M253 CAN 自动波特率调整

NuMicro® 32 位系列微控制器范例程序代码介绍

文件信息

应用简述	本范例代码通过接收 CAN 总在线的数据封包，利用检测 Protocol Error 中断来动态调整 M253 CAN 的波特率，以此匹配 CAN 总线通讯的波特率。
BSP 版本	M253 Series CMSIS BSP V3.00.003
开发平台	NuMaker–M253LE V1.1 、ADM3050E CAN 收发器

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

本范例有两个代码分别为 CANFD_CAN_TxRx 和 CANFD_CAN_Auto_Adjust。其中，CANFD_CAN_TxRx 是一个为了方便测试 CANFD_CAN_Auto_Adjust 的 CAN 收发范例。

CANFD_CAN_Auto_Adjust 透过接收 CAN 总在线的数据封包，利用 Protocol Error 检测进中断方式来动态调整 M253 CAN 的波特率，直到能正常接收到 CAN 总在线的数据，以此匹配 CAN 总线通信波特率。当匹配上 CAN 总线的通信波特率后，会周期地往 CAN 总线发送数据包。程序中设置能检测的波特率范围为 50 Kbps ~ 1 Mbps，间隔为 50 Kbps。

1.1 原理

该应用中，CANFD_CAN_Auto_Adjust 例程会先将 M253 的 CAN 设置为 Bus Monitoring Mode 模式，接收 CAN 总在线的通信数据。在 Bus Monitoring Mode 模式，CAN 内核可以接收有效的数据帧和有效的远程帧，但只在 CAN 总在线发送隐性位，且不会启动发送。如果 CAN 内核需要发送显性位(ACK 位、错误帧)，该位内部直接连接到 RX，所以尽管 CAN 总线保持在隐性状态，CAN 内核可以检测到这个显性位。Bus Monitoring Mode 模式可以用来分析 CAN 总线的传输数据，且不会干扰总在线的数据传输。Bus Monitoring Mode 模式下，CAN 内核的 CAN_TX 和 CAN_RX 的连接如图 1 - 1 所示。

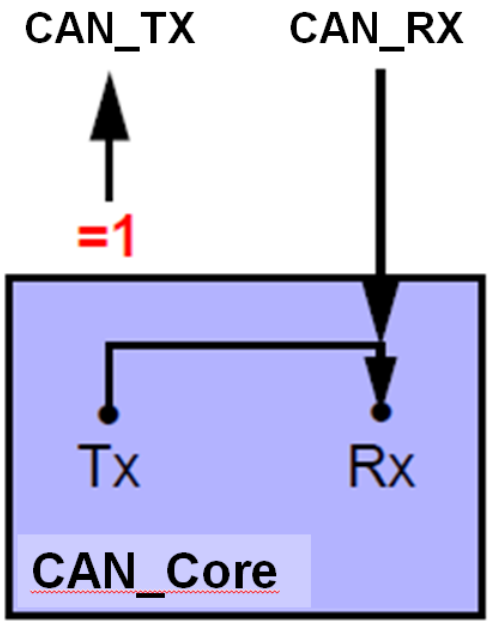

图 1 - 1 Bus Monitoring Mode

1.2 执行结果

用两块 M253 板子，一块刻录 CANFD_CAN_TX_RX 范例，一块刻录 CANFD_CAN_Auto_Adjust_ 范例，然后将两块 M253 板子按图 3-1 示意图进行连接。接着打开 RS232 传输接口，并按提示发送对应的命令。

