AN14253

基于MCXN微控制器的USB转CAN-FD适配器

第1.0版—2024年4月16日 **应用笔记**

文档信息

信息	内容
关键词	AN14253、MCXN、MCXA、MCX_N9XX_EVK开发板、MCX_N9XX_FDRM开发板、软件开发套件（SDK）
摘要	本文介绍了使用MCX_N9XX_EVK和MCX_N9XX_FDRM开发板构建的两个USB转CAN-FD适配器的演示示例。

1 介绍

本应用笔记提供了两个演示示例，用于构建一个USB转CAN-FD适配器，实现USB与CAN总线之间的数据重新传输。采用MCX_N9XX_EVK和MCX_N9XX_FDRM开发板进行演示。恩智浦MCXN系列器件配备高速(HS)USB端口和CAN-FD控制器。HS USB的传输速率可高达480Mbit/s，足以在MCXN上以最高8Mbit/s的CAN波特率传输CAN-FD帧。

为了使系统易于使用并与其他设备兼容，示例采用USB CDC虚拟COM端口进行通讯，并使用Python GUI以ASCII格式显示CAN-FD信息。

2 CAN-FD

CAN-FD在国际标准ISO 11898-1:2015中定义。本章节向熟悉CAN使用的用户介绍CAN-FD的关键特性。如需了解更多CAN使用的详细信息，请访问网址：community.nxp.com/CAN。

2.1 CAN与CAN-FD的区别

经典CAN与CAN-FD之间有两大主要区别。首先，CAN-FD可以使用比经典CAN更高的比特率。经典CAN的速率限制为1Mbit/s，而CAN-FD理论上无限制，但实际应用中会受到收发器的限制。第二个主要区别在于每条CAN消息的数据量增加。经典CAN限制为8字节数据，而CAN-FD的限制为每条消息64字节数据，是经典CAN的8倍。随着每条CAN消息的数据量增加，CAN-FD帧需要更高的比特率以减少通讯延迟时间并提升实时性能。CAN-FD帧能够通过启用比特率切换功能以达到更高的比特率。另一方面，由于比特率更高，比特时间更短了。为了达成甚至比发射器延迟还要更短的数据阶段的比特时间，引入了延迟补偿。如果没有发送器延迟补偿，CAN-FD帧的数据阶段的比特率将受到发射器延迟的限制。

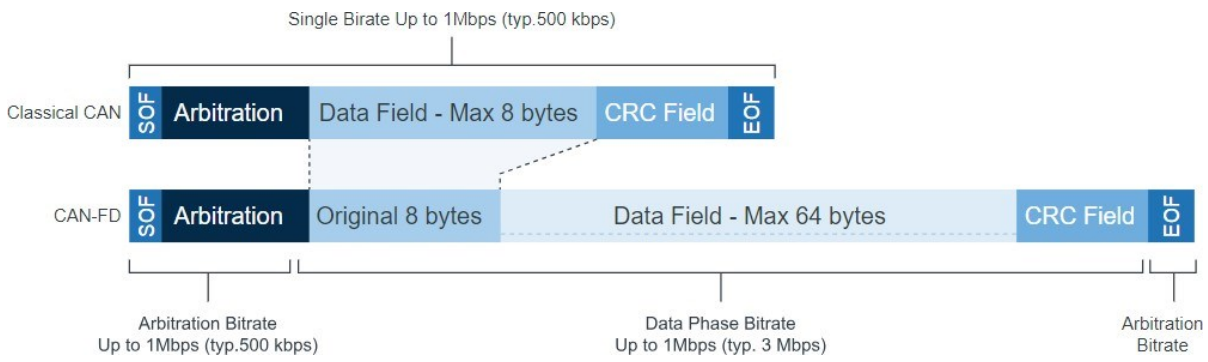

图1. CAN-FD与CAN帧

3 USB CDC类驱动程序

USB通讯设备类（USB CDC）是一种复合通用串行总线设备类别，包括多种接口，如自定义控制接口、数据接口、音频或与大容量存储相关接口等。在这种情况下，USB接口可用于实现虚拟COM端口（VCOM）的功能。PC上的VCOM端口有助于实现PC与嵌入式系统之间的通讯。更多有关USB的信息可通过以下网址获取：USB基础培训。

