

## 1 简介

LPC54018 具有两个支持 CAN-FD 的 CAN 控制器。LPC54018 SDK 提供了 `mcan_interrupt_transfer` 示例，以演示如何使用 CAN 传输数据。基于这个示例，本应用笔记描述了 CAN-FD 比特率开关和发送器延迟补偿功能的使用。启用这两个功能可以提高吞吐量，并消除由发射机延迟引起的误码。

## 2 CAN-FD

CAN-FD 在国际标准 ISO 11898-1:2015 中有定义。为了快速开始使用 CAN-FD，本节针对熟悉 CAN 的用户介绍了 CAN-FD 的一些关键特性。

### 2.1 CAN 和 CAN-FD 之间的区别

经典 CAN 和 CAN-FD 之间有两个主要区别。首先是 CAN-FD 可以使用比传统 CAN 高得多的比特率。经典 CAN 的比特率限制为 1 Mbps。CAN-FD 没有理论上的限制，但在实际应用中，它受到收发机的限制。第二个关键区别是每个 CAN 消息的数据量增加。经典 CAN 限制为 8 个字节。CAN-FD 限制增加了 8 倍，达到每条消息 64 字节。

随着每个 CAN 报文的数据量的增加，CAN-FD 帧需要更高的比特率来减少通信中的延迟时间，提高实时性。通过启用比特率切换功能，CAN-FD 帧可以达到更高的比特率。

另一方面，比特率越高，比特时间越短。为了使数据相位比特时间比发射机延迟更短，引入了延迟补偿。在没有发射器延迟补偿的情况下，CAN-FD 帧的数据相位的比特率受到发射器延迟的限制。

### 2.2 比特率开关

在 CAN-FD 帧中，控制相位，数据相位和 CRC 相位比帧的开头和结尾更高的比特率传输。

通过将 CCCR 寄存器中的 BRSE 位置 1，可以启用位速率切换功能。启用比特率切换时，我们还需要正确设置仲裁相位比特率（启用比特率切换之前）和数据相位比特率（启用比特率切换之后）。仲裁相位比特率由 NBTP 寄存器设置，数据相位比特率由 DBTP 寄存器设置。

