使用M251驱动MCP2515 CAN总线模块

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 M251 透过 SPI 驱动 MCP2515 CAN 总线模块，并透过带有 CAN FD 接口的 M253 作为接收端，演示 CAN 封包的发送与接收。
BSP 版本	M251_M252_M254_M256_M258_Series_BSP_CMSIS_V3.00.006 M253_Series_BSP_CMSIS_V3.00.008
开发平台	NuMaker-M251SD V1.1 NuMaker-M253LEB 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

基于在未含有 CAN FD 通信接口的 MCU 上实现 CAN 传输的应用，可以使用独立的 CAN 控制器以其他通信接口连接 MCU 以达成 CAN 通信。本范例程序将使用广泛使用的 MCP2515 CAN 总线模块，以 SPI 接口连接 NuMaker-M251SD，并使用 NuMaker-M253LEB 为 CAN Slave，以完成 CAN 的封包发送与接收。

1.1 原理

MCP2515 CAN 总线模块，如图 1-1 所示，其主要组件包括以 SPI 作为通信接口的 CAN 控制器 - MCP2515，此 MCP2515 IC 用于处理 CAN 总线逻辑。另一主要组件则是 CAN Transceiver – TJA1050 来处理数字信号转换为差分信号。此模块的电路图请参考链接。

此模块的供应电压，如图 1-2 与图 1-3 撷取自 MCP2515 与 TJA1050 Datasheet，由于受限 TJA1050 的最小电压，因此需至少供应 4.75V 的电压至模块上的 V_{CC}。而 SPI I/O High 的条件，如图 1-2 所示，需查看 MCP2515 的 V_{IH} 规格，V_{IH} 需至少 0.7*V_{DD}，才足够支持 SPI I/O 至 High-Level Input Voltage。

