使用 UART 模擬 ISO7816-3 介面

NuMicro® 32 位元系列微控制器範例代碼介紹

文件資訊

應用簡述	使用 M251 的 UART 模擬基於 ISO7816-3 標準的 Smart Card 介面。
BSP 版本	M251_M252_M254_M256_M258_Series_BSP_CMSIS_V3.02.007
開發平臺	NuMaker-M251SD 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

本範例代碼將示範如何使用 M251 的 UART 和 CLKO 模擬基於 ISO7816-3 標準的 Smart Card 主機介面。實現對 Smart Card 的啟動、冷重置、熱重置、交易、釋放等操作，其中交易使用 T0 非同步半雙工字元傳輸協議。

1.1 原理

本範例代碼主要用到四個 IP: UART0、UART1、CLKO 以及 TIMER0。

- UART0 作為調試串口，主要用於列印調試資訊和 Smart Card 的應答資料，以及輸入交互資訊。
- UART1 用來模擬 ISO7816-3 的 DAT。ISO7816-3 採用非同步半雙工通訊，收發共用一根 DAT 線，因此需要將 UART1 的 TX 和 RX 連在一起。
- CLKO 用來輸出 ISO7816-3 的時鐘 CLK。
- TIMER0 用來實現對 Smart Card 操作時序中的各種延時。

ISO7816-3 傳輸中字元框架格式如圖 1-1 所示。

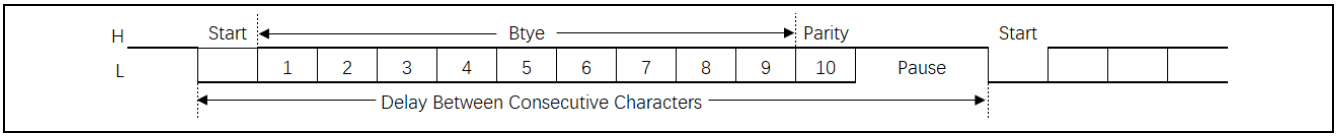


圖 1-1 字元幀

基本時間單元是指傳輸字元幀的一位元需要的時間，計算公式如下：

$$1\ etu = \frac{F}{D} \times \frac{1}{f}$$

參數 f 是 CLK 輸出頻率，範例中由 CLKO 輸出，設置為 3 MHz。

參數 F 和 D 分別是時鐘轉換因數和串列傳輸速率調整因數，在通訊過程中使用的 etu 依賴於 F 和 D 的實際值，在 ATR 期間默認 F = 372，D=1。

由上述可知 UART1 的串列傳輸速率是 1/(1 etu)。

Timer0 的 CLK 也設置為 3 MHz。

當接收時，將 PB.6 引腳複用功能設置為 UART1_RX，PB.7 設置為 GPIO 輸入，使兩個引腳都處於輸入模式，不會影響 Smart Card 輸出資料。同時還需要使能接收資料可用和接收超時中斷，接收完成後需要禁止中斷。

當發送時，保持PB.6為UART1_RX功能，因為UART1_RX是輸入，不會影響DAT上的資料。將 PB.7 引腳複用功能設置為 UART1_TX，並使能發送保持暫存器空中斷，在中斷處理函式裡完成資料的發送。發送完成後需要禁止發送保持暫存器空中斷，並且將 PB.7 再次設置為 GPIO 輸入模式，防止 PB.7 和 Smart Card 同時輸出。

本範例對Smart Card的操作步驟如下：

1) 啟動 Smart Card
2) 冷重置
3) 熱重置(冷重置失敗)
4) 接收重置應答 ATR
5) PPS 交換
6) 交易：選擇文檔、獲取應答
7) 釋放序列