4 演示实现

4.1 概述

USB CDC使用两个USB物理批量端点在PC和MCU之间传输数据。每个端点负责一个单向的数据传输。

示例中每个管道使用两个缓冲区，一个用于USB至CAN-FD总线的通讯，另一个用于CAN-FD总线至USB的通讯。一旦数据到达MCU，MCU负责利用获得的信息构建CAN-FD帧并发送出去；在相反方向上，MCU接收到CAN-FD帧，然后从帧中提取数据，并通过USB CDC将数据发送给PC。

4.2 相关SDK示例

要实现本应用笔记中列出的步骤，用户必须具备USB CDC和CAN-FD使用的基础知识。MCXN的SDK中提供了以下两个示例：

- `mcxn9xxevk_flexcan_interrupt_transfer`示例：

该FlexCAN中断示例展示了如何以非阻塞中断方式使用FlexCAN驱动程序。

在此示例中，两块开发板通过CAN总线连接。当用户在终端上按下任意键时，端点A（开发板A）向端点B（开发板B）发送一个CAN消息。端点B收到消息后，将其内容打印到终端并将消息发回端点A。端点A增加接收到的消息，并等待用户发起下一次传输。

- `mcxn9xxevk_dev_cdc_vcom_bm`示例：

该虚拟COM工程是一个基于SDK的简单演示程序。它被枚举为COM端口，用户可使用Teraterm等终端工具打开此端口。该演示会回显接收到的所有字符，目的是展示如何构建USB的CDC类设备，并提供简单的工程，供进一步开发。

这两个示例均可从MCXN的SDK中导入，网址为：Welcome | MCUXpresso SDK Builder (nxp.com)。在深入阅读之前，用户需要熟悉上述两个示例，因为它们是USB-CAN适配器设计的基本构件。

4.3 硬件

本应用笔记中介绍的示例使用MCX_N9XX_EVK和MCX_N9XX_FDRM开发板。这两个开发板已集成了USB PHY和CAN收发器，无需进行任何硬件改造即可使用。应在board.h文件中使用如下的宏来选择合适的硬件。

```
/*! @brief the board name */
#define MCX_N9XX_EVK    (1U)
#define MCX_N9XX_FRDM   (2U)

#define BOARD_NAME MCX_N9XX_EVK
```

4.3.1 MCX-N9XX-EVK开发板

表1列出了基于MCX-N9XX-EVK开发板的USB-CAN适配器示例使用的GPIO引脚功能。

表1. 基于MCX-N9XX-EVK开发板的USB-CAN适配器使用的GPIO引脚

函数	GPIO	说明
CAN0_TX	P1_18	CAN总线发射信号
CAN0_RX	P1_19	CAN总线接收信号
USB1_DM	USB1_DM	HS USB DM
USB1_DP	USB1_DP	HS USB DP
UART_RXD	P1_8	调试UART RXD
UART_TXD	P1_9	调试UART TXD

4.3.2 MCX-N9XX-FRDM开发板

表2列出了基于MCX-N9XX-FRDM开发板的USB-CAN适配器示例的GPIO引脚功能。

表2. 基于MCX-N9XX-FRDM开发板的USB-CAN适配器使用的GPIO引脚

函数	GPIO	说明
CAN0_TX	P1_10	CAN总线发射信号
CAN0_RX	P1_11	CAN总线接收信号
USB1_DM	USB1_DM	HS USB DM
USB1_DP	USB1_DP	HS USB DP
UART_RXD	P1_8	调试UART RXD
UART_TXD	P1_9	调试UART TXD

4.4 软件

该软件基于两个裸板SDK示例：USB设备CDC VCOM和FlexCAN中断。集成这两个示例后，在以它们为基础的应用程序中采用了一个简单的串行协议。该协议能够将CAN消息转换为ASCII串行消息，并通过USB设备CDC发送。在本例中，消息被发送到Python接口，反之亦然。