CANFD_CAN_TX_RX 范例，设定 CAN 测试波特率和测试封包的 ID Type 与封包讯息，执行结果如图 1- 2 所示：

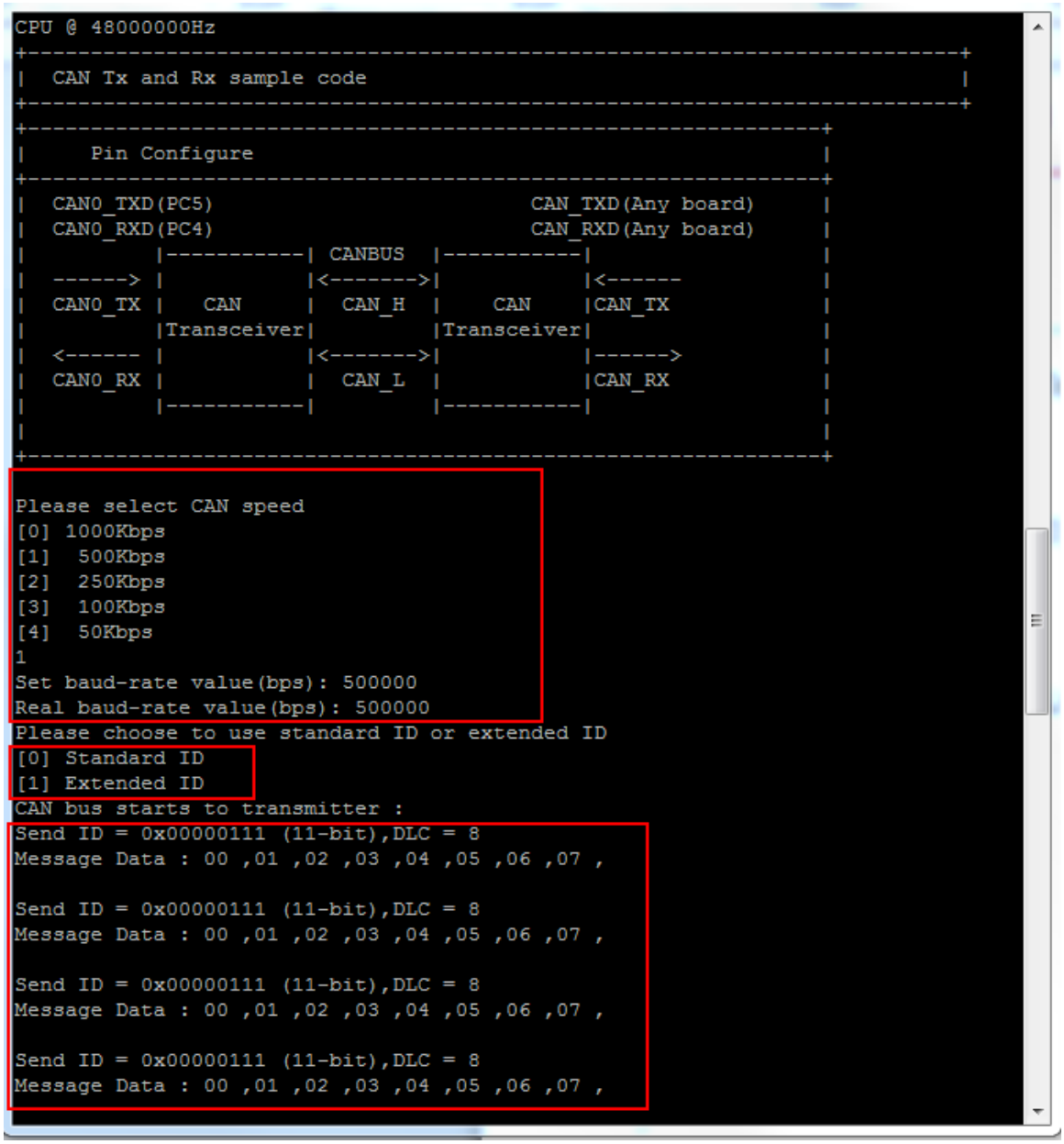

图 1- 2 CANFD_CAN_TX_RX 执行结果

CANFD_CAN_Auto_Adjust 范例，动态调变 CAN 的波特率，直到能正常接收 CAN 总线数据，执行结果如图 1- 3 所示：

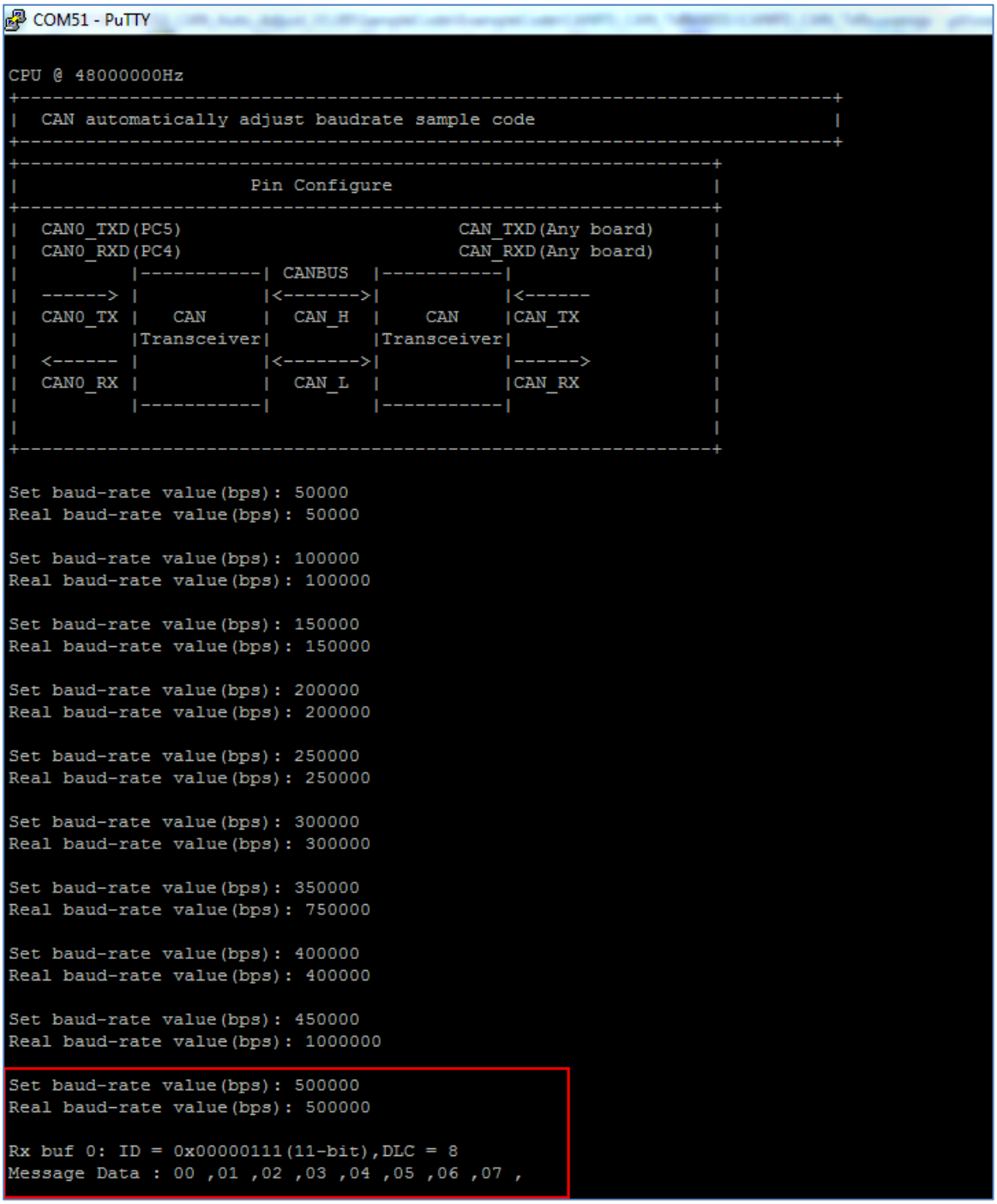