1.2 執行結果

本範例代碼執行結果如圖 1-2 所示。PC 上打開串口工具，連接 UART0 用於列印調試資訊。

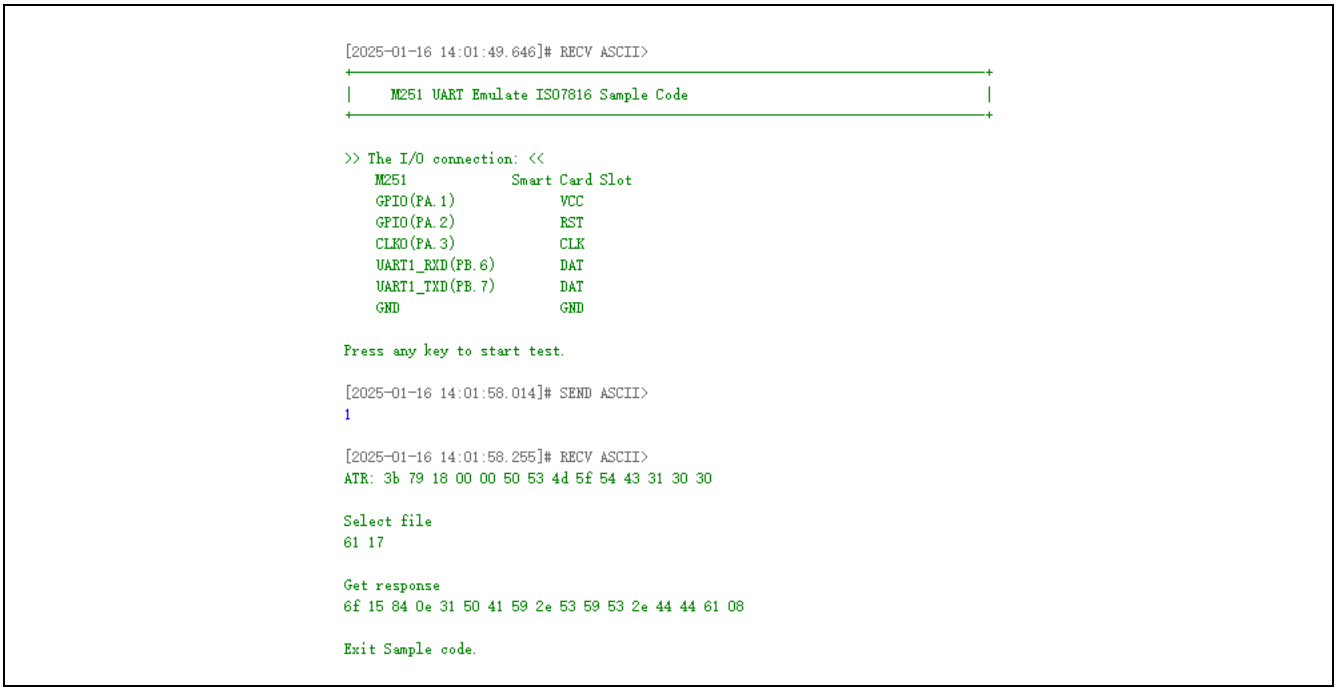


圖 1-2 執行結果

2. 代碼介紹

本範例程式主要分成三部分：硬體驅動、Smart Card 協議庫、主函式。

硬體驅動

IP 和引腳的定義，代碼位於 *sc_drv_config.h*。如果要使用其他引腳、UART 或 Timer，只需要修改這裡的定義，而不需要修改驅動代碼。

```
/* Define function of multi-function pin */
#define SET_CLKO_PA3()       SYS->GPA_MFPL = (SYS->GPA_MFPL & ~SYS_GPA_MFPL_PA3MFP_Msk) |
                             SYS_GPA_MFPL_PA3MFP_CLKO
#define SET_UART1_RX_PB6()   SYS->GPB_MFPL = (SYS->GPB_MFPL & ~SYS_GPB_MFPL_PB6MFP_Msk) |
                             SYS_GPB_MFPL_PB6MFP_UART1_RXD
#define SET_UART1_TX_PB7()   SYS->GPB_MFPL = (SYS->GPB_MFPL & ~SYS_GPB_MFPL_PB7MFP_Msk) |
                             SYS_GPB_MFPL_PB7MFP_UART1_TXD
#define SET_GPIO_PB7()       SYS->GPB_MFPL = (SYS->GPB_MFPL & ~SYS_GPB_MFPL_PB7MFP_Msk)

/* VCC */
#define VCC       PA1  /* pin */
#define VCC_PORT  PA   /* port */
#define VCC_BIT   BIT1 /* bit */

/* RST */
#define RST       PA2  /* pin */
#define RST_PORT  PA   /* port */
#define RST_BIT   BIT2 /* bit */

/* CLK */
#define SC_CLK        PA3  /* pin */
#define CLK_PORT      PA   /* port */
#define CLK_BIT       BIT3 /* bit */
#define SET_CLKO_PIN  SET_CLKO_PA3 /* multi-function = CLKO */

/* UART_TX */
#define UART_TX             PB7   /* pin */
#define UART_TX_PORT        PB    /* port */
#define UART_TX_BIT         BIT7  /* bit */
#define SET_UART_TX_PIN     SET_UART1_TX_PB7 /* multi-function = UART1_TX */
#define CANCEL_UART_TX_PIN  SET_GPIO_PB7     /* multi-function = GPIO */

/* UART_RX */
#define UART_RX          PB6  /* pin */
#define UART_RX_PORT     PB   /* port */
#define UART_RX_BIT      BIT6 /* bit */
#define SET_UART_RX_PIN  SET_UART1_RX_PB6 /* multi-function = UART1_RX */

/* UART */
#define UART      UART1      /* Uart select */
#define UART_IRQ  UART1_IRQn /* IRQ */
#define UART_IRQHandler UART1_IRQHandler /* Interrupt handling function */
#define UART_CLK_SELECT()   CLK_SetModuleClock(UART1_MODULE, CLK_CLKSEL1_UART1SEL_HIRC,
```

```
 CLK_CLKDIV0_UART1(1)) /* Clock source */
#define UART_CLK_ENABLE()   CLK_EnableModuleClock(UART1_MODULE) /* Enable clock */

/* Timer */
#define TIMER   TIMER0
#define TIMER_CLK_SELECT()   CLK_SetModuleClock(TMR0_MODULE, CLK_CLKSEL1_TMR0SEL_HIRC, 0)
#define TIMER_CLK_ENABLE()   CLK_EnableModuleClock(TMR0_MODULE) /* Enable clock */
```