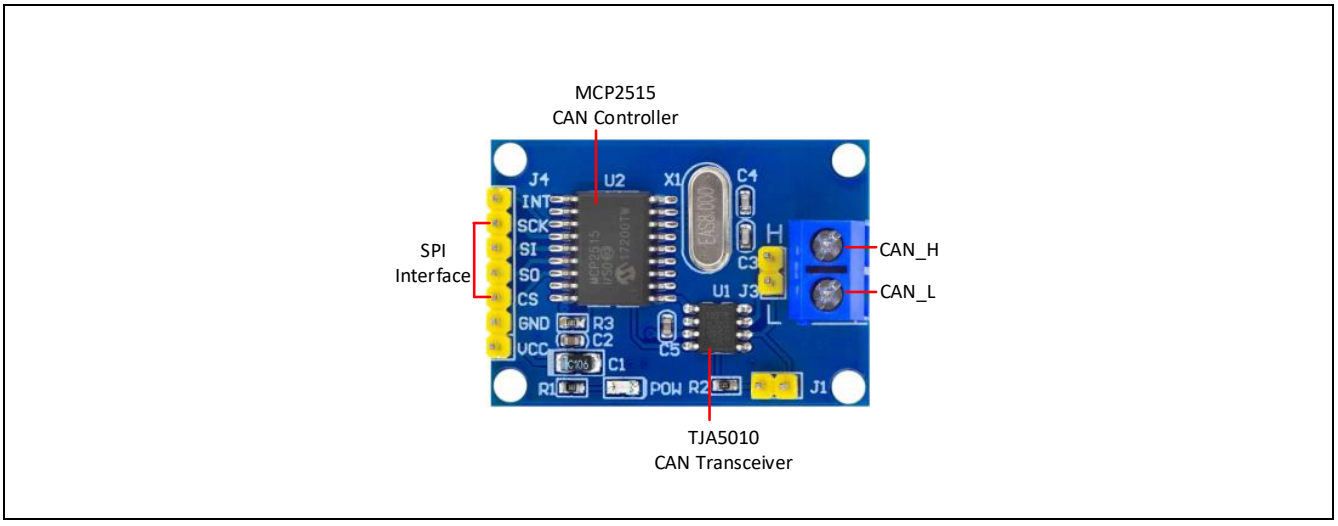

图 1-1 MCP2515 CAN 模块

MCP2515

TABLE 13-1: DC CHARACTERISTICS

DC Characteristics			V_{DD} = 2.7V to 5.5V		Industrial (I): T_{AMB} = -40°C to +85°C Extended (E): T_{AMB} = -40°C to +125°C	
Param. No.	Sym	Characteristic	Min	Max	Units	Conditions
	V_{DD}	Supply Voltage	2.7	5.5	V	
	V_{RET}	Register Retention Voltage	2.4	—	V	
	V_{IH}	**High-Level Input Voltage**				
		RXCAN Pin	2	V_{DD} + 1	V	
		SCK, $\overline{CS}$, SI, $\overline{TXnRTS}$ Pins	0.7 V_{DD}	V_{DD} + 1	V	
		OSC1 Pin	0.85 V_{DD}	V_{DD}	V	
		$\overline{RESET}$ Pin	0.85 V_{DD}	V_{DD}	V	
	V_{IL}	**Low-Level Input Voltage**				
		RXCAN, $\overline{TXnRTS}$ Pins	-0.3	0.15 * V_{DD}	V	
		SCK, $\overline{CS}$, SI Pins	-0.3	0.4 * V_{DD}	V	
		OSC1 Pin	V_{SS}	0.3 * V_{DD}	V	
		$\overline{RESET}$ Pin	V_{SS}	0.15 * V_{DD}	V	
	V_{OL}	**Low-Level Output Voltage**				
		TXCAN Pin	—	0.6	V	I_{OL} = +6.0 mA, V_{DD} = 4.5V
		$\overline{RXnBF}$ Pin	—	0.6	V	I_{OL} = +8.5 mA, V_{DD} = 4.5V
		SO, CLKOUT Pins	—	0.6	V	I_{OL} = +2.1 mA, V_{DD} = 4.5V
		$\overline{INT}$ Pin	—	0.6	V	I_{OL} = +1.6 mA, V_{DD} = 4.5V
	V_{OH}	**High-Level Output Voltage**			V	
		TXCAN, $\overline{RXnBF}$ Pins	V_{DD} – 0.7	—	V	I_{OH} = -3.0 mA, V_{DD} = 4.5V
		SO, CLKOUT Pins	V_{DD} – 0.5	—	V	I_{OH} = -400 µA, V_{DD} = 4.5V
		$\overline{INT}$ Pin	V_{DD} – 0.7	—	V	I_{OH} = -1.0 mA, V_{DD} = 4.5V

图 1-2 MCP2515 Datasheet – DC Characteristics

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V_{CC}	supply voltage		4.75	5.25	V
V_{CANH}	DC voltage at pin CANH	0 < V_{CC} < 5.25 V; no time limit	–27	+40	V
V_{CANL}	DC voltage at pin CANL	0 < V_{CC} < 5.25 V; no time limit	–27	+40	V
$V_{I(dif)(bus)}$	differential bus input voltage	dominant	1.5	3	V
$t_{PD(TXD-RXD)}$	propagation delay TXD to RXD	V_S = 0 V; see Fig.7	–	250	ns
T_{vj}	virtual junction temperature		–40	+150	°C

图 1-3 TJA1050 Datasheet – DC Characteristics

关于控制 MCP2515 的方式，则是以 SPI 接口连接 MCU，所有操作皆透过指令控制/读取寄存器，寄存器总表可于 MCP2515 Datasheet 中查询。

1.2 执行结果

本范例代码执行编译和烧录的结果，分别对 NuMaker-M251SD 与 NuMaker-M253LEB 指定为 Slave 或 Master，便可以看到成功传送与成功接收指定封包，输出后如图 1-4 所示。

