M253 CAN 自動波特率調整

NuMicro® 32 位系列微控制器範例程式碼介紹

文件資訊

應用簡述	本範例代碼通過接收 CAN 總線上的數據封包，利用檢測 Protocol Error 中斷來動態調整 M253 CAN 的波特率，以此匹配 CAN 總線通訊的波特率。
BSP 版本	M253 Series CMSIS BSP V3.00.003
開發平台	NuMaker–M253LE V1.1 、ADM3050E CAN 收發器

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

本範例有兩個代碼分別為 CANFD_CAN_TxRx 和 CANFD_CAN_Auto_Adjust。其中，CANFD_CAN_TxRx 是一個為了方便測試 CANFD_CAN_Auto_Adjust 的 CAN 收發範例。

CANFD_CAN_Auto_Adjust 透過接收 CAN 總線上的數據封包，利用 Protocol Error 檢測進中斷方式來動態調整 M253 CAN 的波特率，直到能正常接收到 CAN 總線上的數據，以此匹配 CAN 總線通信波特率。當匹配上 CAN 總線的通信波特率後，會週期地往 CAN 總線發送數據包。程序中設置能檢測的波特率範圍為 50 Kbps ~ 1 Mbps，間隔為 50 Kbps。

1.1 原理

該應用中，CANFD_CAN_Auto_Adjust 例程會先將 M253 的 CAN 設置為 Bus Monitoring Mode 模式，接收 CAN 總線上的通信數據。在 Bus Monitoring Mode 模式，CAN 內核可以接收有效的數據幀和有效的遠程幀，但只在 CAN 總線上發送隱性位，且不會啟動發送。如果 CAN 內核需要發送顯性位(ACK 位、錯誤幀)，該位內部直接連接到 RX，所以儘管 CAN 總線保持在隱性狀態，CAN 內核可以檢測到這個顯性位。Bus Monitoring Mode 模式可以用來分析 CAN 總線的傳輸數據，且不會干擾總線上的數據傳輸。Bus Monitoring Mode 模式下，CAN 內核的 CAN_TX 和 CAN_RX 的連接如圖 1 - 1 所示。

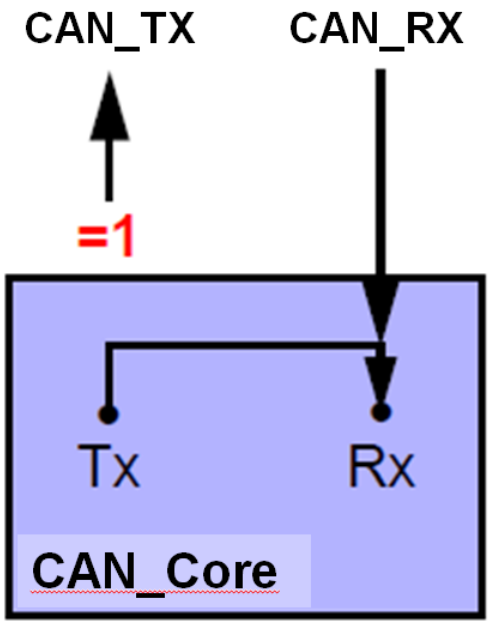

圖 1 - 1 Bus Monitoring Mode

1.2 執行結果

用兩塊 M253 板子，一塊燒錄 CANFD_CAN_TX_RX 範例，一塊燒錄 CANFD_CAN_Auto_Adjust_ 範例，然後將兩塊 M253 板子按圖 3-1 示意圖進行連接。接著打開 RS232 傳輸介面，並按提示發送對應的命令。