按照下列步骤创建USB转CAN工程示例：

1. 使用`mcxn9xxevk_dev_cdc_vcom_bm`作为基础。
2. 集成`mcxn9xxevk_flexcan_interrupt_transfer`演示。
3. 将CAN TxD和RxD引脚配置复制到`pin_mux.c`文件。
4. 将`fsl_flexcan`驱动程序集成到工程的`drivers`文件夹中。
5. 集成`flexcan_interrupt_transfer.c`文件中的函数。
6. 创建一个适配层，用于CAN消息与串行消息之间的转换。
7. 在USB回调函数（***USB_DeviceCdcVcomCallback***）中，确定在哪里处理接收到的消息，将其转换为CAN消息并发送。
8. 在CAN回调函数中，确定接收完成消息，以便知道何时将CAN帧转换为串行消息并通过USB CDC发送。

MCXN软件示例可从以下链接获得：https://github.com/nxp-appcodehub/an-usb-to-can-adaptor-mcxn947。

图2所示为本示例的概览设计框图。

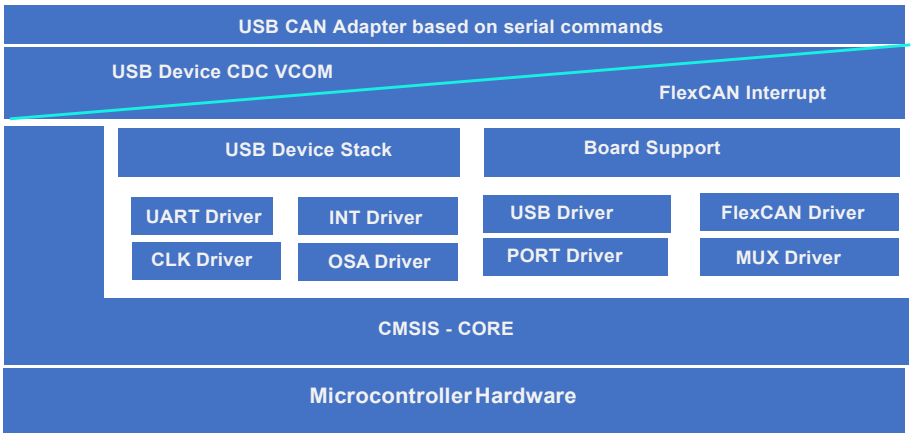

图2. 软件结构框图

此应用的主要函数位于表3所列的文件中。

表3. CAN文件

序号	文件名	说明
1	*can_interface.c*	包含所有CAN相关函数的文件，如CAN发送、CAN接收和FlexCAN初始化函数。
2	*usb_cdc_vcom.c*	包含所有USB相关函数的文件，包含USB CDC发送、USB CDC接收及USB初始化函数。
3	*usb_to_can.c*	支持串行协议的文件，包含用于解析消息的接收输入函数。
4	*usb_can_adapter.c*	包含主函数以调用初始化的文件。

4.5 串行命令帧

USB-CAN适配器在主机上注册为虚拟串行端口。为了方便人机交互，CAN命令在Python界面上以ASCII字符形式接收。同样，此界面发送的ASCII命令在发送前会转换为CAN命令。

为此，必须按照表4所示的特定格式来创建帧。

表4. 帧格式

FD ID	帧起始	CAN ID	DLC	数据
两个字符	1个字符	3个字符	1个字符	2-128个字符，具体取决于DLC

- **FD ID**：字符“FD”，用来标识帧是否为CAN-FD。
- **帧起始**：ASCII字符‘s’或‘S’，用来标识CAN帧的起始位置。
- **CAN ID**：3个字符，有效值为“0至9”或“A至F”，对应实际CAN ID的十六进制值。
- **DLC**：单个字符。有效的DLC选项见表5。

表5. 有效的DLC选项

DLC值	字节长度	字符数
1	1	2
2	2	4

表5. 有效的DLC选项（续）

DLC值	字节长度	字符数
3	3	6
4	4	8
5	5	10
6	6	12
7	7	14
8	8	16
10	16	32
13	32	64
15	64	128