图 1- 3 CANFD_CAN_Auto_Adjust 执行结果

CANFD_CAN_Auto_Adjust 范例，匹配 CAN 的波特率后，于 CANFD_CAN_TX_RX 之间的讯息传送，执行结果如图 1- 4 所示：

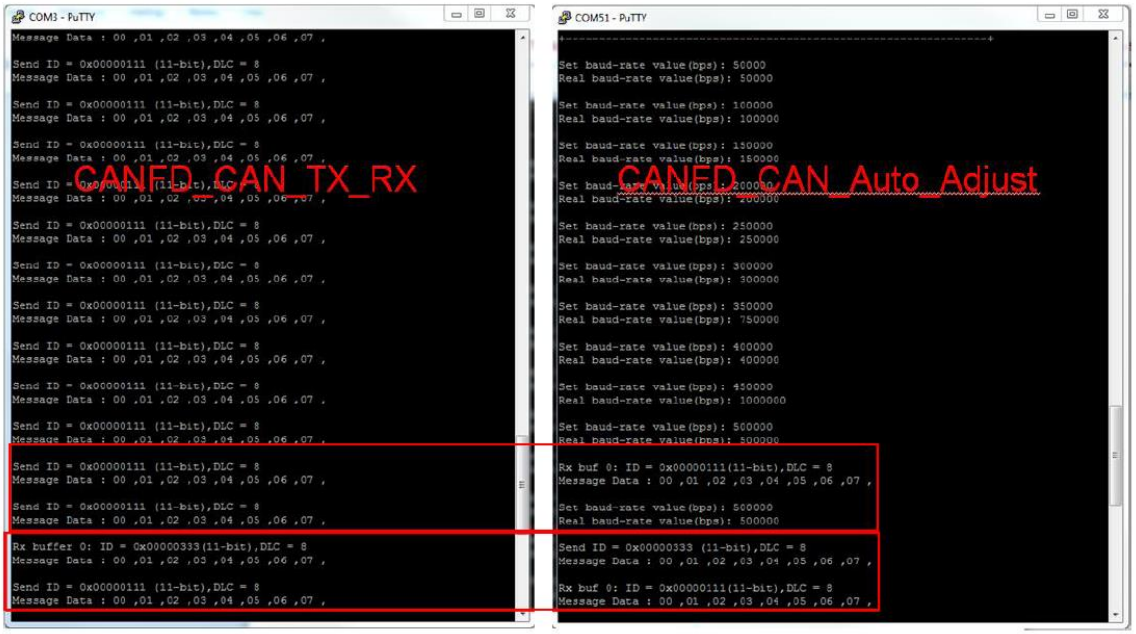

图 1- 4 CANFD_CAN_Auto_Adjust 与 CANFD_CAN_TX_RX 通讯传输执行结果

2 程序代码介绍

以下介绍 CANFD_CAN_Auto_Adjust 范例的主要程序：

波特率检测范围及步进设置：

```
#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 传送讯息封包功能：

```
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");
    }
}
```

动态调变 CAN 的波特率，直到能正常接收 CAN 总线数据，然后退出 Bus Monitoring 模式，之后开始传送讯息封包：