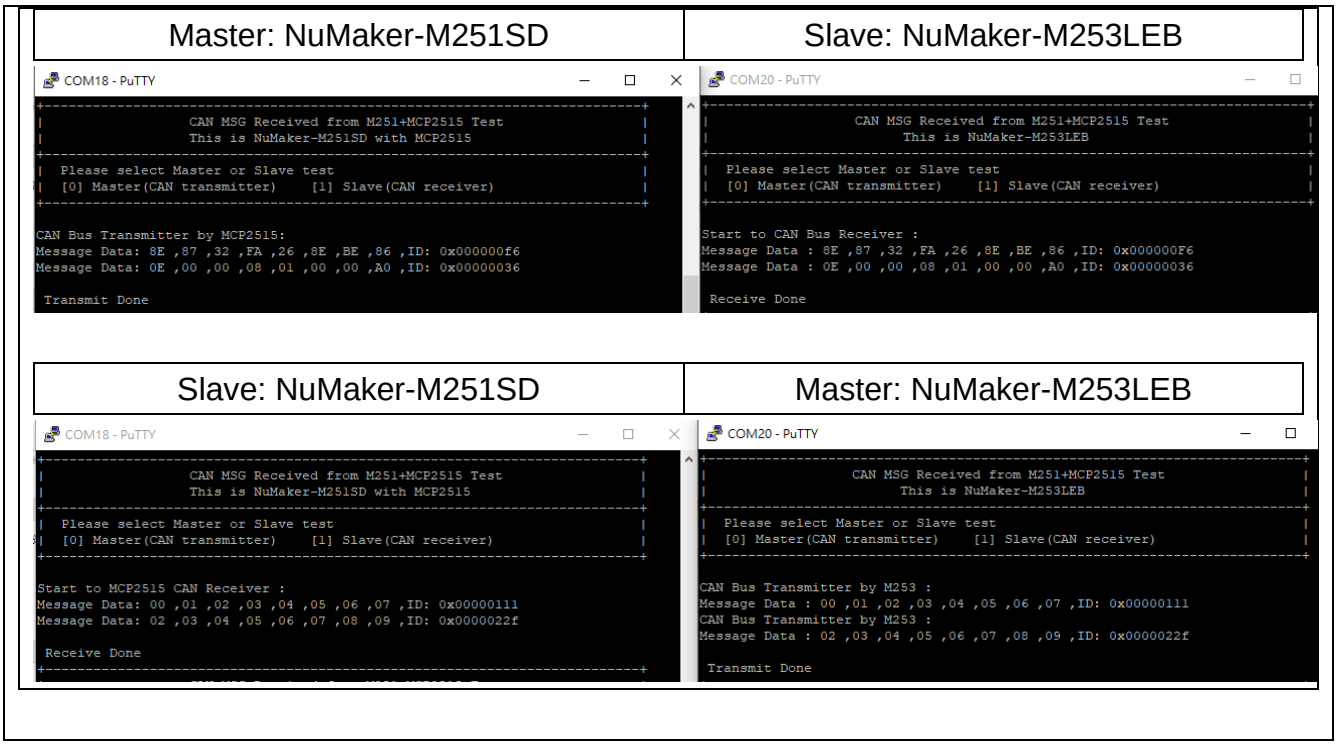

图 1-4 执行编译和烧录结果

2. 代码介绍

M251 为处理 CAN 通信与发送，主程序由 *main.c* 进入，并在 *MCP2515.c* 中设置对应的 SPI register，将 MCP2515 初始化、设置速度与模式。

```
ERROR MCP2515_reset(void)
{
    startSPI();
    SPI_transfer(INSTRUCTION_RESET);

    endSPI();
    TIMER_Delay(TIMER0,500); //delay 500ms
    uint8_t zeros[14];
    memset(zeros, 0, sizeof(zeros));
    MCP2515_setRegisters(MCP_TXB0CTRL, zeros, 14);
    MCP2515_setRegisters(MCP_TXB1CTRL, zeros, 14);
    MCP2515_setRegisters(MCP_TXB2CTRL, zeros, 14);

    MCP2515_setRegister(MCP_RXB0CTRL, 0);
    MCP2515_setRegister(MCP_RXB1CTRL, 0);

    MCP2515_setRegister(MCP_CANINTE, CANINTF_RX0IF | CANINTF_RX1IF | CANINTF_ERRIF |
CANINTF_MERRF);

    // receives all valid messages using either Standard or Extended Identifiers that
    // meet filter criteria. RXF0 is applied for RXB0, RXF1 is applied for RXB1
    MCP2515_modifyRegister(MCP_RXB0CTRL,
                  RXBnCTRL_RXM_MASK | RXB0CTRL_BUKT | RXB0CTRL_FILHIT_MASK,
                  RXBnCTRL_RXM_STDEXT | RXB0CTRL_BUKT | RXB0CTRL_FILHIT);
    MCP2515_modifyRegister(MCP_RXB1CTRL,
                  RXBnCTRL_RXM_MASK | RXB1CTRL_FILHIT_MASK,
                  RXBnCTRL_RXM_STDEXT | RXB1CTRL_FILHIT);

    RXF filters[] = {RXF0, RXF1, RXF2, RXF3, RXF4, RXF5};
    for (int i=0; i<6; i++) {
        bool ext = (i == 1);
        ERROR result = MCP2515_setFilter(filters[i], ext, 0);
        if (result != ERROR_OK) {
            return result;
        }
    }

    MASK masks[] = {MASK0, MASK1};
    for (int i=0; i<2; i++) {
        ERROR result = setFilterMask(masks[i], true, 0);
        if (result != ERROR_OK) {
            return result;
        }
    }

    return ERROR_OK;
}
```

```
ERROR MCP2515_setBitrate(const CAN_SPEED canSpeed)
{
    return MCP2515_setBitrateWithClock(canSpeed, MCP_8MHZ);
}

ERROR MCP2515_setNormalMode()
{
    return MCP2515_setMode(CANCTRL_REQOP_NORMAL);
}
```

