M460 簡單的CANFD 發送代碼

NuMicro® 32位元系列微控制器範例代碼介紹

文件資訊

應用簡述	本範例代碼使用 M467 的 CANFD 介面發送訊息
BSP 版本	M460_Series_BSP_CMSIS_V3.00.001
開發平臺	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

使用 M467 做一個簡單的、CANFD 發送訊息的代碼，用於 M467 CANFD 的測試，也可演示 M467 CANFD 發送訊息的配置。

1.1 原理

M467 的 CANFD，每條等待發送的訊息由 “2wordsID + 資料位元組” 組成，訊息格式如圖 1-1.

Bit	31	24	23	16	15	8	7	0
T0	ESI / XTD / RTR		ID[28:0]					
T1	MM[7:0]		EFC / - / FDF / BRS / DLC[3:0]		MM[15:8]		-	
T2	DB3[7:0]		DB2[7:0]		DB1[7:0]		DB0[7:0]	
T3	DB7[7:0]		DB6[7:0]		DB5[7:0]		DB4[7:0]	

圖 1-1 等待發送的訊息的格式

一條訊息啟動發送後，可能因匯流排忙而處於等候狀態。M467 每個 CANFD 介面最多可配置 32 條等待發送訊息存放在 “專用 RAM 區” 。這個專用 RAM 區有 6144Bytes，用於存放等待發送的訊息、接收到的訊息、及其它訊息配置資訊。

圖 1-2 展示了與發送訊息相關的寄存器：多條等待發送的訊息在專用 RAM 區的首位址由寄存器 CANFDx->TXBC 配置，訊息資料位元組長度由寄存器 CANFDx->TXESC 配置。32 位寄存器 CANFDx->TXBRP 置 1 的位，啟動對應訊息發送，發送成功後自動清 0。