硬體相關驅動函式，代碼位於 *sc_drv.c*。

- Smart Card 介面初始化

```
void SC_Init(void)
{
    /* 1. Set pin multi-function */
    SC_Pin_Init();

    /* 2. UART init */
    UART_Open(UART, 8065);/* By default, F=372 D = 1, Baud Rate = f/372 = 3M/372 = 8065 */

    /* Set Rx Time-out counter */
    UART_SetTimeoutCnt(UART, 0x10);

    /* Set Rx FIFO Interrupt Trigger Level */
    UART->FIFO &= ~ UART_FIFO_RFITL_Msk;

    UART->FIFO |= UART_FIFO_RFITL_4BYTES;

    NVIC_EnableIRQ(UART_IRQn);

    /* 3. Set the clock output of the card reader using clko. f = 3M Hz */
    CLK_EnableModuleClock(CLKO_MODULE); /* Enable clk */

    /* Select CKO clock source. clock source = HCLK, No frequency division */
    CLK_SetModuleClock(CLKO_MODULE, CLK_CLKSEL1_CLKOSEL_HCLK, NULL);

    /* CKO = clock source / 2^(u32ClkDiv + 1) */
    CLK->CLKOCTL = (CLK->CLKOCTL & ~(CLK_CLKOCTL_FREQSEL_Msk | CLK_CLKOCTL_DIV1EN_Msk))
                   | (3 << CLK_CLKOCTL_FREQSEL_Pos) /* div = 2^(3+1) */
                   | (0 << CLK_CLKOCTL_DIV1EN_Pos); /* div enable   f = 48/2^3 = 3M */

    /* 4. Set timer for delay. */
    /* Set timer frequency to SC_CLK (3M Hz) */
    TIMER->CTL = TIMER_CONTINUOUS_MODE | 15;/* 3M = HCLK/(15+1) */
}
```

- DAT 輸出使能函式

```
void SC_DAT_OutPut_Enable(void)
{
    /* Set multi-function pin for UART1 TXD */
    SET_UART_TX_PIN();

    /* Enable transmit holding register empty interrupt */
    UART_EnableInt(UART, UART_INTEN_THREIEN_Msk);
```

```
}
```

- DAT 輸出禁止函式

```
void SC_DAT_OutPut_Disable(void)
{
    /* Disable TX interrupt */
    UART_DisableInt(UART, UART_INTEN_THREIEN_Msk);

    /* Wait for transmission to end */
    while (!UART_IS_TX_EMPTY(UART));

    /* Disable TX pin. Set multi-function pin for GPIO */
    CANCEL_UART_TX_PIN();
}
```