CANFD_CAN_TX_RX 範例，設定 CAN 測試波特率和測試封包的 ID Type 與封包訊息，執行結果如圖 1- 2 所示：

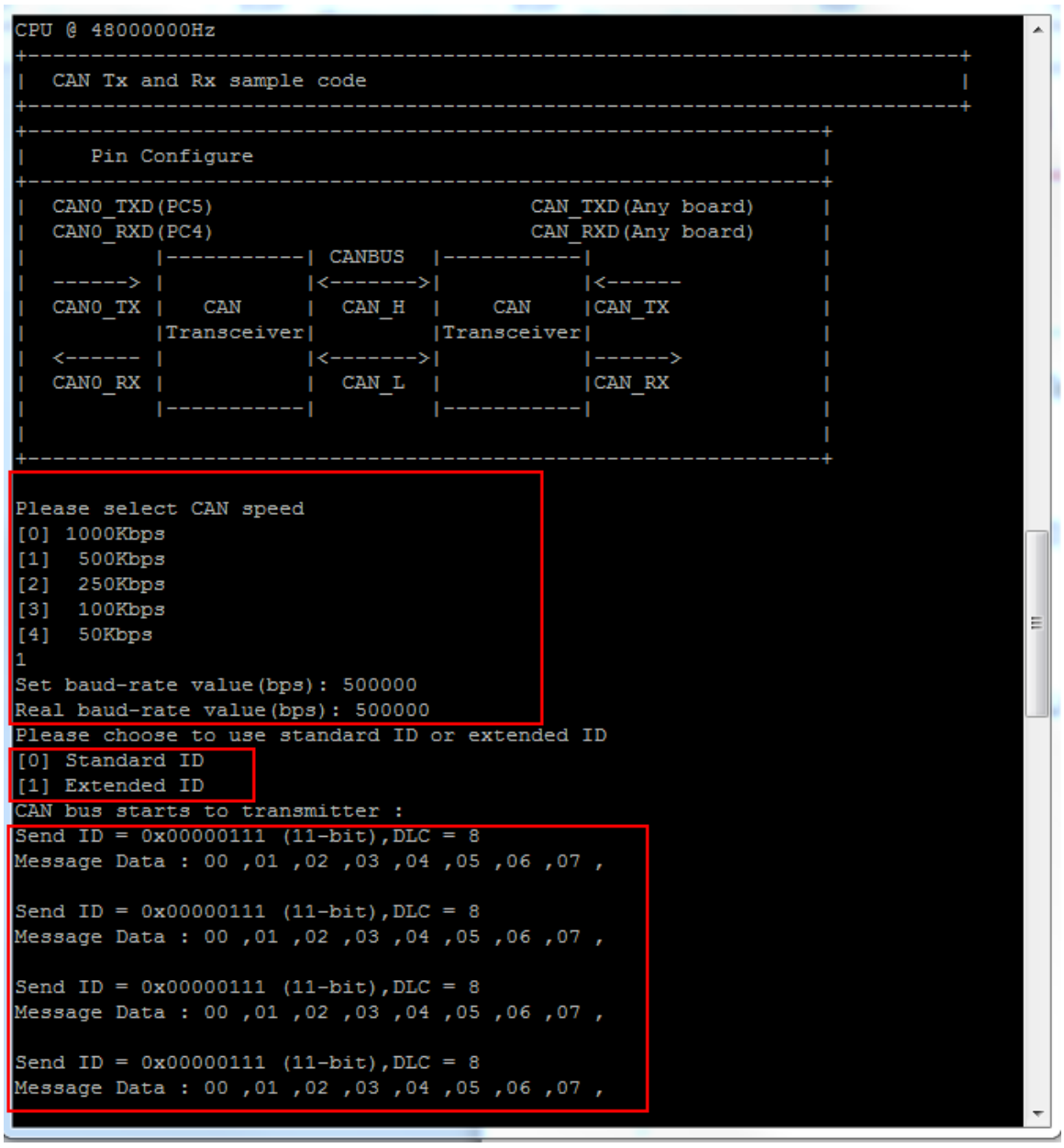

圖 1- 2 CANFD_CAN_TX_RX 執行結果

CANFD_CAN_Auto_Adjust 範例，動態調變 CAN 的波特率，直到能正常接收 CAN 總線數據，執行結果如圖 1- 3 所示：

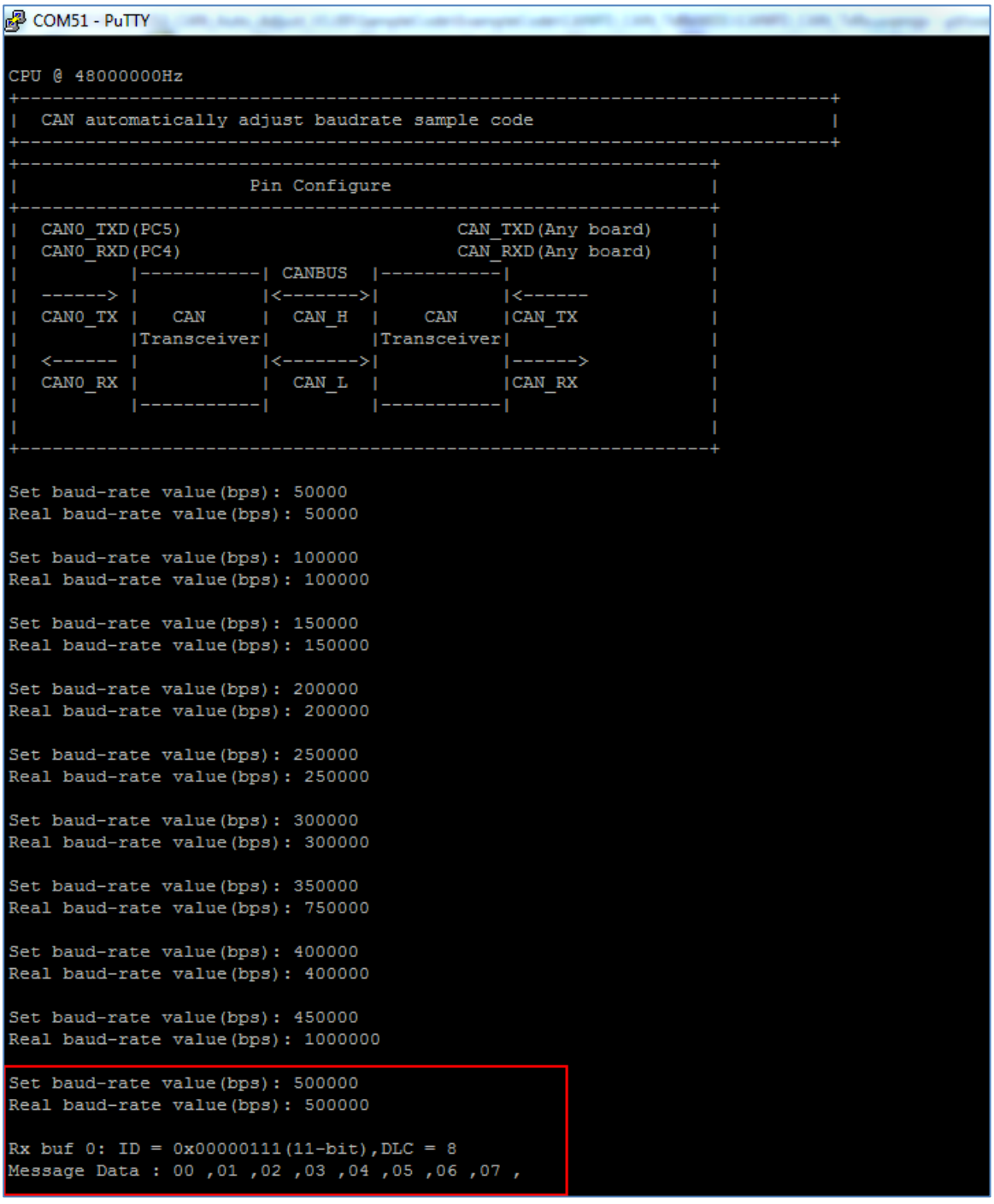

圖 1- 3 CANFD_CAN_Auto_Adjust 執行結果

CANFD_CAN_Auto_Adjust 範例，匹配 CAN 的波特率後，於 CANFD_CAN_TX_RX 之間的訊息傳送，執行結果如圖 1- 4 所示：

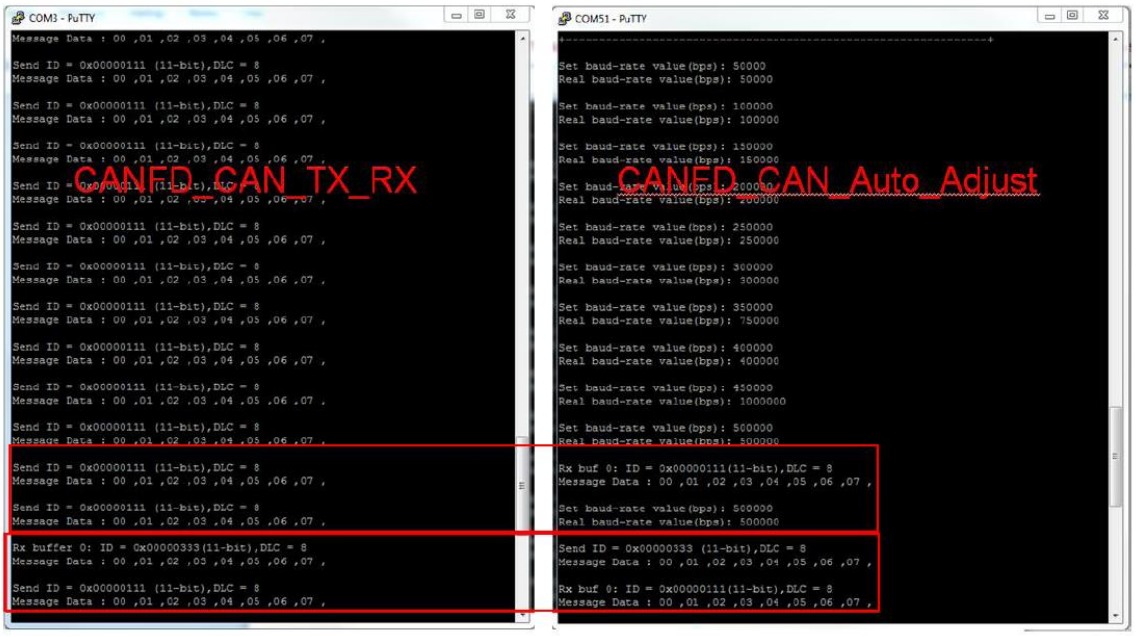

圖 1- 4 CANFD_CAN_Auto_Adjust 與 CANFD_CAN_TX_RX 通訊傳輸執行結果

2 程式碼介紹

以下介紹 CANFD_CAN_Auto_Adjust 範例的主要程式：

波特率檢測範圍及步進設置：

```
#define BAUDRATE_MIN        50000
#define BAUDRATE_MAX        1000000
#define BAUDRATE_OFFSET     50000
#define CAN_MON_DISABLE     0
