

1 引言

LPC55S1x/LPC55S0 拥有带有 CAN-FD 功能的 CAN 控制器。LPC5500 系列的 SDK 软件包提供了 `mcan_interrupt_transfer` 示例，演示了如何使用 SDK 的 CAN API 来传输 CAN 帧。基于此示例，本文描述了如何使用 CAN-FD 的比特率切换和发送延迟补偿功能。启用这两个功能可以提高吞吐量，并消除由收发器延迟引起的误码。

2 CAN-FD

国际标准 ISO 11898-1:2015 中定义了 CAN-FD 的实现。为了快速开始使用 CAN-FD，针对不熟悉 CAN 的用户，本节会介绍 CAN-FD 的一些基本概念。

2.1 CAN 和 CAN-FD 的不同

传统的 CAN 和 CAN-FD 之间主要有两个区别。首先是 CAN-FD 可以使用比传统 CAN 高得多的比特率。传统 CAN 的比特率限制为 1 Mbit/s。CAN-FD 没有理论的极限值，但实际上会受到收发器的限制。第二个区别是每个 CAN 消息所含的数据量增加了。传统的 CAN 最多为 8 个字节，而对于 CAN-FD，每条消息的最大值增加了 8 倍至 64 个字节。

随着每个 CAN 消息所含数据量的增加，CAN-FD 每帧需要更高的比特率，以减少通信中的时间延迟并提高实时性。通过启用比特率切换功能，CAN-FD 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。为了使数据域的比特时间比发送延迟更短，引入了延迟补偿。如果没有发送延迟补偿，CAN-FD 数据相位中的比特率会受到发送延迟的限制。

2.2 比特率切换

在 CAN-FD 的帧控制相位中，数据域和 CRC 域与每一帧的开始和结束域相比，以更高的比特率进行传输。

通过设置 CCCR 寄存器中的 BRSE 位，可以启用比特率切换功能。启用比特率切换时，我们还需要正确设置仲裁相位比特率（arbitration phase bit rate）（启用比特率切换之前）和数据域比特率（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。