- DAT 輸入使能函式

```
void SC_DAT_Input_Enable(void)
{
    /* Enable Receive Data Available Interrupt and RX Time-out Interrupt. */
    UART_EnableInt(UART, (UART_INTEN_RDAIEN_Msk | UART_INTEN_RXTOIEN_Msk));
}
```

- DAT 輸入禁止函式

```
void SC_DAT_Input_Disable(void)
{
    UART_DisableInt(UART, (UART_INTEN_RDAIEN_Msk | UART_INTEN_RXTOIEN_Msk));
}
```

- 延時函式

```
void SC_Timer_Delay(uint32_t ticks)
{
    uint32_t t1;

    t1 = TIMER->CNT;

    while(TIMER->CNT - t1 < ticks);
}
```

Smart Card 協議庫

Smart Card 協議庫的代碼全部位於 SmartcardT0Lib 資料夾下。包含用於應用程式調用的 API 介面函數、協議的實現代碼和協議內部調用的函數。

Smart Card 協議庫的 API 介面函數，代碼位於 *SmartcardT0Lib\Source\sclib_api.c* 文件下。

- 卡啟動

```
int32_t SCLIB_Activate(uint32_t num, uint32_t u32EMVCheck)
{
    SCLIB_DEV_T *dev = &_scDev[num];
```

```
    int32_t retval, clk;

    /* 1. Read clock of card slot. */
    clk = _SCLIB_GetInterfaceClock(num);

    /* Upper and lower clock limits */
    if(clk < 1000 || clk > 5000)
        return SCLIB_ERR_CLOCK;

    /* 2. Set card status=card present */
    dev->bCardRemoved = SCLIB_CARD_PRESENT;

    /* 3. Set whether to check EMV */
    dev->EMV = u32EMVCheck;

    /* 4. Cold Reset */
    retval = SCLIB_ColdReset(num);

    /* 5. Warm Reset */
    if(retval == SCLIB_ERR_ATR_INVALID_PARAM) /* Cold Reset failed. */
        retval = SCLIB_WarmReset(num); /* Warm Reset */

    /* 6. Set Tx and Rx buffers */
    dev->snd_buf = &g_TxBUF[0]; /* Tx buffer */

    dev->rcv_buf = &g_RxBUF[0]; /* Rx buffer */

    return(retval);
}
```

- 冷重置

```
int32_t SCLIB_ColdReset(uint32_t num)
{
    SCLIB_DEV_T *dev = &_scDev[num]; /* Device pointer */

    /* 1. Cold Reset */
    dev->errno = _SCLIB_ResetCard(num, SCLIB_COLDRESET);/* Smartcard interface number,
    Cold Reset*/

    /* Cold Reset failed. */
    if(dev->errno != SCLIB_SUCCESS)
        return (dev->errno);

    /* 2. Update card Capabilities, read from ATR. */
    dev->pps_complete = 0;

    dev->errno = _SCLIB_UpdateCardCapabilities(num);

    /* Update failed */
    if(dev->errno != SCLIB_SUCCESS)
        return (dev->errno);

    /* 3. Parsing ATR, and PPS exchange */
    dev->errno = _SCLIB_ParseATR(num, SCLIB_COLDRESET);/* Smartcard interface number,
    Cold Reset */
```

```
    if(dev->errno != SCLIB_SUCCESS) /* Failed */
        SCLIB_ERR("Cold Reset Parse ATR..Error :%d \n", dev->errno);

    if(dev->errno == SCLIB_SUCCESS) /* Success */
        /* Smartcard has been activated. */
        dev->openflag = 1;

    return (dev->errno);
}
```