MCP2515 设置完成后，进入 MCP2515_CAN_TxRxTest 函数中，做发送或接收的测试，而发送与接收的行为，分别在 *MCP2515.c* 中的 API 完成。

```
void MCP2515_CAN_TxRxTest(void)
{
    uint8_t u8Item;
    uint8_t u8Cnt;
    uint8_t u8RxTestNum = 0;

    printf("+-------------------------------------------------------------------------+\n");
    printf("|                  CAN MSG Received from M251+MCP2515 Test                |\n");
    printf("|                  This is NuMaker-M251SD with MCP2515                    |\n");
    printf("+-------------------------------------------------------------------------+\n");
    printf("|  Please select Master or Slave test                                     |\n");
    printf("|  [0] Master(CAN transmitter)    [1] Slave(CAN receiver)                 |\n");
    printf("+-------------------------------------------------------------------------+\n\n");
    u8Item = getchar();
    u8RxTestNum = 0;
    bool result ;
    if (u8Item == '0')
    {
      printf("CAN Bus Transmitter by MCP2515:\n");
      MCP2515_sendMessage(&canMsg1);
      MCP2515_sendMessage(&canMsg2);
      TIMER_Delay(TIMER0, 100);//delay 100ms
      if((MCP2515_sendMessage(&canMsg1) || MCP2515_sendMessage(&canMsg2)) == 0) {
            printf("Message Data: ");
            for(u8Cnt =0; u8Cnt<canMsg1.can_dlc; u8Cnt++) {
                printf("%02X ,",canMsg1.data[u8Cnt]);
            }
            printf("ID: 0x%08lx \n",canMsg1.can_id);
            printf("Message Data: ");
            for(u8Cnt =0; u8Cnt<canMsg2.can_dlc; u8Cnt++) {
                printf("%02X ,",canMsg2.data[u8Cnt]);
            }
            printf("ID: 0x%08lx \n",canMsg2.can_id);

            printf("\n Transmit Done\n");
        } else {
            printf("\n Transmit Fail\n");
        }
    } else {
        printf("Start to MCP2515 CAN Receiver :\n");
        do {
            if(MCP2515_readMessage(&canMsg0) == ERROR_OK) {
```

```
                printf("Message Data: ");
                for(u8Cnt=0; u8Cnt<canMsg0.can_dlc; u8Cnt++) {
                    printf("%02X ,",canMsg0.data[u8Cnt]);
                }
                printf("ID: 0x%08lx \n",canMsg0.can_id);
                u8RxTestNum++;
            }
        } while(u8RxTestNum < 2);
        printf("\n Receive Done\n");
    }
}

int main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();
    /* Init System, IP clock and multi-function I/O. */
    SYS_Init();
    /* Lock protected registers */
    SYS_LockReg();

    /* Init SPI */
    SPI_Init();

    MCP2515_reset();
    MCP2515_setBitrate(CAN_125KBPS);
    MCP2515_setNormalMode();

    canMsg1.can_id  = 0x0F6;
    canMsg1.can_dlc = 8;
    canMsg1.data[0] = 0x8E;
    canMsg1.data[1] = 0x87;
    canMsg1.data[2] = 0x32;
    canMsg1.data[3] = 0xFA;
    canMsg1.data[4] = 0x26;
    canMsg1.data[5] = 0x8E;
    canMsg1.data[6] = 0xBE;
    canMsg1.data[7] = 0x86;

    canMsg2.can_id  = 0x036;
    canMsg2.can_dlc = 8;
    canMsg2.data[0] = 0x0E;
    canMsg2.data[1] = 0x00;
    canMsg2.data[2] = 0x00;
    canMsg2.data[3] = 0x08;
    canMsg2.data[4] = 0x01;
    canMsg2.data[5] = 0x00;
    canMsg2.data[6] = 0x00;
    canMsg2.data[7] = 0xA0;

    MCP2515_CAN_TxRxTest();

    while(1);
}
```