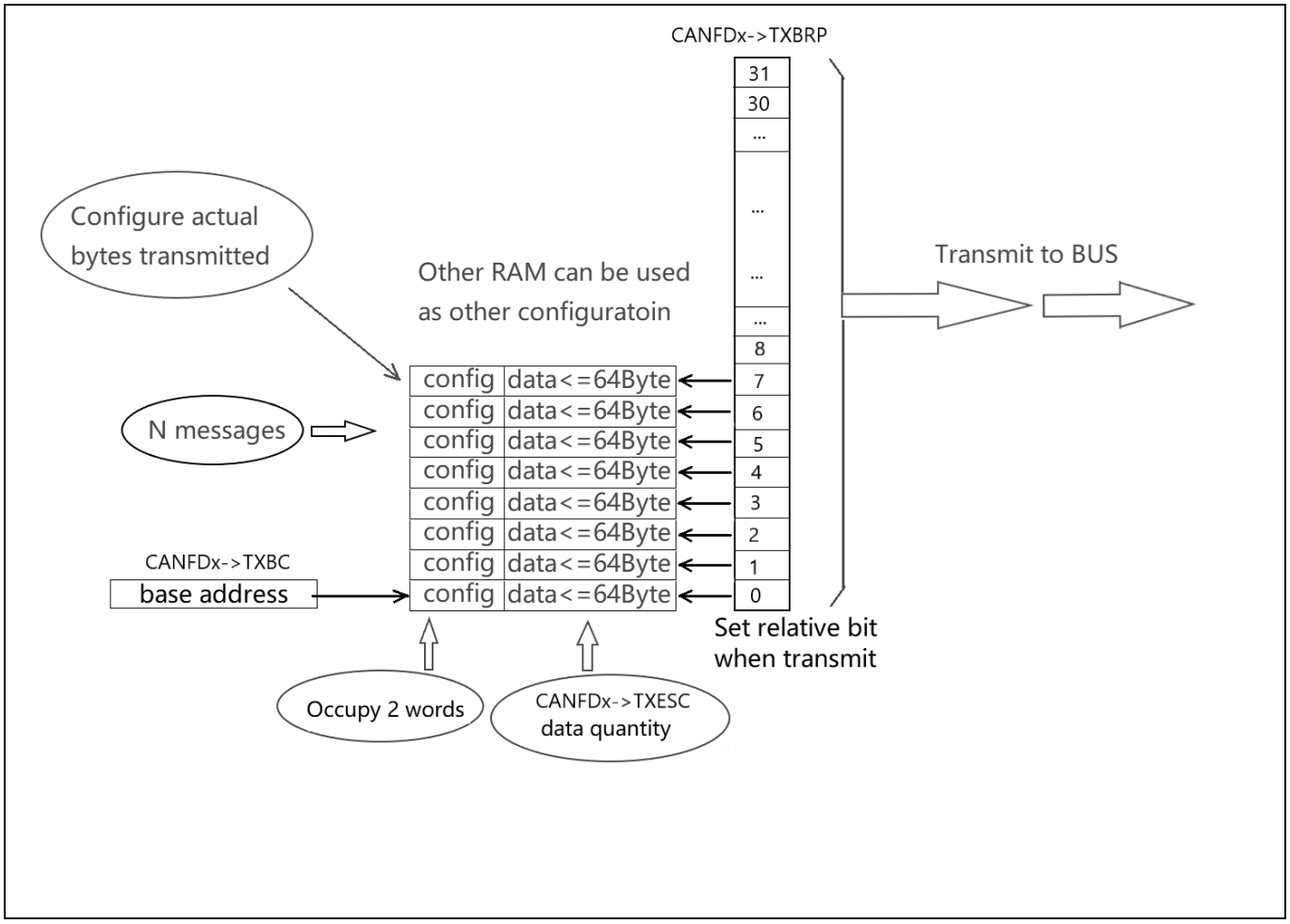

圖 1-2 與發送訊息相關的寄存器

本代碼先配置 CANFD 的位元率寄存器，再配置上述發送訊息資訊，最後，週期性的讓 CANFDx->TXBRP 對應位置 1，啟動訊息的週期性發送。

1.2 執行結果

本代碼在 NuMaker-M467HJ V1.0 板上測試，CANFD 介面連續發出訊息，可用另一塊 NuMaker-M467HJ 板子接收訊息，並把收到的訊息列印出來查看。本代碼迴圈發送 8 個不同格式的訊息，如圖 1-3。

