T = 0xA5;
}

static void CAN1_TX_FIFO_Configuration(void)
{
    // TXATIE disabled; TXQEIE disabled; TXQNIE disabled;
    C1TXQCONL = 0x00;

    // FRESET enabled; UINC disabled;
    C1TXQCONH = 0x04;

    // TXAT 3; TXPRI 1;
    C1TXQCONU = 0x60;

    // PLSIZE 32; FSIZE 6;
    C1TXQCONT = 0xA5;

    // TXEN enabled; RTREN disabled; RXTSEN disabled; TXATIE disabled; RXOVIE
    disabled; TFERFFIE disabled; TFHRFHIE disabled; TFNRFNIE disabled;
    C1FIFOCON2L = 0x80;

    // FRESET enabled; TXREQ disabled; UINC disabled;
    C1FIFOCON2H = 0x04;

    // TXAT Unlimited number of retransmission attempts; TXPRI 1;
    C1FIFOCON2U = 0x60;

    // PLSIZE 32; FSIZE 6;
    C1FIFOCON2T = 0xA5;
}
```

5. 发送

使用 CAN FD 模块发送报文的操作分为两步：写入发送 FIFO 和从该 FIFO 请求发送。发送操作可以由 TXQ 或任何可重新配置的 FIFO 执行（如果配置为发送模式）。

5.1 写入发送 FIFO

对发送 FIFO 进行写操作之前，必须确认 FIFO 未滿。可以通过检查 C1FIFOSTAxL 或 C1TXQSTAL 寄存器的 TFNRFNIF 位进行确认。如果标志置 1，说明 FIFO 未滿。每个发送 FIFO 都包含一组用户地址寄存器（C1FIFOxUA 或 C1TXQUA），用于确定 FIFO 头部的 RAM 位置。对发送 FIFO 进行写操作是为了将发送报文对象写入该地址。发送报文对象的存储方式如下表所示（示例有效负载为 n 个字节，总发送对象大小为 m 个字节）。

字节	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	SID[7:0]							
1	EID[4:0]				SID[10:8]			
2	EID[12:5]							
3	-	-	SID11	EID[17:13]				
4	FDF	BRS	RTR	IDE	DLC[3:0]			
5	SEQ[6:0]							ESI
6	SEQ[14:7]							
7	SEQ[22:15]							
8	发送数据字节 0							
9	发送数据字节 1							
10	发送数据字节 2							
11	发送数据字节 3							
12	发送数据字节 4							
13	发送数据字节 5							
14	发送数据字节 6							
15	发送数据字节 7							
.....								
m-3	发送数据字节 n-3							
m-2	发送数据字节 n-2							
m-1	发送数据字节 n-1							
m	发送数据字节 n							

报文加载到发送 FIFO 中后，将 C1FIFOCONxH 或 C1TXQCONH 寄存器的 UINC 位置 1 会递增相应 FIFO 的头部。

5.2 请求发送

将报文加载到发送 FIFO 中后，通过对 C1FIFOCONxH 或 C1TXQCONH 寄存器的 TXREQ 位进行写操作，即可从 FIFO 发送报文。将 TXREQ 位置 1 会发送各个 FIFO 中的所有报文，所有报文发送完成后，该位清零。可以同时请求

多个 FIFO 和 TXQ 发送操作，这些 FIFO/TXQ 会按照优先级从高到低发送。最高优先级由 C1FIFOCONxU 或 C1TXQCONU 寄存器的 TXPRI[4:0]位确定。FIFO 中配置为发送的报文将以先入先出的方式发送，而 TXQ 中的报文将根据其报文 ID 发送（先发送报文 ID 较小的报文）。

在发送报文的同时，可以继续将报文加载到 FIFO 或 TXQ 中，但在进行该操作时应同时将 TXREQ 和 UINC 位置 1，以确保发送所有附加的报文。

除了位于各个 FIFO 配置寄存器中的 TXREQ 位外，C1TXREQL 寄存器还包含所有 TXREQ 位的镜像位，与 TXREQ 位一样，这些位也可以进行写操作或读操作。这样即可从一个位置轻松监控报文的发送。

5.3 MCC 发送 API

除了设置发送 FIFO 外，MCC 还支持创建 CAN FD 发送 API 以及数据结构，以便更轻松地发送报文对象。其中最重要的是 CAN_MSG_OBJ 结构和 CAN_MSG_FIELD 联合。

这两种数据类型包含了 CAN FD 报文对象中的所有数据，其中 CAN_MSG_FIELD 包含 ID 类型、帧类型、DLC、格式类型和比特率切换，CAN_MSG_OBJ 包含 CAN ID、CAN_MSG_FIELD 和指向数据字节的指针。

CAN_MSG_FIELD 的每个部分都有一个与之关联的枚举，以方便用户使用：

- BRS 模式可设置为 CAN_NON_BRS_MODE 模式或 CAN_BRS_MODE，具体取决于报文的比特率是否变化。
- ID 类型可以设置为 CAN_FRAME_STD 或 CAN_FRAME_EXT，具体取决于报文具有标准 ID 还是扩展 ID。
- 帧模式可设置为 CAN_FRAME_DATA 或 CAN_FRAME_RTR（请记住，此设置仅适用于 CAN 2.0 报文，因为 CAN FD 不支持远程帧）。
- 格式模式可设置为 CAN_2_0_FORMAT 或 CAN_FD_FORMAT，具体取决于报文是 CAN 2.0 报文还是 CAN FD 报文（请记住，CAN 2.0 报文不支持 BRS，CAN FD 报文不支持 RTR）。

使用这些数据结构时需要声明一个 CAN_MSG_OBJ 对象，然后为每个位域分配适当的值。下面的示例为已使能比特率切换并且具有 32 个数据字节的标准 ID CAN FD 报文的设置。

例 5-1. 设置 CAN_MSG_OBJ