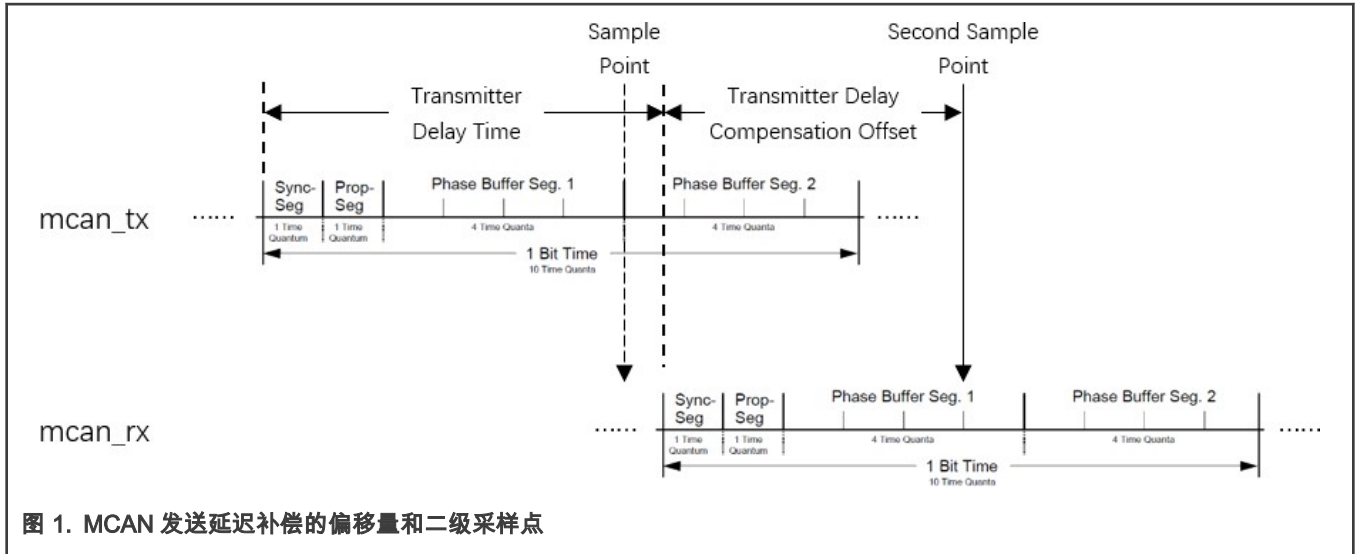
2.3 收发器延迟补偿

MCAN 的协议单元已实现延迟补偿机制，以补偿收发器的延迟。通过收发器延迟补偿，可以对数据位时间短发送延迟的情况进行配置，这在新的 ISO11898-1 中有详细介绍。可以通过设置 DBTP 寄存器中的 TDC 位来启用该功能。TDCR 寄存器中的 TDCO 字段用于设置收发器延迟补偿的偏移量。偏移量定义了从 `m_can_tx` 到 `m_can_rx` 的测量延迟与二级采样点之间的距离。发送延迟补偿的偏移量和二级采样点如 [图 1](#) 所示。

目录

1	引言.....	1
2	CAN-FD.....	1
3	示例演示.....	2
4	修订历史.....	6





3 示例演示

本节简要介绍了如何使用 CAN-FD 传输数据以及启用比特率切换功能和发送延迟补偿。这些基于 LPC5500 系列的 SDK `mcan_interrupt_transfer` 示例。

3.1 硬件环境

- 板子
 - 2 块 LPCXpresso55S16 板子或者 2 块 LPCXpresso55S08 板子
- 其他
 - 2 根 Micro USB 线
 - 1 根 120 欧姆端接的 CAN 线
 - PC
- 板子设置

以 LPC55S16 为例，CAN shield 连接到测试板。用 micro USB 线连接 PC 和测试板上的 J19，用于加载和运行 demo。这也用于 PC 上 UART 终端的 UART 通信。板子的设置和 CAN 连接如 图 2 所示。



图 2. 板子设置和 CAN 连接

3.2 软件环境

- 工具链
 - Keil 或 MCUXpresso11.1 或更高版本
- 软件包
 - LPC55S16 SDK (2.9.0)/LPC55S08 SDK (2.9.0)或更高版本
- UART 终端程序
 - PuTTY 或类似的程序

3.3 使用 CAN-FD 步骤概述

LPC500 系列 SDK 驱动程序示例：MCAN 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
 - 划分 MCAN 模块的系统时钟
- 初始化 MCAN
 - 启用 MCAN 时钟
 - 复位 MCAN 模块
 - 配置 MCAN 控制寄存器，启用 CAN-FD 和比特率切换功能
 - 设置仲裁相位比特率和数据相位比特率

- 启用发送延迟补偿
- 设置 Message RAM
- 设置 message ID 过滤器配置和参数
- 配置 Rx FIFO 和 Tx 缓冲区
- 进入 MCAN 正常模式
- 传输数据
 - 配置 Tx 数据帧数并发送
 - 接收数据

在 LPC5500 系列 SDK 的 `mcan_interrupt_transfer` 示例中，默认使用总线上的 CAN2.0(经典 CAN)节点。需要的数据相位比特率与仲裁相位比特率相同。它们应符合 CAN2.0 总线协议。

但是在本文的示例中，使用了总线上的 CAN-FD 节点。CAN-FD 的仲裁相位比特率设置为 1 Mbit/s，数据相位比特率设置为 5 Mbit/s。需要启用比特率切换和发送延迟补偿功能。

以下各节会介绍使用 CAN-FD 的一些关键步骤。

3.4 配置 CAN-FD 的比特率切换功能

启用 CAN-FD 比特率切换可以提高吞吐量。在本示例中，调用 `MCAN_SetBaudRate()` 函数设置仲裁相位比特率，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 将仲裁相位比特率设置为 1 Mbps

根据 CAN 定义规范，标称比特率是指：理想收发器在没有重新同步的情况下每秒发送的比特数。标称比特率和标称比特时间的关系是：标称比特时间=1/标称比特率。因此，如果将仲裁相位比特率设置为 1 Mbit/s，则仲裁相位的比特时间为 1 μ s。