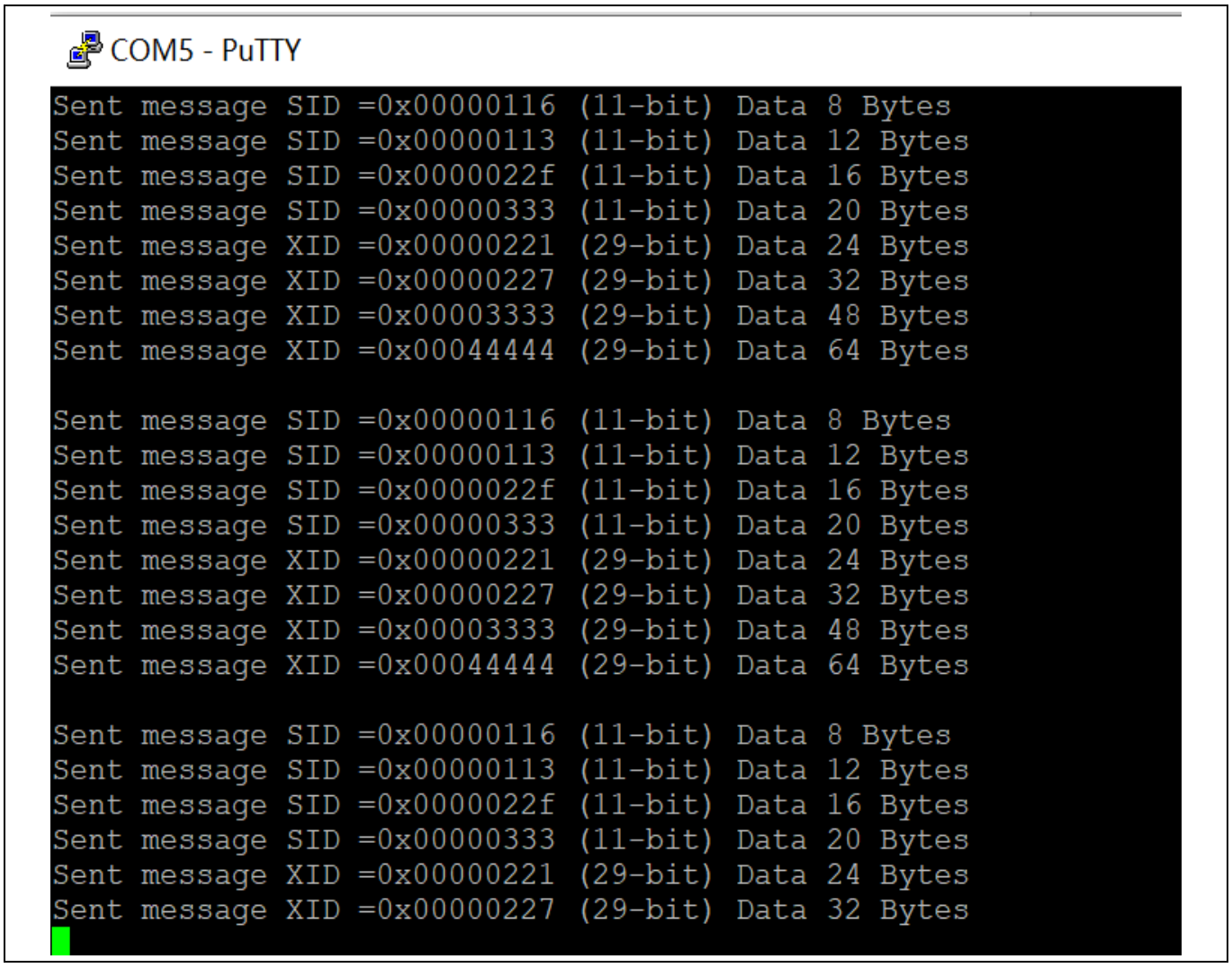

圖 1-3 執行結果，列印資訊

用另一個 NuMaker-M467HJ 板子的 CANFD 接收列印訊息，這個板子執行代碼：M460_Series_BSP_CMSIS_V3.00.001\SampleCode\StdDriver\CANFD_CANFD_TxRxINT，注意調整代碼中的 CANFD 介面到 PJ10,PJ11 腳位，就可列印出如圖 1-4 訊息。

