

M460 简单的 CANFD 接收代码

NuMicro® 32位系列微控制器范例代码介绍

文件信息

应用简述	本范例代码使用 M460 系列的 CANFD 接收报文，并打印出来
BSP 版本	M460_Series_BSP_CMSIS_V3.00.001
开发平台	NuMaker-M467HJ V1.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller and microprocessor based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1. 概述

这是一个使用 M460 系列微控制器（MCU）接收和打印 CANFD 消息的简单示例代码。

1.1 原理

M460 系列有一个功能很强的 CANFD 外设，为了不造成理解上的困难，此处仅介绍 M460 接收 CAN 总线上的所有报文并存放到 FIFO1 然后打印出来。SID 和 XID 过滤器，配置接收的、或拒绝的报文的 ID。

接收框图如图 1-1。

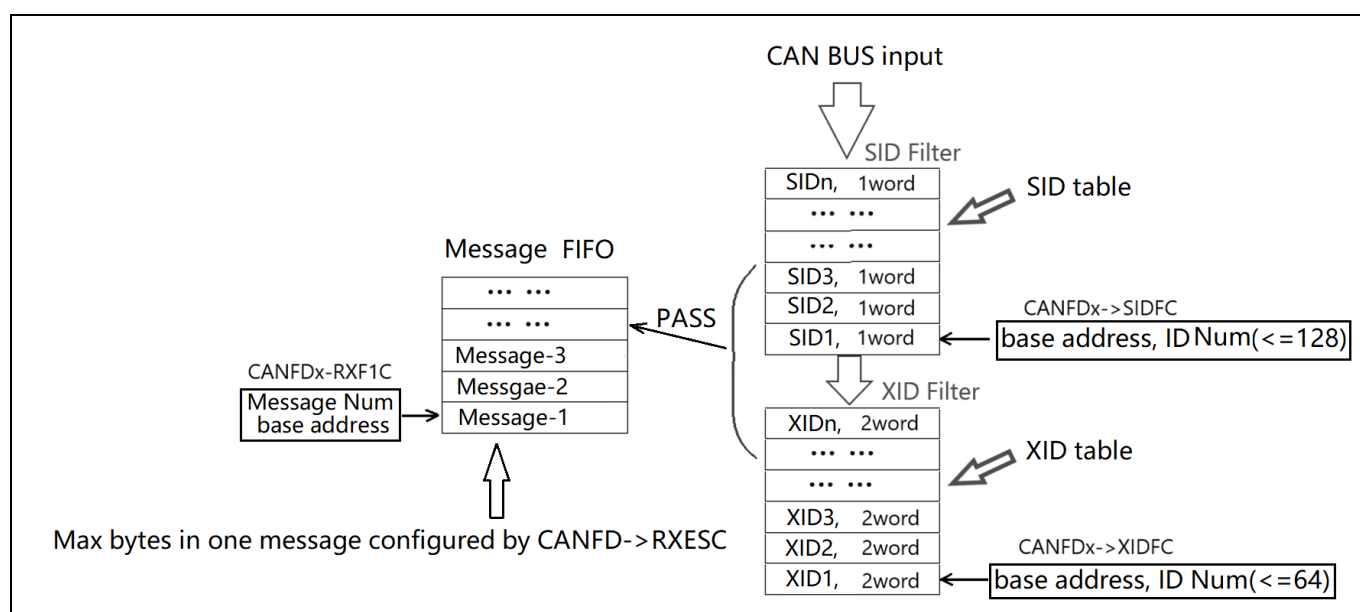


图 1-1 CANFD FIFO 接收结构

图 1-2 展示了标准 ID 的数据结构。

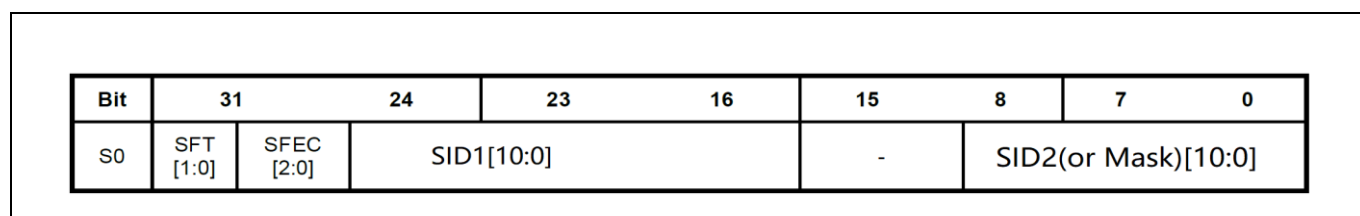


图 1-2 标准 ID 数据结构

图 1-3 显示了扩展 ID 的数据结构。

Bit	31	24	23	16	15	8	7	0
F0	EFECV [2:0]		XID1[28:0]					
F0	EFT [1:0]	-	XID2(or MASK)[28:0]					

图 1-3 扩展 ID 数据结构

1.2 执行结果

执行本代码的结果如图 1-4，UART 会打印出 CAN 总线上的所有报文。

```
COM4 - PuTTY
CAN FD FIFO1 Rxd

IR =0x00000000
Rx FIFO1(Standard ID) ID = 0x00000116
Message Data(08 bytes) : 00 ,01 ,02 ,03 ,04 ,05 ,06 ,07 ,

IR =0x00000000
Rx FIFO1(Standard ID) ID = 0x00000113
Message Data(12 bytes) : 00 ,01 ,02 ,03 ,04 ,05 ,06 ,07 ,08 ,09 ,10 ,11 ,

IR =0x00000000
Rx FIFO1(Standard ID) ID = 0x0000022F
Message Data(16 bytes) : 00 ,01 ,02 ,03 ,04 ,05 ,06 ,07 ,08 ,09 ,10 ,11 ,12 ,13 ,14 ,15 ,