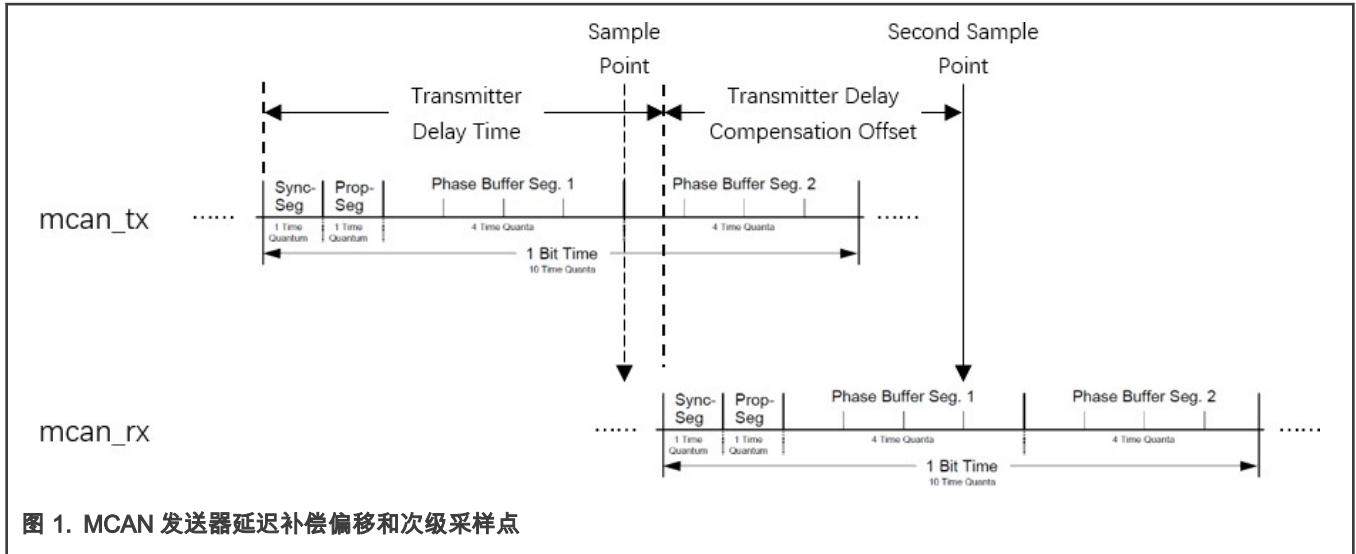
### 2.3 发送器延迟补偿

MCAN 的协议单元实现了一种时延补偿机制来补偿发射机的时延。发送器延迟补偿可实现数据位时间短于发送器延迟的配置。新 ISO11898-1 中对此进行了详细描述。通过将 DBTP 寄存器中的 TDC 位置 1 来启用它。TDCR 寄存器中的 TDCO 字段用于设置发送器延迟补偿偏移量。偏移值定义了从 `m_can_tx` 到 `m_can_rx` 的延迟与辅助采样点之间测得的距离。发射器延迟补偿偏移和次级采样点如图 1 所示。

## 目录

1	简介.....	1
2	CAN-FD.....	1
2.1	CAN 和 CAN-FD 之间的区别.....	1
2.2	比特率开关.....	1
2.3	发送器延迟补偿.....	1
3	示例.....	2
3.1	硬件环境.....	2
3.2	软件环境.....	3
3.3	使用 CAN-FD 的步骤概述.....	3
3.4	配置 CAN-FD 比特率开关.....	4
3.5	配置发送器延迟补偿.....	5
3.6	步骤和结果.....	5





### 3 示例

本节简要介绍如何使用 CAN-FD 传输数据以及启用比特率切换和发送器延迟补偿。它基于 LPC54018 SDK 的 mcan\_interrupt\_transfer 示例。

#### 3.1 硬件环境

- 主板
  - 两块 LPCXpresso54018 板 (OM40003)
  - 两个双 CAN-FD 屏蔽收发器 (OM13099)
- 其他
  - 两个 Micro USB 电缆
  - 一根 120 欧姆端接的 CAN 电缆
  - 个人电脑
- 主板设置

CAN 屏蔽线与评估板相连。每个屏蔽线上的 CAN0 接口使用端与 CAN 电缆连接。将微型 USB 电缆连接到 PC 和评估板上的 J8 跳线上，用来下载演示固件。也用于与 PC 上 UART 端子的 UART 通信。板的设置和 CAN 连接如图 2 所示。



图 2. 电路板设置和 CAN 连接

## 3.2 软件环境

- 工具链
  - MCUXpresso10.3.0 或更高版本
- 软件包
  - `lpcxpresso54018_mcan_interrupt_transfer_canfd.zip`
- UART 终端程序
  - PuTTY 或类似的程序

## 3.3 使用 CAN-FD 的步骤概述

本应用笔记提供了一个 `mcan_interrupt_transfer_canfd` 软件包。它演示了如何使用 CAN - FD 传输数据。在此示例中，需要执行以下步骤：

- 设置系统时钟
- 设置 MCAN 时钟
  - 为 MCAN 模块划分系统时钟
- 初始化 MCAN

- 启用 MCAN 时钟
- 重置 MCAN 模块
- 配置 MCAN 控制寄存器，启用 CAN-FD 和比特率切换
- 设置仲裁相位比特率 and 数据相位比特率
- 启用发射机延迟补偿
- 设置消息 RAM
- 设置消息 ID 过滤器配置和元件
- 设置 Rx FIFO 和 Tx 缓冲区的配置
- 进入 MCAN 正常模式
- 传输数据
  - 配置 TX 帧数据并发送
  - 接收数据

在 LPC54018 SDK 的 `mcan_interrupt_transfer` 示例中，默认情况下它使用总线上的经典 CAN 节点。数据相位比特率设置需要与仲裁相位比特率设置相同。它们还应符合经典的 CAN 总线协议。