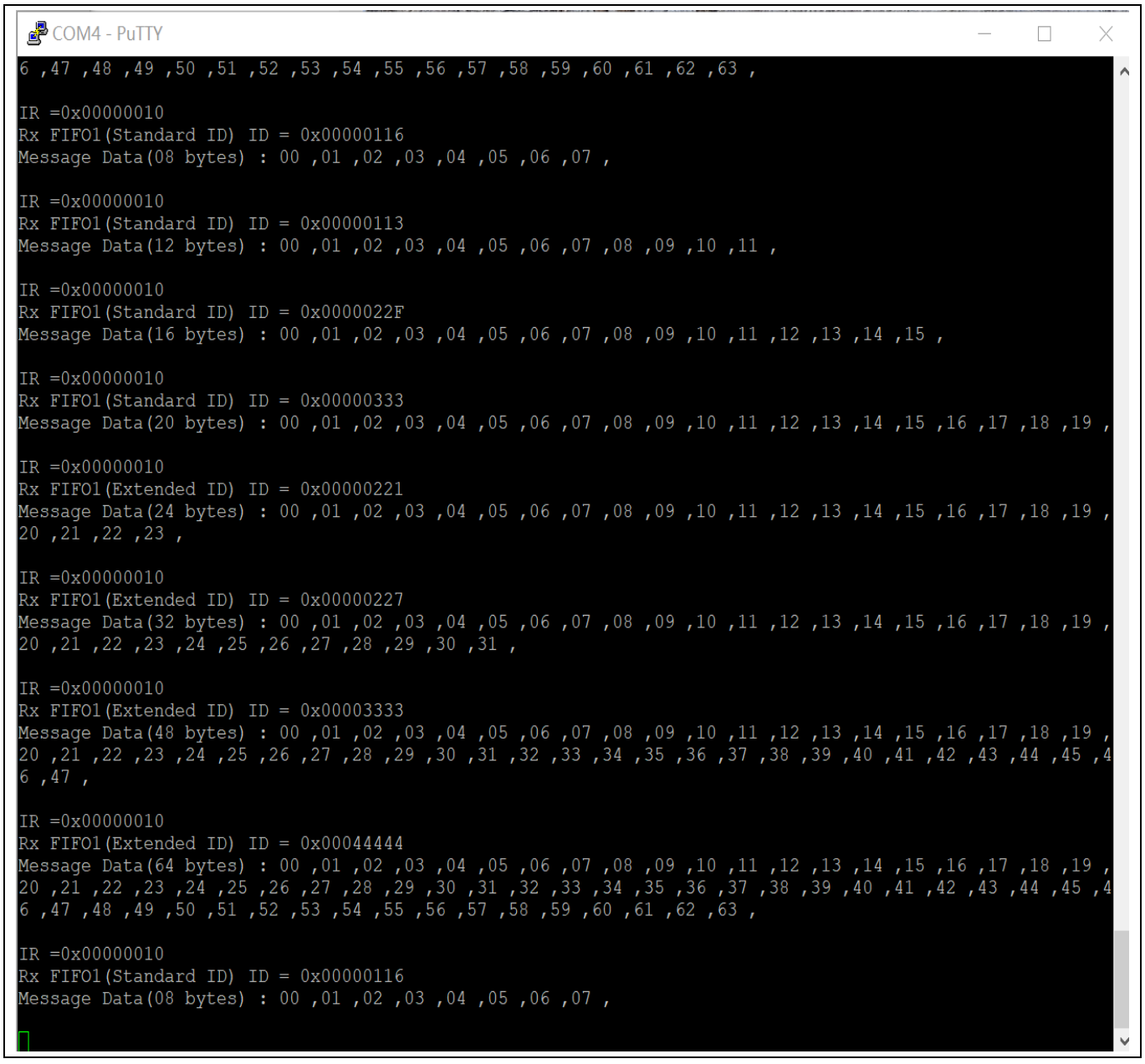

圖 1-4 CANFD 匯流排上的消息

2. 代碼介紹

配置 CANFD 傳輸速率，代碼位於 *main.c*。

```
/// 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  // One bit =1+14+5 =20 CLK
                    (5 - 1);                // NTSG2 = 5  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採樣點左右調整的最大時鐘數
}
```

CANFD 接收時，資料欄在 DTSG1 與 DTSG2 的中間採樣信號電平，其它域時鐘個數定義用 NTSG1 和 NTSG2。一位元的時鐘總數需再加一個時鐘的同步段。如圖 2-1。