#define CAN_MON_ENABLE      1
```

CAN 初始化功能：

```
void CAN_Init(uint32_t u32BitRate,uint8_t u8EnMonMode)
{
    CANFD_FD_T sCANFD_Config;
    CANFD_GetDefaultConfig(&sCANFD_Config, CANFD_OP_CAN_MODE);
    sCANFD_Config.sBtConfig.sNormBitRate.u32BitRate = u32BitRate;
    sCANFD_Config.sBtConfig.sDataBitRate.u32BitRate = 0;
    CANFD_Open(CANFD0, &sCANFD_Config);
    printf("Set baud-rate value(bps): %d\n",sCANFD_Config.sBtConfig.sNormBitRate.u32BitRate);
    printf("Real baud-rate value(bps): %d\n\n", Get_CAN_BitRate(CANFD0));
    /* receive 0x111 in CAN rx message buffer 0 by setting mask 0 */
    CANFD_SetSIDFltr(CANFD0, 0, CANFD_RX_BUFFER_STD(0x111, 0));


    /* receive 0x222 (29-bit id) in CAN rx message buffer 1 by setting mask 3 */
    CANFD_SetXIDFltr(CANFD0, 0, CANFD_RX_BUFFER_EXT_LOW(0x222, 1), CANFD_RX_BUFFER_EXT_HIGH(0x222,
1));
    /*Enable the Bus Monitoring Mode */
    if(u8EnMonMode == 1)
       CANFD0->CCCR |= CANFD_CCCR_MON_Msk;
    else
       CANFD0->CCCR &= ~CANFD_CCCR_MON_Msk;


/* Enable RX buffer new message ,Timeout,Protocol Error,Bus_Off interrupt using interrupt line 0. */
    CANFD_EnableInt(CANFD0, (CANFD_IE_TOOE_Msk | CANFD_IE_DRXE_Msk | CANFD_IE_BOE_Msk
                    |CANFD_IE_PEAE_Msk ), 0, 0, 0);
    /* CAN FD0 Run to Normal mode  */
    CANFD_RunToNormal(CANFD0, TRUE);
}
```