M253 方面，主程序由 *main.c* 进入后，在 CAN_TxRxTest 函数中完成发送或接收指定的 CAN 封包。

```
void CAN_TxRxTest(void)
{
    uint8_t u8Item;
    uint8_t u8Cnt;
    CANFD_FD_MSG_T      sRxFrame;
    CANFD_FD_MSG_T      sTxMsgFrame;

    /* CAN interface initialization*/
    CAN_Init();

    printf("+-----------------------------------------------------------+\n");
    printf("|              CAN MSG Received from M251+MCP2515 Test      |\n");
    printf("|                  This is NuMaker-M253LEB                  |\n");
    printf("+-----------------------------------------------------------+\n");
    printf("|  Please select Master or Slave test                       |\n");
    printf("|  [0] Master(CAN transmitter)    [1] Slave(CAN receiver)   |\n");
    printf("+-----------------------------------------------------------+\n\n");
    u8Item = getchar();

    uint8_t u8ErrFlag = 0;
    uint8_t u8RxTestNum = 0;

    if(u8Item == '0') {
        uint8_t u8TxTestNum = 0;

        /* Send 2 messages with different ID and data size */
        for(u8TxTestNum = 0; u8TxTestNum < 2 ; u8TxTestNum++) {
            printf("CAN Bus Transmitter by M253 :\n");

            /* Set the ID Number */
            if(u8TxTestNum == 0)      sTxMsgFrame.u32Id = 0x111;
            else if(u8TxTestNum == 1) sTxMsgFrame.u32Id = 0x22F;
            /*Set the ID type*/
            sTxMsgFrame.eIdType = eCANFD_SID;
            /*Set the frame type*/
            sTxMsgFrame.eFrmType = eCANFD_DATA_FRM;
            /*Set the bitrate switch */
            sTxMsgFrame.bBitRateSwitch = 0;
            /*Set the data lenght */
            sTxMsgFrame.u32DLC = 8;

            for(u8Cnt = 0; u8Cnt < sTxMsgFrame.u32DLC; u8Cnt++) sTxMsgFrame.au8Data[u8Cnt]
= u8Cnt + u8TxTestNum*2;
            printf("Message Data : ");
            for(u8Cnt = 0; u8Cnt < sTxMsgFrame.u32DLC; u8Cnt++) {
                printf("%02X ,", sTxMsgFrame.au8Data[u8Cnt]);
            }
            printf("ID: 0x%08x \n", sTxMsgFrame.u32Id);

            /* use message buffer 0 */
            if(CANFD_TransmitTxMsg(CANFD0, 0, &sTxMsgFrame) != eCANFD_TRANSMIT_SUCCESS) {
                printf("Failed to transmit message\n");
            }
```

```
        }
        printf("\n Transmit Done\n");
    } else {
        printf("Start to CAN Bus Receiver :\n");
        /* Receive 2 messages with different ID */
        do {
            /* check for any received messages on CAN FD0 message buffer 0 */
            if(CANFD_ReadRxBufMsg(CANFD0, 0, &sRxFrame) == eCANFD_RECEIVE_SUCCESS) {
                printf("Message Data : ");
                for(u8Cnt = 0; u8Cnt < sRxFrame.u32DLC; u8Cnt++) {
                    printf("%02X ,", sRxFrame.au8Data[u8Cnt]);
                }
                printf("ID: 0x%08X\n", sRxFrame.u32Id);

                /* Check Standard ID number and Data Length*/
                if(((sRxFrame.u32Id != 0x0F6) && (sRxFrame.u32Id != 0x036)) ||
                   (RX_TEMP_LEN != sRxFrame.u32DLC) || (sRxFrame.eIdType != eCANFD_SID)) {
                    u8ErrFlag = 1;
                }
                if(u8ErrFlag == 1) {
                    printf("CAN STD ID or Data Error \n");
                }

                u8RxTestNum++;
            }
        } while(u8RxTestNum < 2);
        printf("\n Receive Done\n");
    }
}
```

3. 软件与硬件需求

3.1 软件需求

- BSP 版本
  - M251_M252_M254_M256_M258_Series_BSP_CMSIS_V3.00.006
  - M253_Series_BSP_CMSIS_V3.00.008
- IDE 版本
  - Keil uVersion 5.38