但是在本文档的示例中，它使用总线上的 CAN-FD 节点。CAN-FD 仲裁相位比特率设置为 1 Mbps，数据相位比特率设置为 5 Mbps。它必须启用比特率开关和发送器延迟补偿功能。

下面各节介绍了使用 CAN-FD 的一些关键步骤。

### 3.4 配置 CAN-FD 比特率开关

启用 CAN-FD 比特率开关可以提高吞吐量。在该示例中，它调用 `MCAN_SetBaudRate()` 函数设置仲裁相位比特率，并调用 `MCAN_SetBaudRateFD()` 函数设置数据相位比特率。MCAN 时钟设置为 60 MHz。

#### 3.4.1 将仲裁阶段的比特率设置为 1 Mbps

根据 CAN 规范，标称比特率是指理想发射机在没有重新同步的情况下每秒传输的比特数。标称比特率和标称比特时间之间的关系为：标称比特时间 = 1 / 标称比特率。因此，如果仲裁阶段比特率设置为 1 Mbps，则仲裁阶段比特时间为 1  $\mu$ s。

时间量 ( $t_q$ ) 是从 MCAN 时钟周期得出的固定时间单位。存在一个可编程的预分频器，其整数值的范围至少为 1 到 32。从 MCAN 时钟周期开始， $t_q$  的长度为

$t_q = m * \text{MCAN 时钟周期} = m / \text{MCAN 时钟}$  其中  $m$  为预分频器的值。

在 `MCAN_SetBaudRate()` 函数中，我们需要定义一个类型为 `mcan_timing_config_t` 的变量，以设置仲裁阶段位时间。结构 `mcan_timing_config_t` 定义如图 3。

```
typedef struct _mcan_timing_config
{
    uint16_t preDivider; /*!< Clock Pre-scaler Division Factor. */
    uint8_t rJumpwidth; /*!< Re-sync Jump Width. */
    uint8_t seg1; /*!< Data Time Segment 1. */
    uint8_t seg2; /*!< Data Time Segment 2. */
} mcan_timing_config_t;
```

图 3. MCAN 协议时序特性配置结构

预分频器  $m$  相当于  $(preDivider + 1)$ 。具有结构元素 `preDivider` 的  $t_q$  的长度可以为  $t_q = (preDivider + 1) / 60 \text{ MHz} = (preDivider + 1) / 60 \text{ } (\mu\text{s})$ 。

可以将仲裁阶段位时间中的  $t_q$  总数设置为 4 至 385 个时间量。仲裁阶段位时间的长度 =  $MCAN\_TIME\_QUANTA\_NUM\_ARBIT * t_q$ 。

在示例中，我们将宏 `MCAN_TIME_QUANTA_NUM_ARBIT` 定义为 20。仲裁阶段的位时间为  $1 \mu s$ ， $t_q$  为  $1/20 \mu s$ ，`preDivider` 为 2。

宏：

```
MCAN_TIME_QUANTA_NUM_ARBIT = 1 + (seg1 + 1) + (seg2 + 1)
```

其中结构元素 `seg1` 和 `seg2` 分别代表相位缓冲段 1 和 2 减一。在示例中，我们将元素 `seg1` 设置为值 13，并将元素 `seg2` 设置为 4。调用更新的 `MCAN_SetBaudRate()` 函数，我们完成将仲裁阶段的比特率设置为 1 Mbps。

### 3.4.2 将数据相位比特率设置为 5 Mbps

设置数据相位比特率类似于设置仲裁相位比特率。数据相位比特率在 `MCAN_SetBaudRateFD()` 函数中设置。

其中一个区别是，数据相位位时间中的  $t_q$  总数可以编程在 4 到 49 个时间粒度的范围内。

在该示例中，我们将宏 `MCAN_TIME_QUANTA_NUM_DATA` 定义为 12，作为数据相位位时间内  $t_q$  的总数。元素 `seg1` 为 7，元素 `seg2` 为 2。元素 `preDivider` 为 4。调用更新的 `MCAN_SetBaudRateFD()` 函数，我们将数据相位比特率设置为 5 Mbps，这是板载收发器可以支持的最高比特率。