CAN 傳送訊息封包功能：

```
void CAN_SendMessage(CANFD_FD_MSG_T *psTxMsg, E_CANFD_ID_TYPE eIdType, uint32_t u32Id, uint8_t
u8Len)
{
    uint8_t u8Cnt;
    psTxMsg->u32Id = u32Id;
    psTxMsg->eIdType = eIdType;
    psTxMsg->eFrmType = eCANFD_DATA_FRM;
    psTxMsg->bBitRateSwitch = 0;
    psTxMsg->u32DLC = u8Len;

    /* use message buffer 0 */
    if (eIdType == eCANFD_SID)
        printf("Send ID = 0x%08x (11-bit),DLC = %d\n", psTxMsg->u32Id,psTxMsg->u32DLC);
    else
        printf("Send ID = 0x%08x (29-bit),DLC = %d\n", psTxMsg->u32Id,psTxMsg->u32DLC);

    printf("Message Data : ");
    for (u8Cnt = 0; u8Cnt < psTxMsg->u32DLC; u8Cnt++)
    {
      psTxMsg->au8Data[u8Cnt] = u8Cnt;
      printf("%02X ,", psTxMsg->au8Data[u8Cnt]);
    }
     printf("\n\n");
    if (CANFD_TransmitTxMsg(CANFD0, 0, psTxMsg) != 1)
    {
        printf("Failed to transmit message\n");
    }
}
```

動態調變 CAN 的波特率，直到能正常接收 CAN 總線數據，然後退出 Bus Monitoring 模式，之後開始傳送訊息封包：

```
    while(1)
    {
        /* Start automatic bit rate adjustment */
        if(g_u8AutoAdjustOkFlag == 0)
        {
            /* Protocol Error in Arbitration Phase */
            if(g_u8CanBitRateAdjust == 20)
            {
```

```
                u32CanBitRate = u32CanBitRate + BAUDRATE_OFFSET;
                if(u32CanBitRate > BAUDRATE_MAX)
                {
                    u32CanBitRate = BAUDRATE_MIN;
                }
                /* CAN FD0 run to INIT mode  */
                CANFD_RunToNormal(CANFD0, FALSE);
                /* Can bus bit rate adjustment */
                CAN_Init(u32CanBitRate,CAN_MON_ENABLE);
                g_u8CanBitRateAdjust = 0;
            }
           if(g_u8RxBufRcvOk == 1)
           {
                g_u8AutoAdjustOkFlag = 1;
                /* CAN FD0 run to INIT mode  */
                CANFD_RunToNormal(CANFD0, FALSE);
                CAN_Init(u32CanBitRate,CAN_MON_DISABLE);
           }

       }
       /*Start data transmission communication*/
       if(g_u8AutoAdjustOkFlag == 1)
       {
         CAN_SendMessage(&sTxMsgFrame,eCANFD_SID,0x333,8);
         CLK_SysTickLongDelay(1000000);
         CAN_SendMessage(&sTxMsgFrame,eCANFD_XID,0x444,8);
         CLK_SysTickLongDelay(1000000);
       }
   }
```