```
CAN_MSG_OBJ Periodic_Transmit;
    uint8_t
    Transmit_Data[32]={0x00,0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0A,0x0B,
    0x0C,0x0D,0x0E,0x0F,0x10,0x11,0x12,0x13,0x14,0x15,0x16,0x17,
    0x18,0x19,0x1A,0x1B,0x1C,0x1D,0x1E,0x1F};
    Periodic_Transmit.msgId=0x001;
    Periodic_Transmit.field.formatType=CAN_FD_FORMAT;
    Periodic_Transmit.field.idType = CAN_FRAME_STD;
    Periodic_Transmit.field.dlc = DLC 32;
    Periodic_Transmit.field.brs = CAN_BRS_MODE;
    Periodic_Transmit.field.frameType = CAN_FRAME_DATA;
    Periodic_Transmit.data = Transmit_Data;
```



CAUTION CAN_MSG_OBJ 的任何部分初始化失败都可能产生不良后果，因为这些字段在 RAM 中可能有未关联的剩余数据。

创建 CAN_MSG_OBJ 并完成初始化后，MCC 就会设置一些函数以协助发送报文对象。

CAN1_TransmitFIFOStatusGet(const CAN1_TX_FIFO_CHANNELS fifoChannel) 会返回 FIFO 是否已满的消息，而 CAN1_Transmit(const CAN1_TX_FIFO_CHANNELS fifoChannel, CAN_MSG_OBJ *txCanMsg) 函数需要使用 CAN 发送 FIFO 的参数并引用 CAN_MSG_OBJ。接着上一个示例，下列代码可检查 TXQ 是否已满，如果未满足，则发送先前初始化的报文对象。