### 3.4.3 启用比特率开关

将 `MCAN_CCCR` 寄存器的 `BRSE` 位设置为 1，可启用 CAN-FD 的位速率开关。在本应用笔记的示例中，CAN-FD 帧的控制相位，数据相位和 CRC 相位以 5 Mbps 的比特率传输，而 CAN-FD 帧的其他相位以 1 Mbps 的比特率传输。

## 3.5 配置发送器延迟补偿

在本应用笔记的示例中，定义了一个函数用于启用 CAN-FD 的发送器延迟补偿。

### 3.5.1 MCAN\_配置发送器延迟补偿 FD

将 `DBTP` 寄存器中的 `TDC` 位设置为 1，以启用发射机延迟补偿。`TDCR` 寄存器中的 `TDCO` 字段设置发送器延迟补偿偏移量。

此函数将第二个采样点设置在 `mcan_rx` 位时间的中间。

```
发射机延迟补偿偏移量 = MCAN_TIME_QUANTA_NUM_DATA / 2
```

其中，宏 `MCAN_TIME_QUANTA_NUM_DATA` 是通过设置数据相位比特率阶跃来定义的。

## 3.6 步骤和结果

在此 CAN-FD 演示中，总线上有两个节点用于发送和接收数据。一个节点被设置为 A，另一个节点设置为 B。按节点 A 终端控制台上的任意键以触发一次发送。节点 B 接收该次发送的数据，并将其发送回节点 A。节点 A 接收数据，完成一次单次传输。

基本步骤是：

1. 硬件设置
  - 有关板卡设置和 CAN 连接的信息，请参阅[硬件环境](#)。
2. 构建并下载
  - 将此演示的软件包导入 MCUXpresso IDE 并进行构建。
  - 使用调试器下载可执行文件。
3. 设置 UART 终端程序
  - 检查 PC 上的 COM 编号，在设备管理器中将其模拟为 LPC LinkII。
  - 在 PC 上打开两个 UART 终端程序，并将一个评估板与一个 UART 终端程序连接。将通信协议配置为  $115200 + 8 + N + 1$ 。

#### 4. 运行

- 按下每个评估板上的 SW1 ( 复位 ) 按钮，复位两个评估板。一个评估板选择作为节点 A，另一个选择作为节点 B。按下节点 A 终端控制台上的任意键以触发一次发射。消息出现在节点 A 终端上如 图 4 所示，消息出现在节点 B 终端上如 图 5 所示。

```
Please select local node as A or B:
Note: Node B should start first.
Node:A
Press any key to trigger one-shot transmission

Received Frame ID: 0x123
Received Frame DATA: 0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9 0xa 0xb 0xc 0xd 0xe
0xf 0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1a 0x1b 0x1c 0x1d 0x1e
0x1f 0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2a 0x2b 0x2c 0x2d 0x2e
0x2f 0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3a 0x3b 0x3c 0x3d 0x3e
0x3f
Press any key to trigger the next transmission!
```

图 4. 打印在节点 A 终端上的消息

```
Please select local node as A or B:
Note: Node B should start first.
Node:B
Start to Wait data from Node A

Received Frame ID: 0x321
Received Frame DATA: 0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9 0xa 0xb 0xc 0xd 0xe
0xf 0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1a 0x1b 0x1c 0x1d 0x1e
0x1f 0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2a 0x2b 0x2c 0x2d 0x2e
0x2f 0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3a 0x3b 0x3c 0x3d 0x3e
0x3f
Wait Node A to trigger the next transmission!
```

图 5. 打印在节点 B 终端上的消息

**How To Reach Us**

**Home Page:**

[nxp.com](http://nxp.com)

**Web Support:**

[nxp.com/support](http://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](http://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](mailto: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 word 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. 2020-2021.

All rights reserved.

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

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: 2020 年 3 月 23 日

Document identifier: AN12799