- 数据：2至128个字符，有效值为“0至9”或“A至F”，对应CAN帧中的十六进制值。

下面的帧示例的帧格式示例见表6。

帧示例：FDs12381122334455667788

表6. 帧格式示例

FD ID	帧起始	CAN ID	DLC	数据
FD	s	123	8	1122334455667788

4.6 Python GUI界面

Python是近年来发展迅速的编程语言之一。Python社区开发了许多有用的库和工具，可支持自动化处理和界面开发。

此示例使用Python修订版3.10.10以及Tkinter模块和pySerial库。所有这些工具在网上都有大量的文档记录，还有许多优秀的示例可供参考。此示例的代码包含在工程的`python_gui`文件夹中。

4.7 界面介绍

图3所示为Python应用程序的GUI。

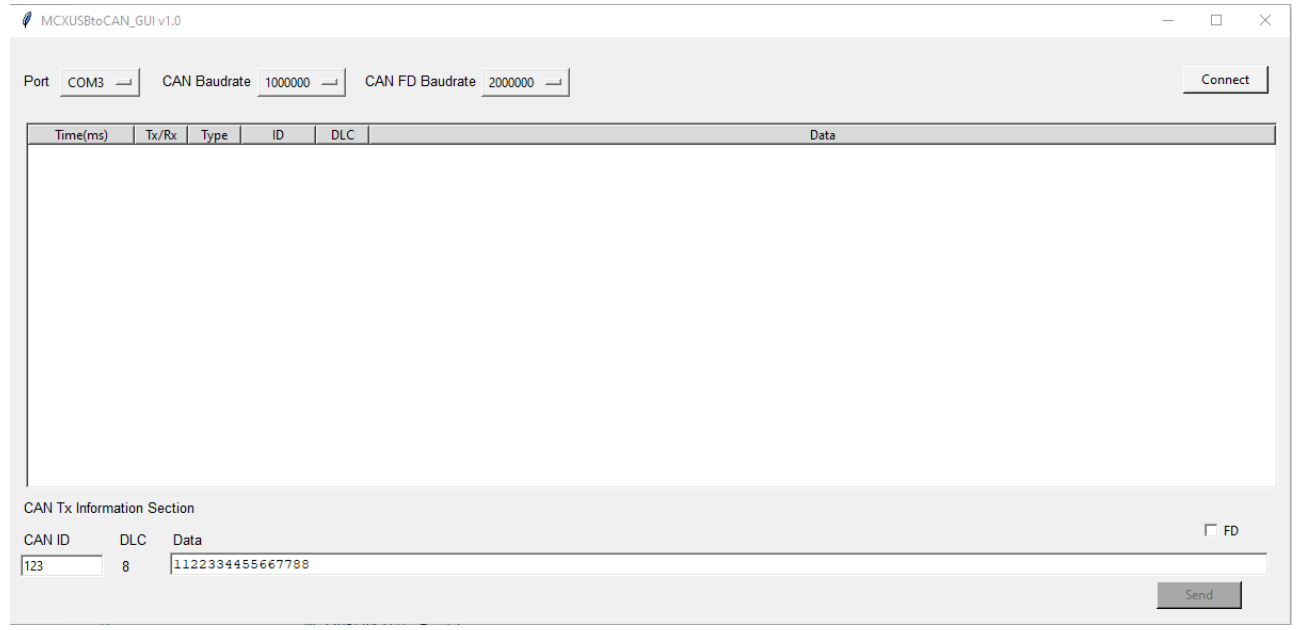

图3. Python GUI界面

- **Port**：此列表框支持您选择USB CDC板的COM端口。
- **CAN Baudrate**：选择仲裁阶段的波特率。
- **CAN-FD Baudrate**：选择数据阶段的波特率。
- **Connect**按钮：选择了端口和波特率之后应点击该按钮。这将启动与恩智浦设备的串行通讯。
- **CAN Tx Information Section**：在这个中心窗口中，用户可以查看接收和发送的CAN消息。
- **FD**：该复选框支持选择CAN或CAN-FD传输模式。此复选框不控制微控制器的配置，仅控制通过串口发送的串行消息。
- **CAN ID**：选择要发送消息的CAN ID。
- **DLC**：指示数据长度的DLC。如果数据长度不允许，则显示错误。
- **Data**：要发送的消息。长度应为偶数，根据DLC说明，长度为2至16、32、64或128个字符。
- **Send**：发送按钮