```
if(CAN_TX_FIFO_AVAILABLE == (CAN1_TransmitFIFOStatusGet(TXQ) & CAN_TX_FIFO_AVAILABLE))
{
    CAN1_Transmit(TXQ, &Periodic_Transmit);
}
```

CAN1Transmit 函数将自动处理 FIFO 写操作、FIFO 递增和发送请求设置。

6. 接收

在处理 FIFO 时，CAN FD 模块的报文接收过程与发送过程类似，但有一个额外的考虑因素：屏蔽器和过滤器。CAN 协议基于报文，任何应用程序都只会响应特定的 CAN ID。屏蔽器和过滤器会使 CAN FD 模块忽略不在该特定列表中的所有报文。

6.1 屏蔽器和过滤器设置

过滤器使用 C1FLTCONx 和 C1FLTOBJx 寄存器配置。每个过滤器都有一组 C1FLTOBJ 寄存器，其中包含 11/12 个标准标识符位、18 个扩展标识符位和一个 EXIDE 位，其中 EXIDE 位用于确定过滤器只接受扩展 ID 还是标准 ID 报文。通过将 EXIDE 位置 1 或清零，然后将 SID/EID 位设置为过滤器应接受的报文 ID，即可将该过滤器配置为接受该报文 ID。

然后使用 C1FLTCONx 寄存器将过滤器与 FIFO 相关联。C1FLTCON0L 配置过滤器 0，C1FLTCON0H 配置过滤器 1，C1FLTCON0U 配置过滤器 2，依此类推，一直到 C1FLTCON2T 配置过滤器 11。FnBP 位用于确定每个过滤器所关联的 FIFO，FLTENn 位用于使能过滤器。

注：在相关的 C1FLTOBJn 和 FnBP 寄存器完全配置之前，不得将 FLTENn 置 1，因为一旦使能过滤器，这些寄存器或位将会锁定，对这些寄存器或位的任何写操作都不会反映在实际寄存器中。

过滤器允许接受单个报文 ID，而屏蔽器允许各个 FIFO 将 ID 的某些位视为“无关”，从而只使用一个或两个过滤器接受多个报文。每个过滤器都有一个由 C1MASKn 配置的相关屏蔽器，该屏蔽器的位与过滤器寄存器的位相同（标准 ID 和扩展 ID 以及 MIDE 位，MIDE 位决定了过滤器是否关注一个报文是标准 ID 还是扩展 ID，或者同时接受两种 ID）。

6.2 从接收 FIFO 读取

如前文所述，在接收 FIFO 中处理数据与在发送 FIFO 中处理数据非常相似。在大多数用例中，通过中断触发可以明显看出接收 FIFO 非空（如 [CAN 接收中断](#) 所示）。但如果情况并非如此，则在尝试从 FIFO 读取数据之前，需要检查 TFNRFNIF 位以确保 FIFO 非空。完成检查后，C1FIFOxUA 寄存器将使用以下格式给出 RAM 中包含 CAN 报文对象的位置：

字节	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	SID[7:0]							
1	EID[4:0]					SID[10:8]		
2	EID[12:5]							
3	-	-	SID11	EID[17:13]				
4	FILHIT[4:0]					-	-	ESI