```
    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 软件与硬件需求

3.1 软件需求

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

3.2 硬件需求

- 电路组件
  - NuMaker-M253LE V1.1
  - ADM3050E CAN 收发器
- 示意图

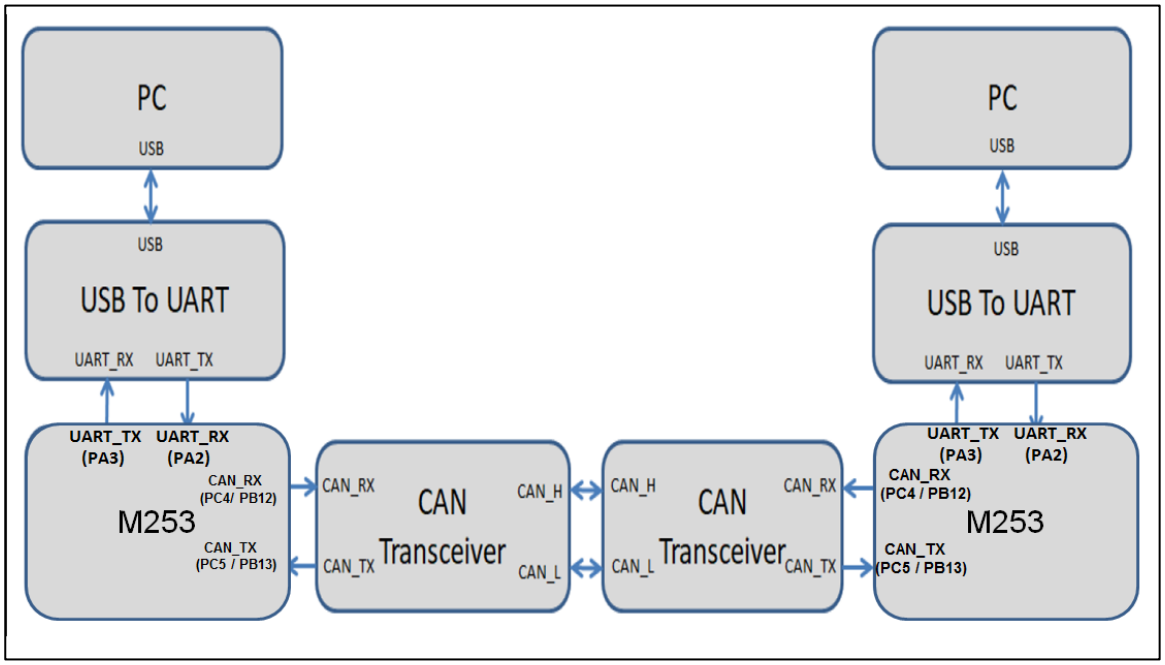

图 3-1 硬件线路连接示意图

本范例测试时，使用的 CAN 收发器是 ADM3050E，其与 M253 的连接示意如图 3-2 所示。

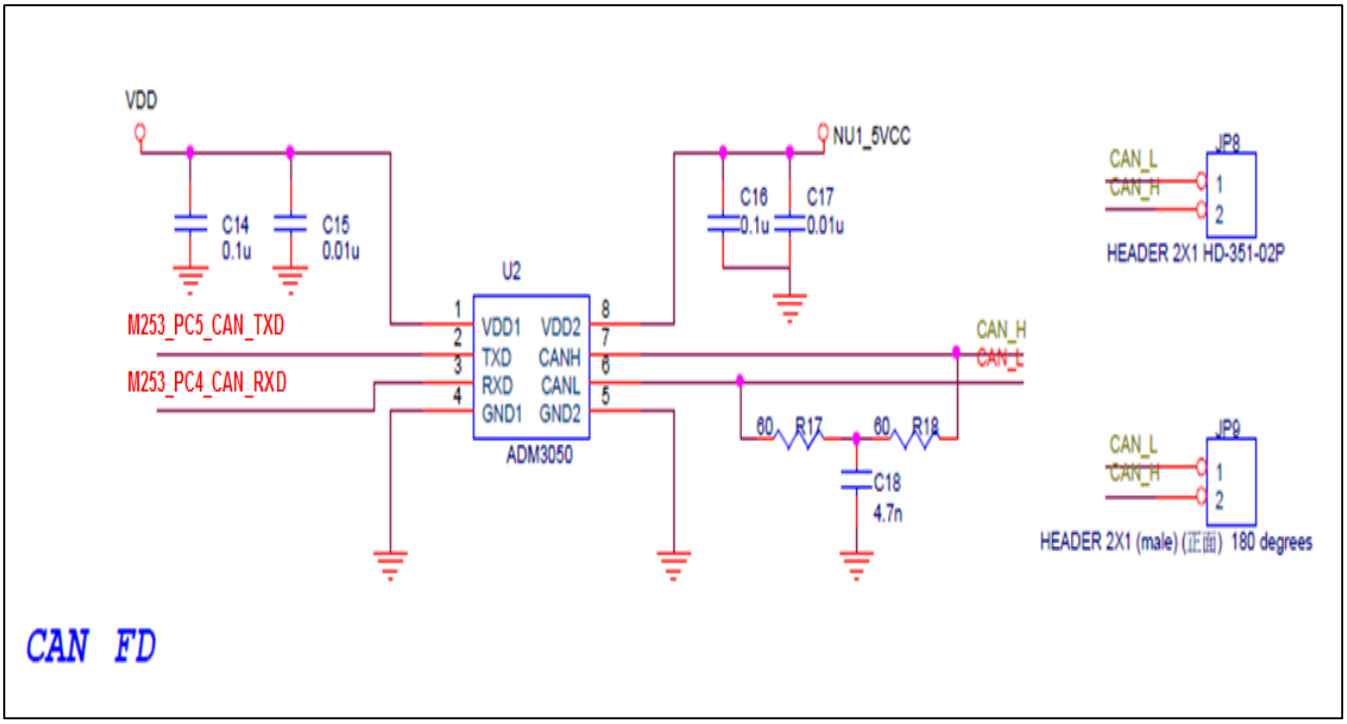

图 3-2 M253 与 CAN 收发器连接示意图