5 运行演示

以下示例展示了如何使用USB转CAN适配器与CAN设备进行通讯或监测CAN网络中的通讯。

5.1 直接通讯

此示例需要两块开发板。一块运行USB转CAN适配器代码，另一块运行`mcxn9xxevk_flexcan_interrupt_transfer`演示工程。

准备示例步骤如下：

- 将两块开发板的J5调试USB端口都与PC用USB线连接。
- 使用另一根USB线将运行USB转CAN代码的开发板的J27 USB设备端口与PC相连。
- 开发板间的CAN连接方式参见图4。

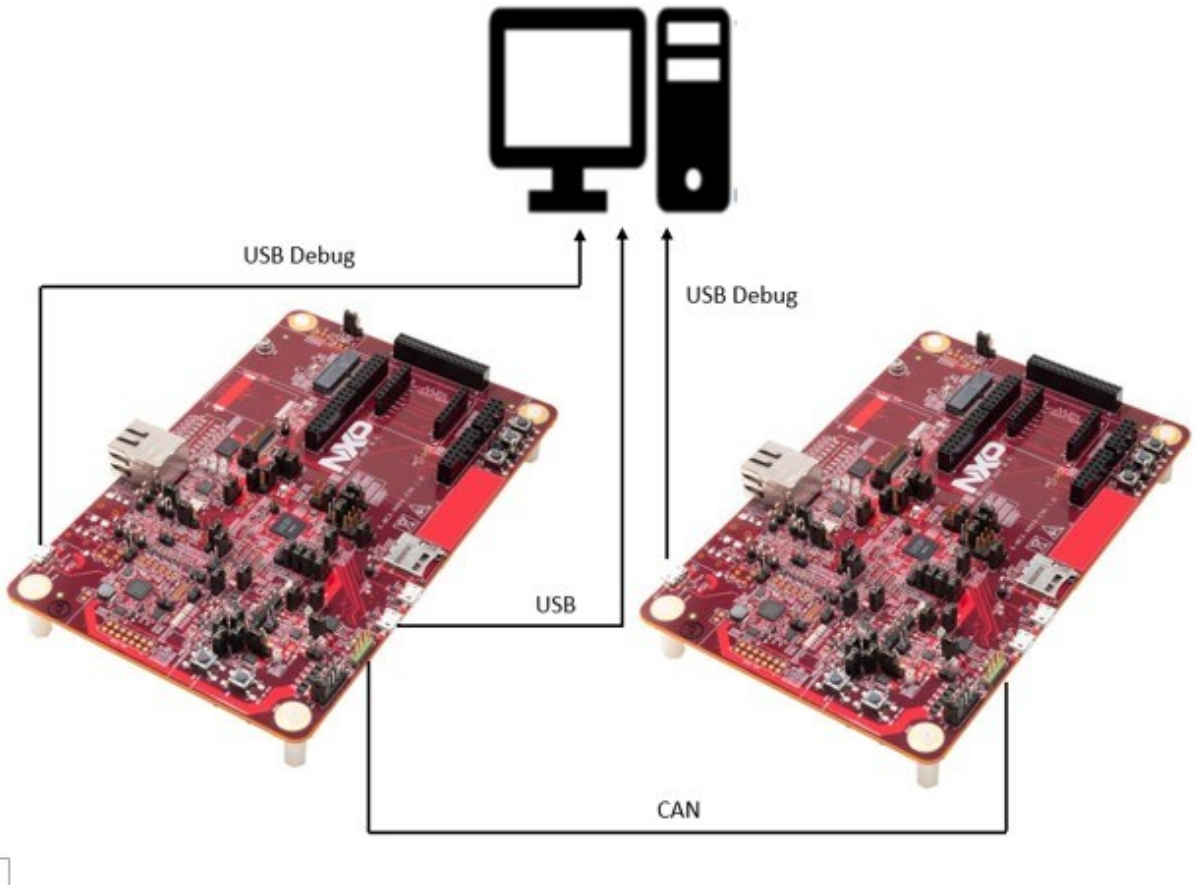

图4. 开发板间的CAN连接

表7. 演示运行后的串行终端

Node A USB转CAN		Node B CAN中断演示	
信号名称	开发板位置	信号名称	开发板位置
CANH	J29-1	CANH	J29-1

表7. 演示运行后的串行终端（续）

Node A USB转CAN		Node B CAN中断演示	
信号名称	电路板位置	信号名称	电路板位置
CANL	J29-2	CANL	J29-2
GND	J29-4	GND	J29-4

- 将示例代码下载到两块开发板上。一块开发板必须使用本应用笔记附带的USB转CAN适配器源代码进行编程，另一块则需使用直接从MCXN9 SDK导入的`flexcan_interrupt_transfer`演示工程来编程。
- 在装有`mcxn9xxevk_flexcan_interrupt_transfer`演示工程的开发板上，按照以下设置打开PC上的串行终端：
  - 波特率：115200
  - 数据位：8
  - 校验位：无
  - 停止位：1
  - 流控：无
- 按下开发板的复位按钮或在IDE中启动调试器来开始运行演示。

运行示例步骤：

1. 打开Python界面程序`MCXUSBtoCAN_GUI.py`或`MCXUSBtoCAN_GUI.exe`。
2. 选择USB CDC对应的COM。
3. 此示例中，所使用的CAN波特率为1000000，CAN-FD波特率为2000000。
4. 点击**Connect**按钮。
5. 设置**FD**复选框。
6. 在`mcxn9xxevk_flexcan_interrupt_transfer`演示中，选择节点A选项。
7. 在串行终端按任意键发送一条CAN消息。
8. 在Data区域输入值01并点击**Send**按钮。
9. 现在，重复步骤7和8。**mcxn9xxevk_flexcan_interrupt_transfer**演示工程在发送一条CAN消息后进入等待循环，等着接收消息，或接收到消息后，保持等待，直到使用终端发送一条CAN消息。