- 啟動傳輸

```
int32_t SCLIB_StartTransmission(uint32_t num, uint8_t *cmdBuf, uint32_t cmdLen, uint8_t
*rspBuf, uint32_t *rspLen)
{
    int32_t retval = SCLIB_SUCCESS;
    SCLIB_DEV_T *dev = &_scDev[num];

   /* 1. Not activate yet.*/
    if(dev->openflag == 0)      /* Try to access smartcard before activation. */
        return SCLIB_ERR_DEACTIVE;

    /* 2. Smartcard busy. */
    if(dev->op_state)           /* Smartcard busy, previous transmission not complete. */
        return SCLIB_ERR_CARDBUSY;

    /* Give up data remain in dev->buf, if anything remains... */
    dev->snd_pos = 0; /* Send buffer offset = 0 */

    dev->rcv_pos = 0; /* Receive buffer offset = 0 */

    /* 3. Send a command and get response from smartcard according to protocol */
    if(dev->CardCapabilities.Protocol.Selected == SCLIB_PROTOCOL_T0) /* T0 */
    {
        /* Using T=0 protocol */
        retval = _SCLIB_T0Transmit(num,     /* Smartcard interface number */
                                   cmdBuf,  /* cmd buffer */
                                   cmdLen,  /* cmd len */
                                   rspBuf,  /* response buffer */
                                   rspLen); /* response len */
    }
    else if(dev->CardCapabilities.Protocol.Selected == SCLIB_PROTOCOL_T1) /* T1 */
    {
        retval = _SCLIB_T1Transmit(num, cmdBuf, cmdLen, rspBuf, rspLen);
    }

    /* 4. Stop all timers here, the transmission is complete no matter success or not. */
    SC_Timer_Stop();

    return retval;
}
```

- 卡傳輸事件處理

此函數需要在 UART 的中斷處理函數裡被調用。

```
uint32_t SCLIB_CheckTxRxEvent(uint32_t num)
{
   ……
    /* 1. Transmit buffer empty interrupt */
    if (SC_GetTxIntFlag())
    {
        retval = 1;

        if( dev->op_state == SCLIB_OP_WRITE )/* Operation = Write */
        {
            ……
        }
    }
    /* 2. Receive Data Available Interrupt or Rx TimeOut interrupt. */
    if(SC_GetRxIntFlag())
    {
        ……
        if(dev->op_state == SCLIB_OP_READ) /* Operation = Read */
        {
            ……
        }
        else if(dev->op_state == SCLIB_OP_ATR_READ)  /* Operation = Read ATR */
        {
            ……
        }
        else {
            ……
        }
    }
out:
    return retval;
}
```

另外 T0 非同步半雙工字元傳輸協議的實現代碼位於 *SmartcardT0Lib\Source\sclib_t0protocol.c*。Smart Card 協議庫的內部相關函數代碼位於 *SmartcardT0Lib\Source\sclib_util.c*。

主函式

```
int32_t main(void)
{
    SCLIB_CARD_INFO_T s_info;
    int retval, i;
    uint32_t u32Ret;

    /* Unlock protected registers */
    SYS_UnlockReg();

    /* Init System, peripheral clock and multi-function I/O */
    SYS_Init();

    /* Lock protected registers */
    SYS_LockReg();
```

```
#if defined (__GNUC__) && !defined(__ARMCC_VERSION) && defined(OS_USE_SEMIHOSTING)
    initialise_monitor_handles();
#endif

    /* Init UART0 for printf and test */
    UART0_Init();

    printf("+-------------------------------------------------------------------------+\n");
    printf("|     M251 UART Emulate ISO7816 Sample Code                               |\n");
    printf("+-------------------------------------------------------------------------+\n\n");
    printf(">> The I/O connection: <<\n");
    printf("    M251             Smart Card Slot\n");
    printf("    GPIO(PA.1)             VCC\n");
    printf("    GPIO(PA.2)             RST\n");
    printf("    CLKO(PA.3)             CLK\n");
    printf("    UART1_RXD(PB.6)        DAT\n");
    printf("    UART1_TXD(PB.7)        DAT\n");
    printf("    GND                    GND\n");
    printf("\nPress any key to start test.\n");

    getchar();

    /* Init ISO7816 */
    SC_Init();

    /* Activate Smartcard interface 0 */
    retval = SCLIB_ActivateDelay(0,      /* Smartcard interface number */
                                 FALSE,  /* Disable EMV error check */
                                 10000); /* Delay clk */

    if (retval == SCLIB_SUCCESS) /* Activated */
    {
        /* Copy ATR information to variable. */
        SCLIB_GetCardInfo(0, &s_info);  /* Smartcard interface number, variable */

        /* Print ATR information */
        printf("ATR: ");

        for (i = 0; i < s_info.ATR_Len; i++)
            printf("%02x ", s_info.ATR_Buf[i]);

        printf("\n\n");

        /* Smartcard transmissiontransmission test: Select file MF */
        u32Ret = SCLIB_StartTransmission(0,              /* Smartcard interface number */
                                         g_cmd11buf,     /* Cmd buffer */
                                         g_cmd1len,      /* Cmd len */
                                         g_cmdrsp,       /* Response buffer */
                                         &g_cmd1rsplen); /* Response len */

        if (u32Ret != 0) /* Error */
        {
            dwErrorNum++;
        }
        else
```

```
        {
            /* Print cmd response data */
            printf("Select file\n");

            for (i = 0; i < g_cmd1rsplen; i++)
                printf("%02x ", g_cmdrsp[i]);

            printf("\n\n");

            /* Smartcard transmission test: get response */
            u32Ret = SCLIB_StartTransmission(0,           /* Smartcard interface number */
                                             g_cmd3buf,      /* Cmd buffer */
                                             g_cmd3len,      /* Cmd len */
                                             g_cmdrsp,       /* Response buffer */
                                             &g_cmd1rsplen); /* Response len */

            if (u32Ret != 0)
            {
                dwErrorNum++;
            }
            else
            {
                /* Print cmd response data */
                printf("Get response\n");

                for (i = 0; i < g_cmd1rsplen; i++)
                    printf("%02x ", g_cmdrsp[i]);

                printf("\n");
            }
        }
    }
    else
    {
        printf("Smartcard activate failed\n");
    }

    printf("\nExit Sample code.\n");

    while (1);

}
```