时间量 (t_q) 是 MCAN 时钟周期的固定时间单位。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，我们需要定义一个变量，其类型为 `mcan_timing_config_t`，用于设置仲裁相位的比特时间。结构 `mcan_timing_config_t` 如 图 3 所示进行定义。

```
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 协议时序特征配置结构

预分频器 m 等于 $(preDivider+1)$ 。 t_q 的长度为 $t_q = (preDivider+1)/60\text{MHz} = (preDivider+1)/60 (\mu\text{s})$ ，其中 `preDivider` 为结构元素。

可以将仲裁相位的比特时间中的 t_q 总数设置为 4 到 385 个时间量。仲裁相位的比特时间长度 = `MCAN_TIME_QUANTA_NUM_ARBIT * t_q` 。

在此示例中，我们将宏 `MCAN_TIME_QUANTA_NUM_ARBIT` 定义为 20。仲裁相位的比特时间为 1 μ s， t_q 为 1/20 μ s，`preDivider` 的值为 2。

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

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

3.4.2 将数据相位比特率设置为 5 Mbit/s

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。

差异之一是，可以在 4 到 49 个时间粒度范围内，对数据相位的比特时间中的 t_q 总数进行编程。

在本示例中，我们将宏 MCAN_TIME_QUANTA_NUM_DATA 定义为 12，作为数据相位的比特时间中 t_q 的总数。元素 seg1 为 7，元素 seg2 为 2，元素 preDivider 为 4。调用函数 MCAN_SetBaudRateFD()，将数据相位比特率设置为 5 Mbit/s，这是板载收发器可以达到的最高比特率。

3.4.3 启用比特率切换功能

将 MCAN CCCR 寄存器的 BRSE 设置为 1，启用 CAN-FD 的比特率切换功能。在本文的示例中，CAN_FD 每帧的控制相位、数据相位和 CRC 相位以 5 Mbit/s 的比特率传输，而 CAN-FD 每帧的其他相位以 1 Mbit/s 的比特率传输。

3.5 配置发送延迟补偿

在本文的示例中，定义了一个函数，用于启用 CAN-FD 的发送延迟补偿。

```
static void MCAN_SetTransmitterDelayCompensationFD(CAN_Type *base)
{
    /* Cleaning previous TDC Setting. */
    base->DBTP &= ~(CAN_DBTP_TDC_MASK);

    /* Updating TDC Setting according to configuration structure. */
    base->DBTP |= CAN_DBTP_TDC(true);

    /* Cleaning previous TDCO Setting. */
    base->TDCR &= ~CAN_TDCR_TDCO_MASK;

    /* Updating TDCO Setting according to configuration structure. */
    base->TDCR |= CAN_TDCR_TDCO(MCAN_TIME_QUANTA_NUM_DATA/2);
}
```

3.5.1 MCAN_SetTransmitterDelayCompensationFD

通过将 DBTP 寄存器中的 TDC 设置为 1，可以启用发送延迟补偿。发送延迟补偿的偏移量由 TDCR 寄存器中的 TDCO 字段设置。

此函数将第二个采样点设置在 mcan_rx 比特时间的中间。

发送延迟补偿的偏移量=MCAN_TIME_QUANTA_NUM_DATA/2

其中宏 MCAN_TIME_QUANTA_NUM_DATA 是在设置数据相位比特率的步骤中定义的，该值在 fsl_mcan.c 中定义。

3.6 步骤和结果

在此 CAN-FD demo 中，总线上有两个节点用于发送和接收数据。一个节点被选择为 A，另一个节点被选择为 B。按下节点 A 所在终端控制台上的任意键以触发单次发送。节点 B 接收该单次发送的数据，并将其发送回节点 A。节点 A 接收数据，该单次传输完成。

基本步骤如下：

1. 硬件设置

有关板子设置和 CAN 连接的信息，请参见[硬件环境](#)。

2. 构建并下载

- 将该 demo 的软件包导入 MCUXpresso IDE 并进行构建。
- 使用 debugger 下载可执行文件。

3. 设置 UART 终端程序

- 在 PC 的设备管理器中查看用于 LPC-LinkII 仿真的 COM 编号。
- 在 PC 上打开两个 UART 终端程序，然后将一个测试板与一个 UART 终端程序连接。将通信协议配置为 115200+8+N+1。

4. 运行

按下每块测试板上的 SW1 (reset) 按钮来进行复位。一块测试板选择作为节点 A，另一块测试板选择作为节点 B。按下节点 A 所在终端控制台上的任意键以触发单次发送。消息显示在节点 A 的终端上，如 图 4 所示。消息显示在节点 B 的终端上，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. 在节点 A 终端上显示的消息

```
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. 在节点 B 终端上显示的消息

4 修订历史

版本号	日期	重大变化
0	02/2020	初始版本
1	03/2020	因 MCUXpresso 工具版本变化更新了第 3.2 节
2	10/2020	用 LPC5500 替代 LPC55S16

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 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.

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: 2020 年 10 月 30 日

Document identifier: AN12728