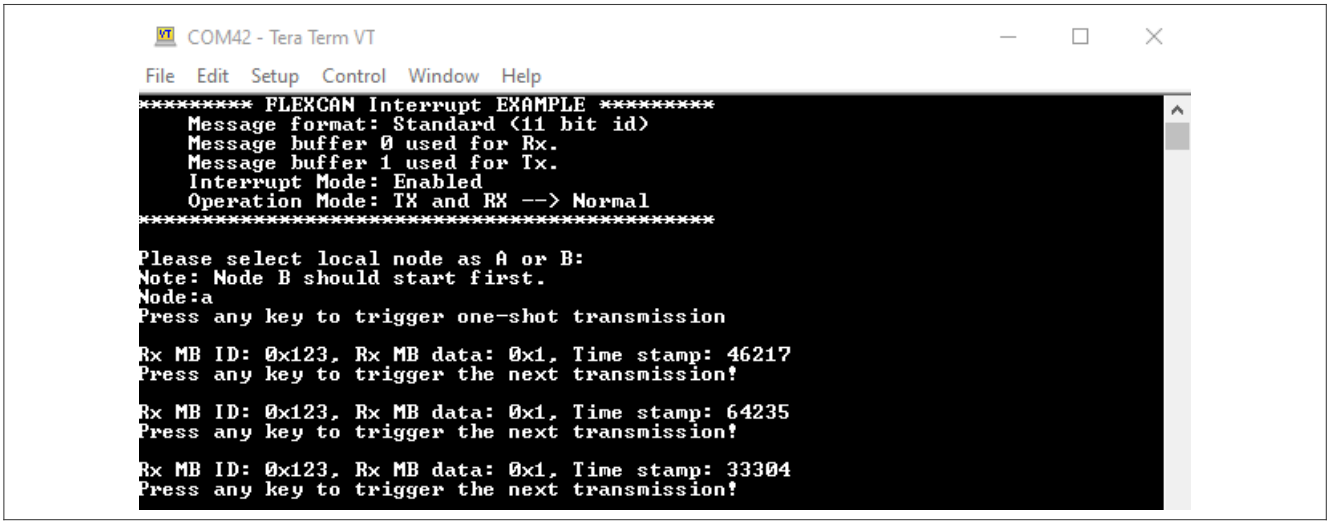

图5. 演示运行后的Python GUI界面

5.2 监测CAN网络

本示例展示了包含两个及以上设备的CAN网络，目的是监测网络上的所有流量。此示例要求一块搭载USB转CAN适配器代码的MCX_N9XX_EVK或MCX_N9XX_FRDM开发板接入网络。此演示使用两块恩智浦开发板作为CAN网络，两块开发板上都运行**can_interrupt_transfer**演示。

准备示例：

1. 使用USB线将开发板上的J5调试USB端口与PC相连。
2. 使用USB线将PC与开发板上的J27 USB设备端口相连。
3. CAN与网络的连接如图6所示：

表8. CAN连接：节点A USB转CAN

信号名称	开发板位置
CANH	J29-1
CANL	J29-2
GND	J29-4

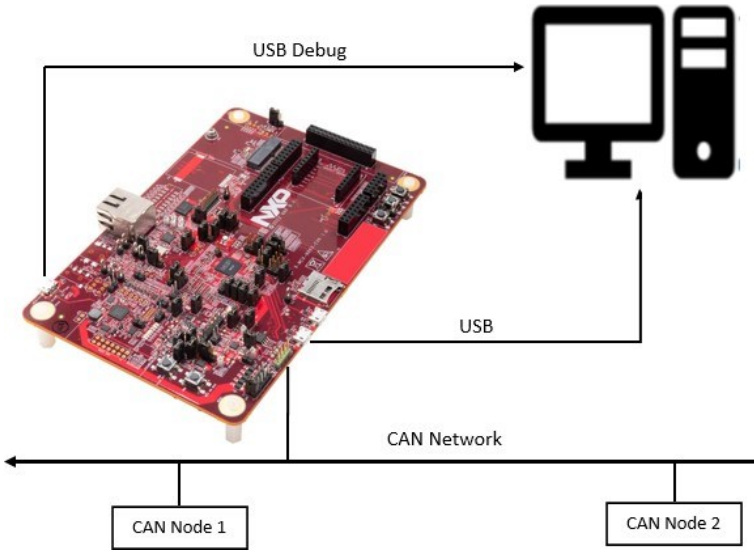

图6. CAN与网络的连接

4. 将示例代码下载到开发板上。
5. 按下开发板上的**Reset**按钮或在IDE中启动调试器来运行演示。

运行示例

1. 打开Python界面程序`MCXUSBtoCAN_GUI.py`或`MCXUSBtoCAN_GUI.exe`
2. 选择USB CDC对应的COM。
3. 本例使用的CAN波特率为1000000，CAN-FD波特率为2000000。
4. 点击**Connect**按钮。
5. 设置**FD**复选框。
6. 在CAN网络上开始传输数据，并检查在窗口区域显示的CAN流量。见图7。