3. 軟體與硬體需求

3.1 軟體需求

- BSP 版本
  - M251_M252_M254_M256_M258_Series_BSP_CMSIS_V3.02.007
- IDE 版本
  - Keil uVision 5.40.0.0

3.2 硬體需求

- 電路元件
  - NuMaker-M251SD V1.0
- 線路示意圖
  - 將 PA.1、PA.2、PA.3 分別連接到 Smart Card 槽的 VCC、RST、CLK。
  - 將 UART1 TX (PB.7) 腳和 UART1_RX(PB.6) 都連接到 Smart Card 槽的 DAT。
  - 將 UART0 TX (PB.13) 腳和 UART0 RX (PB.12) 腳通過 Nu-Link-me 轉接到 PC。

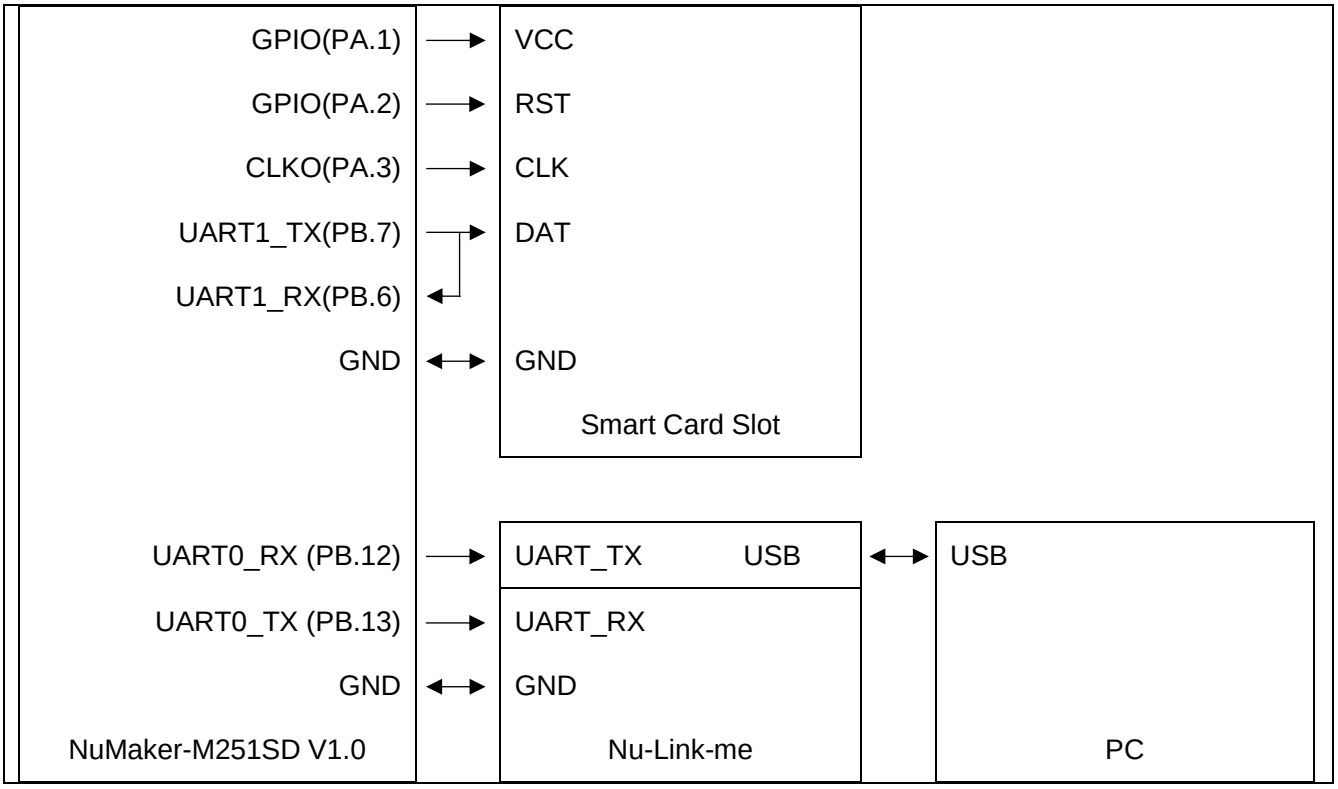

GPIO(PA.1)
GPIO(PA.2)
CLKO(PA.3)
UART1_TX(PB.7)