IR =0x00000000
Rx FIFO1(Standard ID) ID = 0x00000333
Message Data(20 bytes) : 00 ,01 ,02 ,03 ,04 ,05 ,06 ,07 ,08 ,09 ,10 ,11 ,12 ,13 ,14 ,15 ,16 ,17 ,18 ,19 ,

IR =0x00000000
Rx FIFO1(Extended ID) ID = 0x00000221
Message Data(24 bytes) : 00 ,01 ,02 ,03 ,04 ,05 ,06 ,07 ,08 ,09 ,10 ,11 ,12 ,13 ,14 ,15 ,16 ,17 ,18 ,19 ,20 ,21 ,22 ,23 ,

IR =0x00000000
Rx FIFO1(Extended ID) ID = 0x00000227
Message Data(32 bytes) : 00 ,01 ,02 ,03 ,04 ,05 ,06 ,07 ,08 ,09 ,10 ,11 ,12 ,13 ,14 ,15 ,16 ,17 ,18 ,19 ,20 ,21 ,22 ,23 ,24 ,25 ,26 ,27 ,28 ,29 ,30 ,31 ,

IR =0x00000000
Rx FIFO1(Extended ID) ID = 0x00003333
Message Data(48 bytes) : 00 ,01 ,02 ,03 ,04 ,05 ,06 ,07 ,08 ,09 ,10 ,11 ,12 ,13 ,14 ,15 ,16 ,17 ,18 ,19 ,20 ,21 ,22 ,23 ,24 ,25 ,26 ,27 ,28 ,29 ,30 ,31 ,32 ,33 ,34 ,35 ,36 ,37 ,38 ,39 ,40 ,41 ,42 ,43 ,44 ,45 ,46 ,47 ,

IR =0x00000000
Rx FIFO1(Extended ID) ID = 0x00044444
Message Data(64 bytes) : 00 ,01 ,02 ,03 ,04 ,05 ,06 ,07 ,08 ,09 ,10 ,11 ,12 ,13 ,14 ,15 ,16 ,17 ,18 ,19 ,20 ,21 ,22 ,23 ,24 ,25 ,26 ,27 ,28 ,29 ,30 ,31 ,32 ,33 ,34 ,35 ,36 ,37 ,38 ,39 ,40 ,41 ,42 ,43 ,44 ,45 ,46 ,47 ,48 ,49 ,50 ,51 ,52 ,53 ,54 ,55 ,56 ,57 ,58 ,59 ,60 ,61 ,62 ,63 ,
```