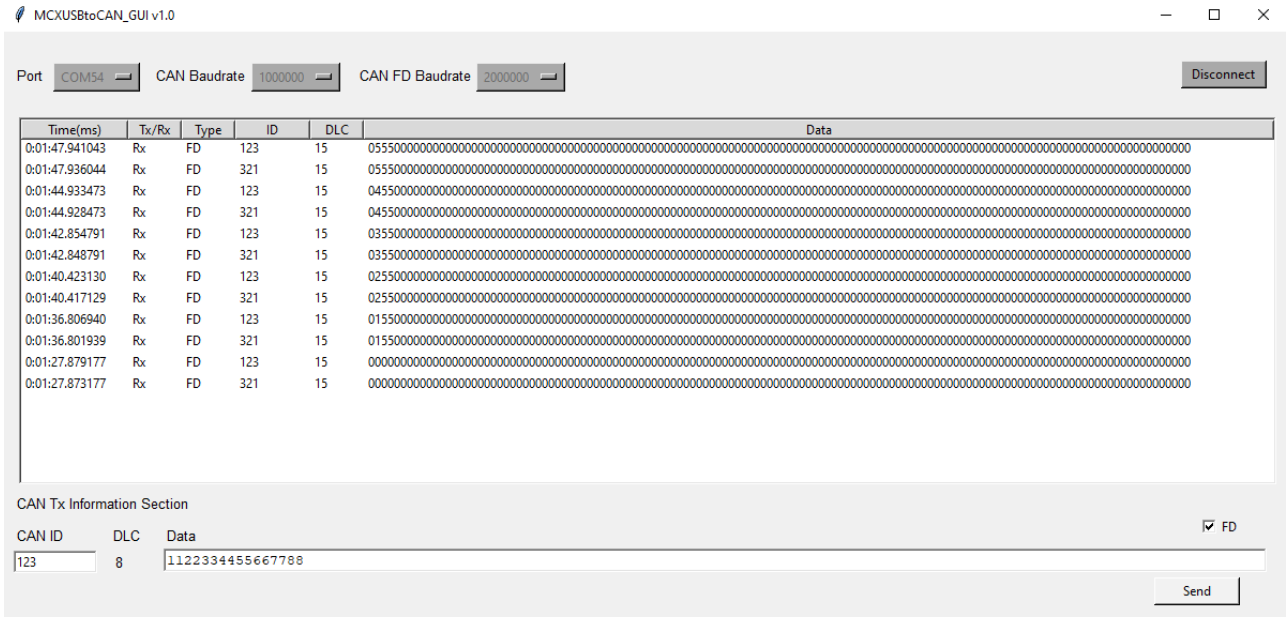

图7. 展示网络上的CAN流量的Python界面

6 缩略语

表9列出并解释了本文中所用的缩略语和缩写。

表9. 缩略语

术语	说明
CAN	控制器局域网
CDC	通讯设备类
CAN-FD	具有灵活数据速率的CAN
IDE	集成设计环境
MCU	微控制器单元
SDK	软件开发套件
USB	通用串行总线
VCOM端口	虚拟COM（通讯）端口

7 关于本文中源代码的说明

本文中所示的示例代码具有以下版权和BSD-3-Clause许可：

2024年恩智浦版权所有；在满足以下条件的情况下，可以源代码和二进制文件的形式重新分发和使用本源代码（无论是否经过修改）：

1. 重新分发源代码必须保留上述版权声明、这些条件和以下免责声明。
2. 以二进制文件形式重新分发时，必须在文档和/或随分发提供的其他材料中复制上述版权声明、这些条件和以下免责声明。
3. 未经事先书面许可，不得使用版权所有者的姓名或参与者的姓名为本软件的衍生产品进行背书或推广。

本软件由版权所有者和参与者“按原样”提供，不承担任何明示或暗示的担保责任，包括但不限于对适销性和特定用途适用性的暗示保证。在任何情况下，无论因何种原因或根据何种法律条例，版权所有者或参与者均不对因使用本软件而导致的任何直接、间接、偶然、特殊、惩戒性或后果性损害（包括但不限于采购替代商品或服务；使用损失、数据损失或利润损失或业务中断）承担责任，无论是因合同、严格责任还是侵权行为（包括疏忽或其他原因）造成的，即使事先被告知有此类损害的可能性也不例外。

8 修订历史

表10列出了本文档的修订历史。

表10. 修订历史

文档ID	发布日期	说明
AN14253 v.1.0	2024年4月16日	首次公开发布

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

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.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com.cn/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. 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 B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

MCX — is a trademark of NXP B.V.

目录

Date of release: 16 April 2024
Document identifier: AN14253