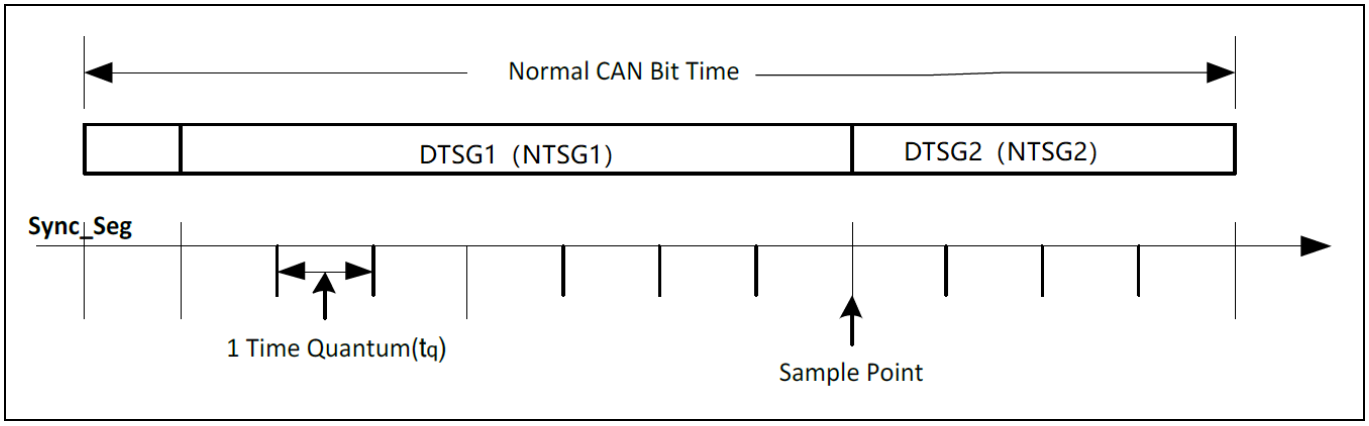

圖 2-1 CANFD 一位時間的定義

發送訊息開頭的 2words 配置消息 ID、資料長度等，這部分由函數 CANFD_Tx_Init()先配置好，資料欄在訊息發送前即時更新(當然發送訊息前，也可以更新這 2words 包頭)。

```
    // There configured 8 Txbuff elements
void CANFD_Tx_Init(CANFD_T *psCanfd, int16_t TxBuff_Quantity, int16_t DLC_Max, uint16_t
TxBuff_StartAddr)
{
    psCanfd->TXBC = (TxBuff_Quantity << CANFD_TXBC_NDTB_Pos)
                    + TxBuff_StartAddr ;                    // Start address of messages RAM

    psCanfd->TXESC = DLC_Max <<  CANFD_TXESC_TBDS_Pos;       // 7 mean 64 Bytes

// u32Config = 0x00300000 mean EFC=0，FDF=1, BRS=1 // If DLC <8,the Bytequantiry =DLC
```

```
    CANFD_TxBuff[0].u32Id       = 0x111 << 18 ;                    // SID = 0x111
    CANFD_TxBuff[0].u32Config   = 0x00300000 + 0x00080000 ;      // DLC=8 =8 Bytes
//CANFD_TxBuff[0].u32Config = 0x00300000 + 0x00050000 ;  // It has DLC data if(DLC <=8)

    CANFD_TxBuff[1].u32Id       = 0x113 << 18 ;                    // SID = 0x113
    CANFD_TxBuff[1].u32Config   = 0x00300000 + 0x00090000 ;      // DLC=9 =12 Bytes
    CANFD_TxBuff[2].u32Id       = 0x22F << 18 ;                    // SID = 0x22F
    CANFD_TxBuff[2].u32Config   = 0x00300000 + 0x000A0000 ;   // DLC=A =16 Bytes
    CANFD_TxBuff[3].u32Id       = 0x333 << 18 ;                    // SID = 0x333
    CANFD_TxBuff[3].u32Config   = 0x00300000 + 0x000B0000 ;      // DLC=B =20 Bytes

    CANFD_TxBuff[4].u32Id       = 0x40000000 + 0x221 ;             // XID = 0x221
    CANFD_TxBuff[4].u32Config   = 0x00300000 + 0x000C0000 ;     // DLC=C =24 Bytes
    CANFD_TxBuff[5].u32Id       = 0x40000000 + 0x227 ;             // XID = 0x227
    CANFD_TxBuff[5].u32Config   = 0x00300000 + 0x000D0000 ;     // DLC=D =32 Bytes
    CANFD_TxBuff[6].u32Id       = 0x40000000 + 0x3333 ;            // XID = 0x3333
    CANFD_TxBuff[6].u32Config   = 0x00300000 + 0x000E0000 ;      // DLC=E =48 Bytes
    CANFD_TxBuff[7].u32Id       = 0x40000000 + 0x44444 ;           // XID = 0x44444
    CANFD_TxBuff[7].u32Config   = 0x00300000 + 0x000F0000 ;      // DLC=F =64 Bytes
}
```