3 軟體與硬體需求

3.1 軟體需求

- BSP 版本
  - M253 Series CMSIS BSP V3.00.003
- IDE 版本
  - Keil uVersion 5.26

3.2 硬體需求

- 電路元件
  - NuMaker-M253LE V1.1
  - ADM3050E CAN 收發器
- 示意圖

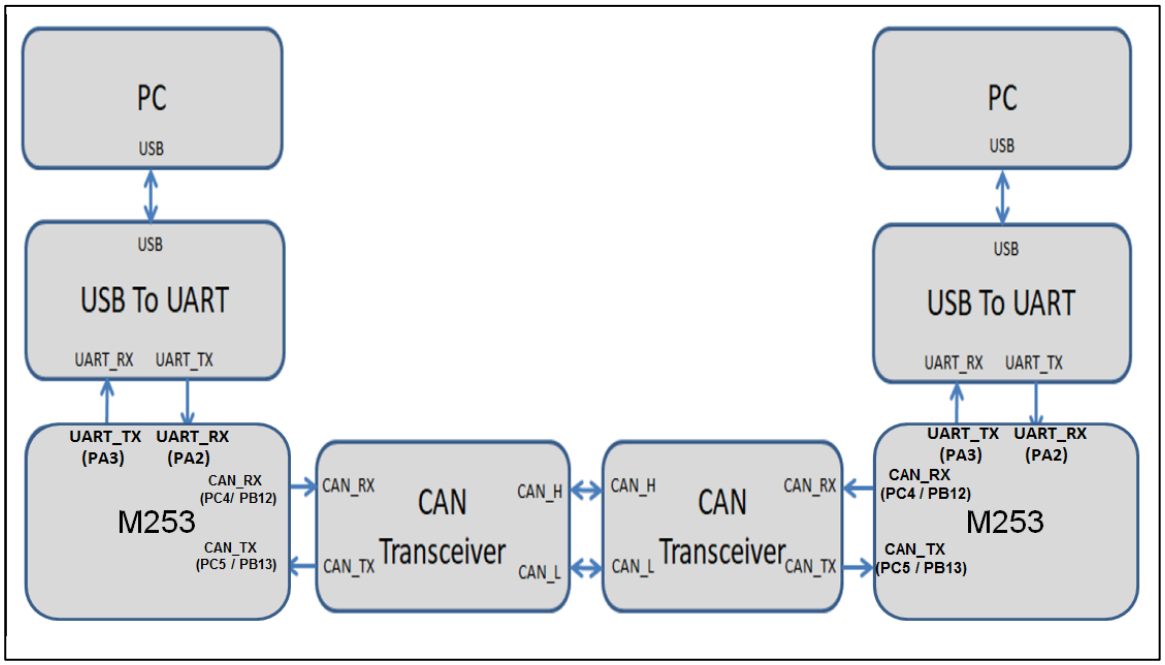

圖 3-1 硬體線路連接示意圖

本範例測試時，使用的 CAN 收發器是 ADM3050E，其與 M253 的連接示意如圖 3-2 所示。

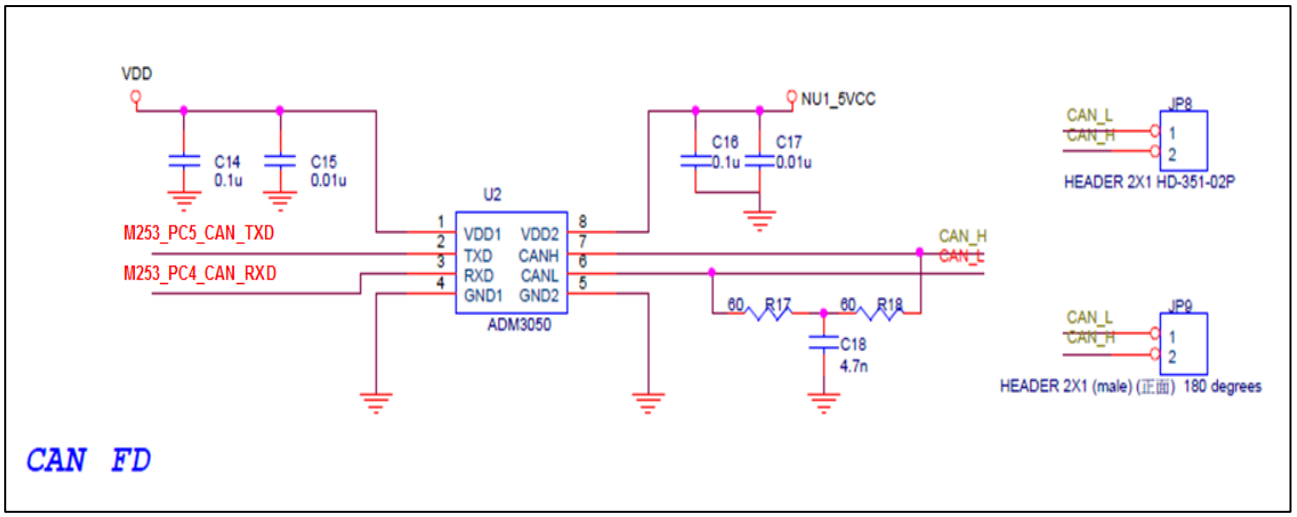

圖 3-2 M253 與 CAN 收發器連接示意圖

若使用其它 CAN 收發器，連接可能會有差異，詳情可參考對應的 CAN 收發器的數據手冊。

4 目錄資訊

EC_M253_CAN_Auto_Adjust_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，雙擊 CANFD_CAN_Auto_Adjust.uvprojx 和 CANFD_CAN_TxRx.uvprojx。
2. 進入編譯模式介面
   a. 編譯
   b. 下載代碼至記憶體
   c. 進入 / 離開除錯模式
3. 進入除錯模式介面
   a. 執行代碼

6 修訂紀錄

Date	Revision	Description
2021.11.29	1.00	1. 初始發布.

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.