图 1-4 打印收到的报文

2. 代码介绍

图 2-1 是位时钟分配，看此图，比较容易理解位时钟的配置。

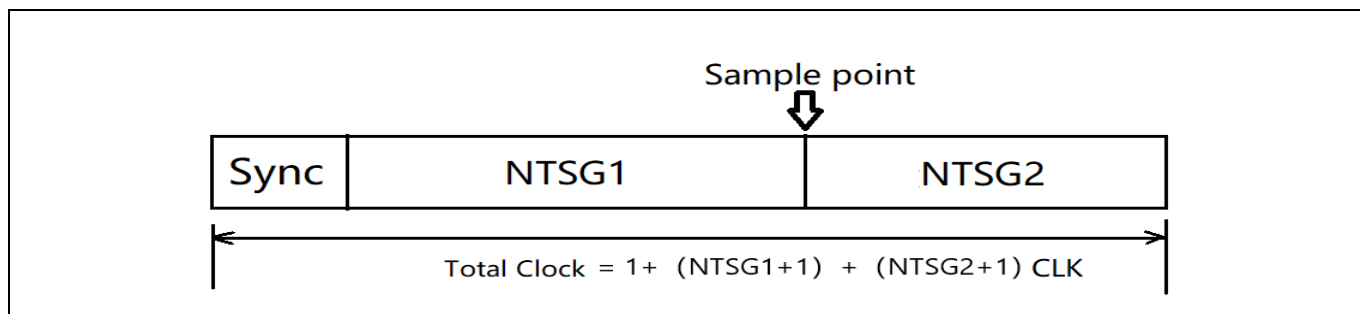


图 2-1 位时钟分配

配置位时钟的函数在 main.c 中

```
/*-----*/
/*          Init CAN FD bit Rate          */
/*-----*/
// input: psCanfd, The value must be CANFD0,CANFD1,CANFD2 or CANFD3
void CANFD_BitRate_Init(CANFD_T *psCanfd)
{
    // CANFD_CLK = 200M/5 = 40MHz

    psCanfd->NBTP = (3 << 25) + // NSJW = 3+1 =4 CLK
                   (1 << 16) + // NBRP = 1+1 =2 , // prescaler = 2
                   (13 << 8) + // NTSG1 = 13+1=14 CLK
                   (5 - 1); // NTSG2 = 5 CLK // One bit = 1+14+5 = 20 CLK

    psCanfd->DBTP = ((1 - 1) << 16) + // DBRP = 1 prescaler
                   ((6 - 1) << 8) + // DTSG1 = 6 CLK // One bit = 1+6+3 = 10 CLK
                   ((3 - 1) << 4) + // DTSG2 = 3 CLK
                   (2 - 1); // DSJW = 2CLK
}
```

标准 ID 和扩展 ID 的配置函数在 main.c 中，在 #if 1 和 #else 之间的代码，配置接收 CAN 总线上所有的报文。

```
void CANFD0_RX_Initial(void)
{
    // 0=>8Byte, 1=>12Byte, 2=>16Byte, 3=>20Byte, 4=>24Byte, 5=>32Byte, 6=>48Byte, 7=>64Byte
    CANFD0->RXESC=(CANFD0->RXESC & (~CANFD_RXESC_F1DS_Msk)) | (7 << CANFD_RXESC_F1DS_Pos);

    CANFD0->RXF1C = (60 << CANFD_RXF1C_F1WM_Pos) // watermark = 60 messages
                   | (64 << CANFD_RXF1C_F1S_Pos) // FIFO1 can hold 64 messages
                   | 0x300 ; // FIFO1 start address =0x40020000 +0x300

    CANFD0->SIDFC = (16 << 16) + 0 ; // 16-SID at 0 address
    CANFD0->XIDFC = (16 << 16) + 0x40 ; // 16-XID at 0x40 address
}
```

```
#if 1
    CANFD0_SID_Buff[0].VALUE = CANFD_RX_FIFO1_STD_MASK(0, 0); // receive all SID messages

    CANFD0_XID_Buff[0].LOWVALUE = CANFD_RX_FIFO1_EXT_MASK_LOW(0) ;
    CANFD0_XID_Buff[0].HIGHVALUE=CANFD_RX_FIFO1_EXT_MASK_HIGH(0); //receive all XID messages

#else
    .....
#endif

    CANFD_EnableInt(CANFD0, (CANFD_IE_TOOE_Msk | CANFD_IE_RF1NE_Msk), 0, 0, 0);
    NVIC_EnableIRQ(CANFD00_IRQn);
}
```