若使用其它 CAN 收发器，连接可能会有差异，详情可参考对应的 CAN 收发器的数据手册。

4 目录信息

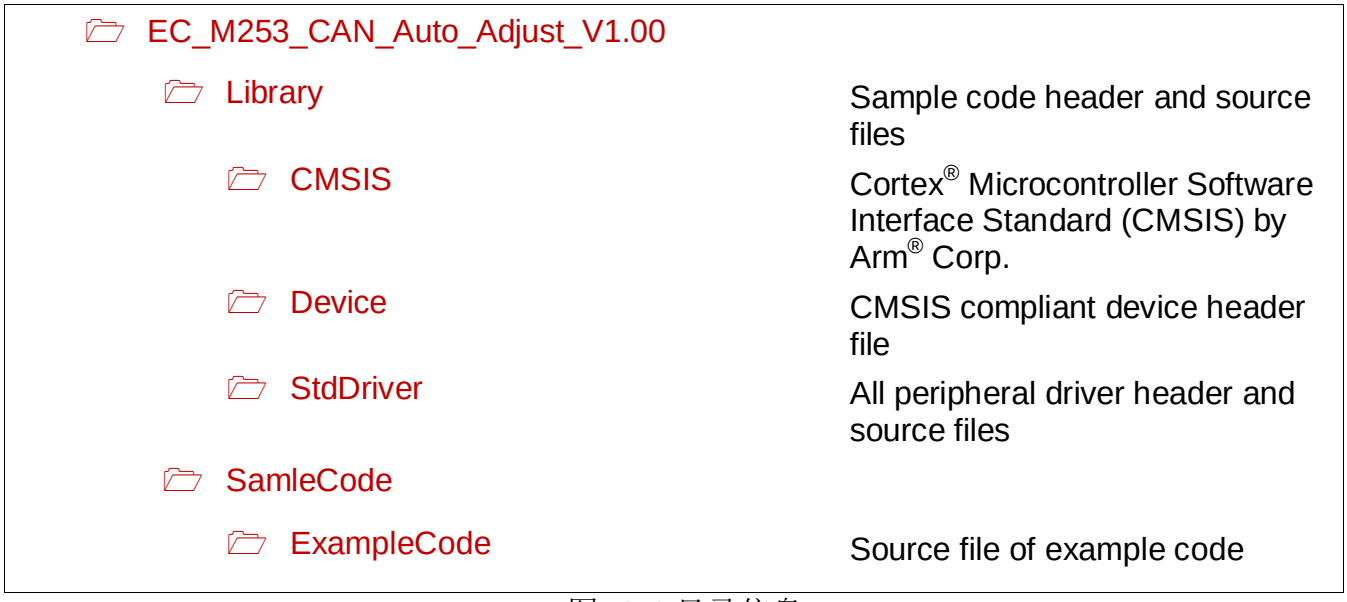

图 4-1 目录信息

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 CANFD_CAN_Auto_Adjust.uvprojx 和 CANFD_CAN_TxRx.uvprojx。
2. 进入编译模式接口
   a. 编译
   b. 下载代码至内存
   c. 进入 / 离开除错模式
3. 进入除错模式接口
   a. 执行代码

6 修订纪录

Date	Revision	Description
2021.11.29	1.00	1. 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.