3.2 硬件需求

- 电路组件
  - NuMaker-M251SD V1.1
  - NuMaker-M253LEB V1.0
- 线路示意图
  - 为确保工作电压稳定，另使用电源供应器提供 MCP2515 模块 5V 电压，其他线路连接请参考图 3-1 与表 3-1 的说明与线路图。

Power Supply	MCP2515	NuMaker-M251SD	NuMaker-M253LEB
5V	VCC	-	-
-	SCK	SPI0_CLK Pin 42 (PD.2)	-
-	SI	SPI0_MOSI Pin 44 (PD.0)	-
-	SO	SPI0_MISO Pin 43 (PD.1)	-
-	CS	SPI0_SS Pin 41 (PD.3)	-
GND	GND	GND	GND
-	CAN_H	-	CAN_H
-	CAN_L	-	CAN_L

表 3-1 脚位说明与 MCU 脚位对照表

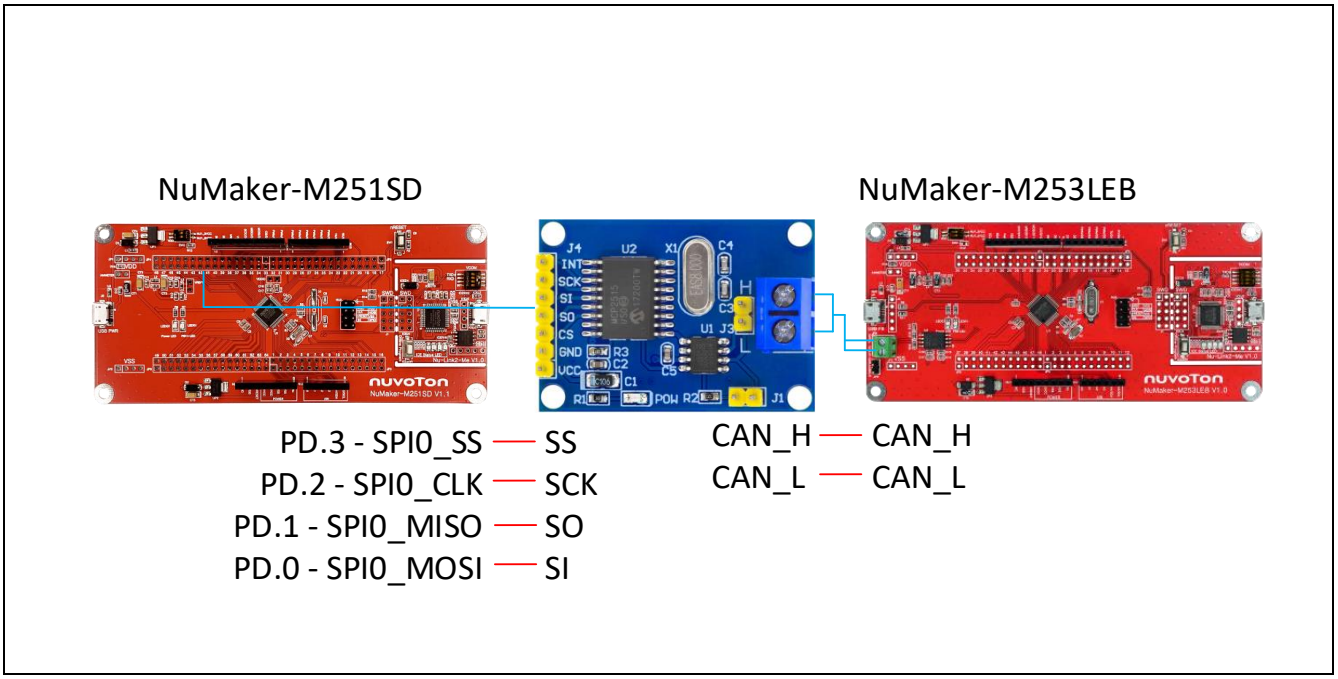
NuMaker-M251SD
NuMaker-M253LEB
PD.3 - SPI0_SS
SS
PD.2 - SPI0_CLK
SCK
PD.1 - SPI0_MISO
SO
PD.0 - SPI0_MOSI
SI
CAN_H
CAN_H
CAN_L
CAN_L

图 3-1 线路连接示意图

4. 目录信息

EC_M251_CAN_Controller_MCP2515_V1.00	
M251_SPI_MCP2515_Master	
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	
Project	Source file of example code
M253_CAN_Slave	
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	
Project	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 *M251_SPI_MCP2515_Master.uvprojx 以及 M253_CAN_Slave.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
2024.05.17	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.