UART1_RX(PB.6)
GND
VCC
RST
CLK
DAT
GND
Smart Card Slot
UART0_RX (PB.12)
UART0_TX (PB.13)
GND
UART_TX
USB
UART_RX
GND
USB
NuMaker-M251SD V1.0
Nu-Link-me
PC

圖 3-1 線路示意圖

4. 目錄資訊

目錄結構如下圖所示。

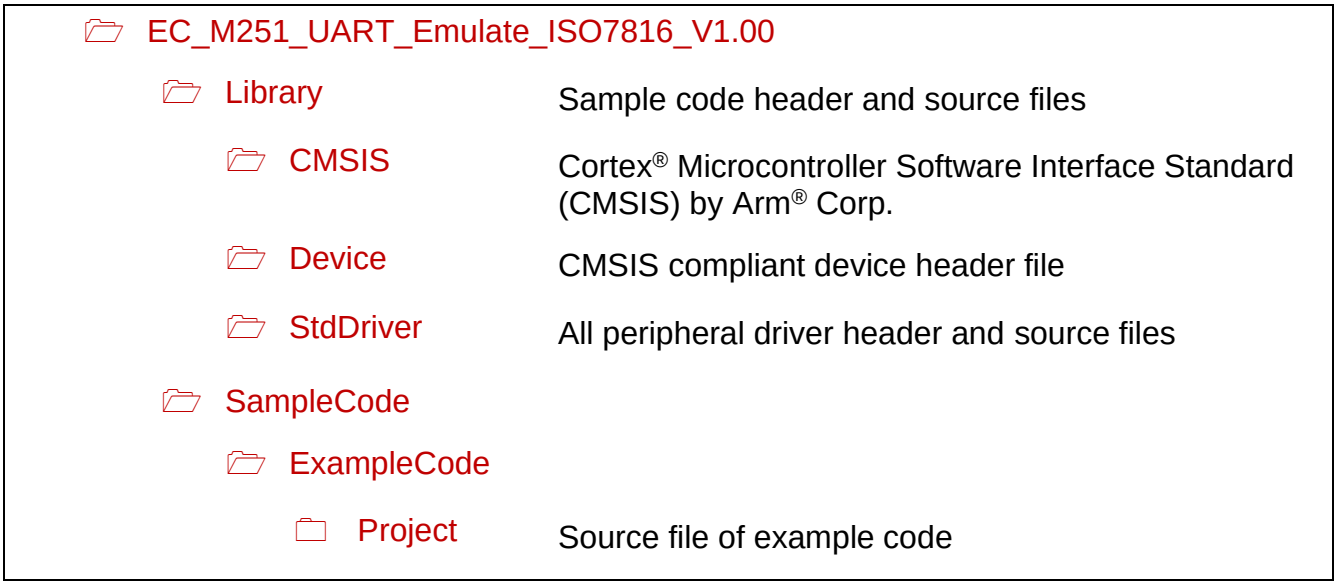


圖 4-1 目錄資訊

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 *M251_UART_Emulate_ISO7816.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
2025.02.11	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.