5	FDF	BRS	RTR	IDE	DLC[3:0]			
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-
8	接收数据字节 0							
9	接收数据字节 1							
10	接收数据字节 2							
11	接收数据字节 3							
12	接收数据字节 4							
13	接收数据字节 5							
14	接收数据字节 6							

15	接收数据字节 7
.....	
m-3	接收数据字节 n-3
m-2	接收数据字节 n-2
m-1	接收数据字节 n-1
m	接收数据字节 n

从接收 FIFO 读取报文对象后，需要将 C1FIFOxUA 位置 1，让 C1FIFOxUA 寄存器递增到下一个报文位置，从而使 FIFO 的下一个报文对象成为下一个接收的报文。

6.3 CAN 接收中断

CAN 接收的常见用例是中断驱动，CAN FD 模块有一个专用于此类用例的中断。顶层 CANRX 中断标志（CANRXIF）是一个只读中断标志，可由三个不同的接收事件触发：

1. 接收 FIFO 不为空
2. 接收 FIFO 半满
3. 接收 FIFO 已满

此外，每个接收 FIFO 可以单独允许这些中断，并且可以触发顶层中断。

简而言之，要设置接收器中断，必须遵循以下程序：

1. 为所有应产生中断的接收 FIFO 选择并允许所需的 FIFO 中断触发（非空、半满和满）。这些触发分别由相关 FIFO 编号的 C1FIFOCONxL TFERFFIE（满）、TFHRFHIE（半满）和 TFNRFNIE（非空）位控制。
2. 在 C1INTU 寄存器中使能接收对象中断允许（RXIE）位。
3. 使能 PIE 寄存器中的 CAN 接收中断允许位，其位置可能因器件而异。

使能的接收 FIFO 条件将在此处产生中断，用户代码必须检查触发中断的 FIFO/条件。可以通过 C1RXIFL 寄存器来辅助检查，该寄存器针对每个 FIFO 都有单独的位，表明该 FIFO 有待处理的中断，C1FIFOSTAxL 寄存器中也有单独的 TFERFFIF、TFHRFHIF 和 TFNRFNIF 位与该 FIFO 相对应。

6.4 MCC 接收设置/API

MCC Easy Setup 支持配置屏蔽器和过滤器以及设置过滤对象。通过选择与过滤器相关联的接收 FIFO 和过滤器应接受的报文 ID，可以单独使能和配置每个过滤器。过滤器和屏蔽器配置代码将自动生成并添加到项目代码中。

接收操作使用的数据结构与先前的 [MCC 发送 API](#) 部分提到的数据结构相同，最主要的是 CAN_MSG_OBJ 数据类型。在这种情况下，需要声明一个报文对象，然后通过轮询或中断驱动操作来使用 CAN1_Receive(CAN_MSG_OBJ *rxCanMsg) 函数。该函数会将接收的报文中的所有数据放入声明的报文对象中，并处理 FIFO 递增。然后，可以通过声明的 CAN_MSG_OBJ 处理所有必需的数据。例如，使用以下代码，可以接收报文并将第一个数据字节输出到连接端口 B 和端口 D 的一系列 LED 上。

例 6-1. 通过 MCC 接收 CAN FD 报文