最多可配置 32 條等待發送的訊息，如果想啟動某條訊息的發送，只要在 CANFD 空閒時，把寄存器 TXBAR 的對應位置 1 就可以了。

```
 while (CANFD_GET_COMMUNICATION_STATE(psCanfd) != eCANFD_IDLE)
 {
   if (--u32TimeOutCnt == 0) break;            // waiting for CANFD idle
 }

 psCanfd->TXBAR = 1 << 3 ;       // Tx_Element_Buff 3 Request transmitting
```

3. 軟體與硬體需求

3.1 軟體需求

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

3.2 硬體需求

- 電路元件
  - NuMaker-M467HJV1.0
- 示意圖
  - 將兩個 NuMaker-M467HJ 板的 CANFD 介面連接，以顯示範例代碼的執行結果。

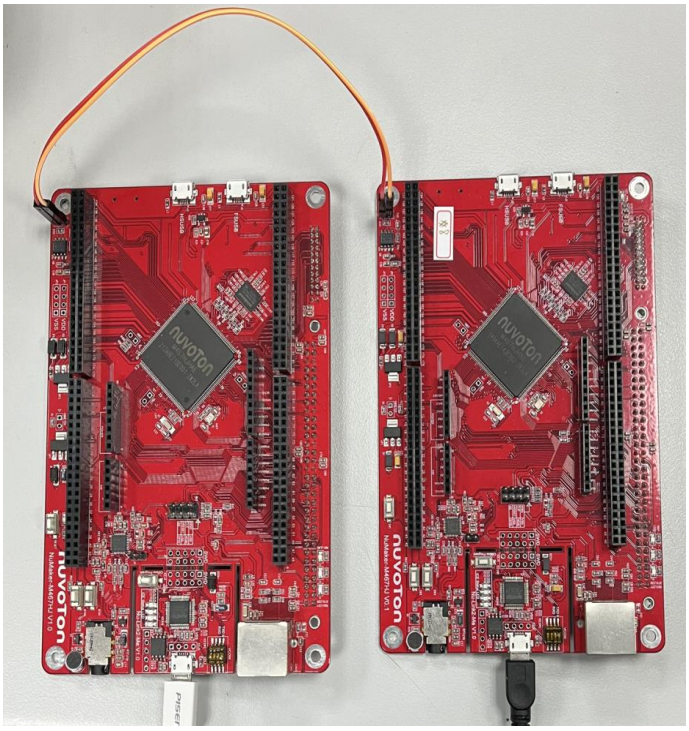

圖 3-1 兩個 NuMaker-M467HJ 板的 CANFD 介面連接

4. 目錄資訊

EC_M460_CANFD_Tx_Simply _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

圖 4-1 目錄資訊

5. 範例程式執行

1. 根據目錄資訊章節進入 ExampleCode 路徑中的 KEIL 資料夾，按兩下 *EC_M460_CANFD_Tx_Simply.uvprojx*。
2. 進入編譯模式介面
   - 編譯
   - 下載代碼至記憶體
   - 進入 / 離開模擬模式
3. 進入模擬模式介面
   - 執行代碼

6. 修訂紀錄

Date	Revision	Description
2023.02.20	1.00	初始發佈。

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.