3. 软件与硬件需求

3.1 软件需求

- BSP 版本
 - M460_Series_BSP_CMSIS_V3.00.001
- IDE 版本
 - Keil uVersion 5.38

3.2 硬件需求

- 电路组件
 - NuMaker-M467HJ V1.0
- 示意图
 - 两个 NuMaker-M467HJ 板的 CANFD 连接如图 3-1，一个执行代码 EC_M460_CANFD_TX_Simply_V1.00 发送报文，另一板子执行本例代码接收报文

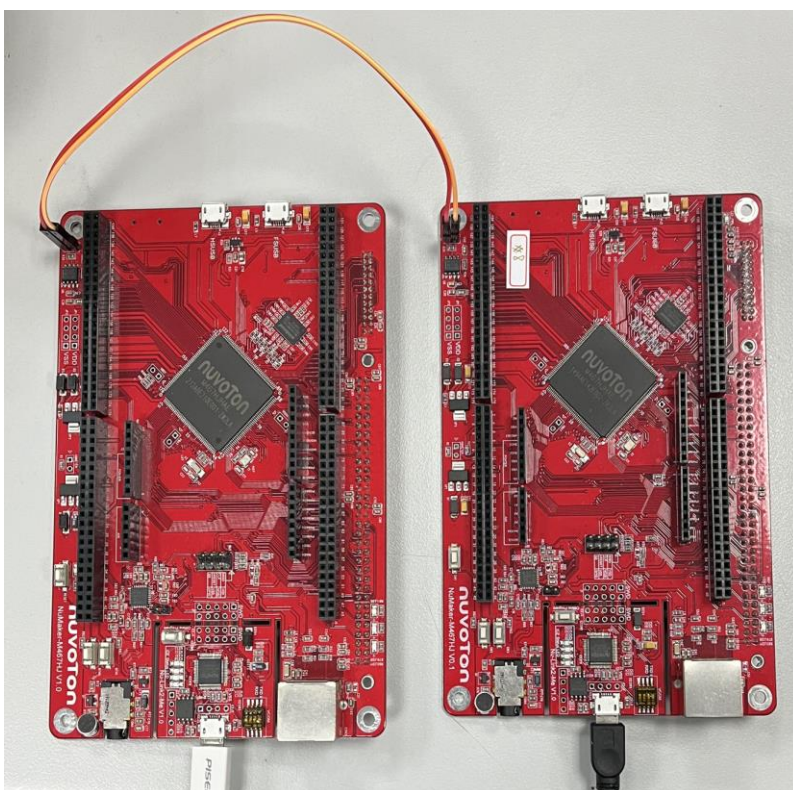


图 3-1 两个 NuMaker-M467HJ 板的 CANFD 接口连接

4. 目录信息








	EC_M460_CANFD_FIFO_RX_V1.00	
	Library	Sample code header and source files
	CMSIS	Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
	Device	CMSIS compliant device header file
	StdDriver	All peripheral driver header and source files
	SampleCode	
	ExampleCode	Source file of example code

图 4-1 目录信息

5. 范例程序执行

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 *M460_CANFD_FIFO_RX.uvprojx*。
2. 进入编译模式界面
 - 编译
 - 下载代码至内存
 - 进入 / 离开仿真模式
3. 进入仿真模式界面
 - 执行代码

6. 修订纪录

Date	Revision	Description
2023.06.23	1.00	初始发布。

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, “Insecure Usage”.

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer’s risk, and in the event that third parties lay claims to Nuvoton as a result of customer’s Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

*Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.*