```

CAN_MSG_OBJ LED_Message;
uint8_t LED_Holder;
//polling or interrupt code here
CAN1_Receive(&LED_Message)
LED_Holder=LED_Message.data[0]
LATDbits.LATD0 = LED_Holder>>7;
LATDbits.LATD1 = (LED_Holder-128)>>6;
LATDbits.LATD2 = (LED_Holder-192)>>5;
LATDbits.LATD3 = (LED_Holder-224)>>4;
LATBbits.LATB0 = (LED_Holder-240)>>3;
LATBbits.LATB1 = (LED_Holder-248)>>2;

```

```
LATBbits.LATB2 = (LED_Holder-252)>>1;  
LATBbits.LATB3 = LED_Holder-254;
```

7. 结论

无论在功能上还是在模块的使用方式上，CAN FD 模块都代表了 PIC18F 单片机的一种新型 CAN 接口。本文档概述了在使用 CAN FD 模块发送或接收数据之前需要配置的主要方面。其中重点介绍了比特率配置、FIFO 和存储区使用，以及发送/接收报文对象设置和报文交换。此外，本文档还概述了如何使用 MPLAB® 代码配置器（MPLAB Code Configurator, MCC）简化每个配置步骤中的代码创建过程。

Microchip 网站

Microchip 网站 (www.microchip.com/) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。我们的网站提供以下内容:

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **一般技术支持**——常见问题解答 (FAQ)、技术支持请求、在线讨论组以及 Microchip 设计伙伴计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

产品变更通知服务

Microchip 的产品变更通知服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时, 收到电子邮件通知。

欲注册, 请访问 www.microchip.com/pcn, 然后按照注册说明进行操作。

客户支持

Microchip 产品的用户可通过以下渠道获得帮助:

- 代理商或代表
- 当地销售办事处
- 应用工程师 (ESE)
- 技术支持

客户应联系其代理商、代表或 ESE 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 www.microchip.com/support 获得网上技术支持。

Microchip 器件代码保护功能

请注意以下有关 Microchip 器件代码保护功能的要点:

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信: 在正常使用的情况下, Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前, 仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知, 所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿意与关心代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案 (Digital Millennium Copyright Act)》。如果这种行为导致他人在未经授权的情况下, 能访问您的软件或其他受版权保护的成果, 您有权依据该法案提起诉讼, 从而制止这种行为。

法律声明

提供本文档的中文版本仅为了便于理解。请勿忽视文档中包含的英文部分, 因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物及其提供的信息仅适用于 Microchip 产品, 包括设计、测试以及将 Microchip 产品集成到您的应用中。以其他方式任何方式使用这些信息都将被视为违反条款。本出版物中的器件应用信息仅为您提供便利, 将来可能会发生更新。如

需额外的支持，请联系当地的 Microchip 销售办事处，或访问 <https://www.microchip.com/en-us/support/design-help/client-supportservices>。

Microchip “按原样”提供这些信息。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对非侵权性、适销性和特定用途的适用性的暗示担保，或针对其使用情况、质量或性能的担保。

在任何情况下，对于因这些信息或使用这些信息而产生的任何间接的、特殊的、惩罚性的、偶然的或间接的损失、损害或任何类型的开销，Microchip 概不承担任何责任，即使 Microchip 已被告知可能发生损害或损害可以预见。在法律允许的最大范围内，对于因这些信息或使用这些信息而产生的所有索赔，Microchip 在任何情况下所承担的全部责任均不超出您为获得这些信息向 Microchip 直接支付的金额（如有）。如果将 Microchip 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切损害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任。除非另外声明，在 Microchip 知识产权保护下，不得暗或以其他方式转让任何许可证。

商标

Microchip 的名称和徽标组合、Microchip 徽标、Adaptec、AnyRate、AVR、AVR 徽标、AVR Freaks、BesTime、BitCloud、chipKIT、chipKIT 徽标、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi 徽标、MOST、MOST 徽标、MPLAB、OptoLyzer、PackeTime、PIC、picoPower、PICSTART、PIC32 徽标、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST 徽标、SuperFlash、Symmetricom、SyncServer、Tachyon、TempTrackr、TimeSource、tinyAVR、UNI/O、Vectron 和 XMEGA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的注册商标。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus 徽标、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath 和 ZL 均为 Microchip Technology Incorporated 在美国的注册商标。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoAutomotive、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet 徽标、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICKit、PICtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、ViewSpan、WiperLock、Wireless DNA 和 ZENA 均为 Microchip Technology Incorporated 在美国和其他国家或地区的商标。

SQTP 是 Microchip Technology Incorporated 在美国的服务标记。

Adaptec 徽标、Frequency on Demand、Silicon Storage Technology 和 Symmcom 为 Microchip Technology Inc. 在其他国家或地区的注册商标。

GestIC 是 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2022, Microchip Technology Incorporated, 美国印刷, 版权所有。

ISBN: 978-1-6683-0402-0

质量管理体系

有关 Microchip 的质量管理体系的信息，请访问 www.microchip.com/quality。

全球销售及服务中心

美洲	亚太地区	亚太地区	欧洲
公司总部 2355 West Chandler Blvd. Chandler, AZ 85224-6199 电话: 480-792-7200 传真: 480-792-7277 技术支持: www.microchip.com/support 网址: www.microchip.com	澳大利亚 - 悉尼 电话: 61-2-9868-6733 中国 - 北京 电话: 86-10-8569-7000 中国 - 成都 电话: 86-28-8665-5511 中国 - 重庆 电话: 86-23-8980-9588 中国 - 东莞 电话: 86-769-8702-9880 中国 - 广州 电话: 86-20-8755-8029 中国 - 杭州 电话: 86-571-8792-8115 中国 - 香港特别行政区 电话: 852-2943-5100 中国 - 南京 电话: 86-25-8473-2460 中国 - 青岛 电话: 86-532-8502-7355 中国 - 上海 电话: 86-21-3326-8000 中国 - 沈阳 电话: 86-24-2334-2829 中国 - 深圳 电话: 86-755-8864-2200 中国 - 苏州 电话: 86-186-6233-1526 中国 - 武汉 电话: 86-27-5980-5300 中国 - 西安 电话: 86-29-8833-7252 中国 - 厦门 电话: 86-592-2388138 中国 - 珠海 电话: 86-756-3210040	印度 - 班加罗尔 电话: 91-80-3090-4444 印度 - 新德里 电话: 91-11-4160-8631 印度 - 浦那 电话: 91-20-4121-0141 日本 - 大阪 电话: 81-6-6152-7160 日本 - 东京 电话: 81-3-6880-3770 韩国 - 大邱 电话: 82-53-744-4301 韩国 - 首尔 电话: 82-2-554-7200 马来西亚 - 吉隆坡 电话: 60-3-7651-7906 马来西亚 - 槟榔屿 电话: 60-4-227-8870 菲律宾 - 马尼拉 电话: 63-2-634-9065 新加坡 电话: 65-6334-8870 台湾地区 - 新竹 电话: 886-3-577-8366 台湾地区 - 高雄 电话: 886-7-213-7830 台湾地区 - 台北 电话: 886-2-2508-8600 泰国 - 曼谷 电话: 66-2-694-1351 越南 - 胡志明市 电话: 84-28-5448-2100	奥地利 - 韦尔斯 电话: 43-7242-2244-39 传真: 43-7242-2244-393 丹麦 - 哥本哈根 电话: 45-4485-5910 传真: 45-4485-2829 芬兰 - 埃斯波 电话: 358-9-4520-820 法国 - 巴黎 电话: 33-1-69-53-63-20 传真: 33-1-69-30-90-79 德国 - 加兴 电话: 49-8931-9700 德国 - 哈恩 电话: 49-2129-3766400 德国 - 海尔布隆 电话: 49-7131-72400 德国 - 卡尔斯鲁厄 电话: 49-721-625370 德国 - 慕尼黑 电话: 49-89-627-144-0 传真: 49-89-627-144-44 德国 - 罗森海姆 电话: 49-8031-354-560 以色列 - 若那那市 电话: 972-9-744-7705 意大利 - 米兰 电话: 39-0331-742611 传真: 39-0331-466781 意大利 - 帕多瓦 电话: 39-049-7625286 荷兰 - 德卢内市 电话: 31-416-690399 传真: 31-416-690340 挪威 - 特隆赫姆 电话: 47-72884388 波兰 - 华沙 电话: 48-22-3325737 罗马尼亚 - 布加勒斯特 电话: 40-21-407-87-50 西班牙 - 马德里 电话: 34-91-708-08-90 传真: 34-91-708-08-91 瑞典 - 哥德堡 电话: 46-31-704-60-40 瑞典 - 斯德哥尔摩 电话: 46-8-5090-4654 英国 - 沃金厄姆 电话: 44-118-921-5800 传真: 44-118-921-5820
亚特兰大 德卢斯, 佐治亚州 电话: 678-957-9614 传真: 678-957-1455 奥斯汀, 德克萨斯州 电话: 512-257-3370 波士顿 韦斯特伯鲁, 马萨诸塞州 电话: 774-760-0087 传真: 774-760-0088 芝加哥 艾塔斯卡, 伊利诺伊州 电话: 630-285-0071 传真: 630-285-0075 达拉斯 阿迪森, 德克萨斯州 电话: 972-818-7423 传真: 972-818-2924 底特律 诺维, 密歇根州 电话: 248-848-4000 休斯顿, 德克萨斯州 电话: 281-894-5983 印第安纳波利斯 诺布尔斯特维尔, 印第安纳州 电话: 317-773-8323 传真: 317-773-5453 电话: 317-536-2380 洛杉矶 米慎维荷, 加利福尼亚州 电话: 949-462-9523 传真: 949-462-9608 电话: 951-273-7800 罗利, 北卡罗来纳州 电话: 919-844-7510 纽约, 纽约州 电话: 631-435-6000 圣何塞, 加利福尼亚州 电话: 408-735-9110 电话: 408-436-4270 加拿大 - 多伦多 电话: 905-695-1980 传真: 905-695-2078			