M032 FreeMODBUS 從機 RTU 模式移植

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

應用簡述	此範例代碼示範如何將 FreeMODBUS 從機 RTU 模式移植到 M032 系列微控制器。
BSP 版本	M031_Series_BSP_CMSIS_V3.06.00
開發平臺	NuMaker-M032SE V1.3

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

本範例代碼展示了如何使用 UART 介面並在 RTU 模式下實現 MODBUS 從機。MODBUS RTU 是主 / 從結構的開放式串列通訊協議。MODBUS RTU 消息編碼是 16 進制及具有 CRC 校驗封包。

本代碼移植 FreeMODBUS 作為 MODBUS 從機實作，FreeMODBUS 是一個針對嵌入式應用免費通用 MODBUS 協議庫，FreeMODBUS 提供了 RTU，ASCII 傳輸模式及 TCP 協定支援。FreeMODBUS 遵循 BSD 許可證，這意味著用戶可以將 FreeMODBUS 應用於商業環境中。

使用者可以下載 FreeMODBUS 源代碼從以下官網連結：

FreeMODBUS Downloads - Embedded Experts (embedded-experts.at)

1.1 原理

MODBUS 協定為主(master)/從(slave)通訊協定，每個從機必須有唯一的地址，最多接 247 個裝置，MODBUS 通訊介面可以選用 RS232、RS485 或 Ethernet。其通信遵循以下的過程:

1. 主設備向從設備發送請求。
2. 從設備分析並處理主設備的請求，然後向主設備發送結果。
3. 如果出現任何差錯，從設備將返回一個異常功能碼。

MODBUS 的傳輸模式，有以下 3 種:

- ASCII (American Standard Code for Information Interchange, 美國資訊交換標準代碼)。
- RTU (remote terminal unit, 遠程終端單元)。
- TCP (Transmission Control Protocol, 傳輸控制協定)。

本範例 MODBUS 使用 RTU 模式，使用 16 進制編碼，2 個位元組的 CRC 校驗碼，消息中的每個位元組包含兩個 4 位元的 16 進位字元，在同樣的串列傳輸速率下，可比 ASCII 方式傳送更多的資料，每兩個位元組之間發送或者接收的時間間隔不能超過 3.5 倍字元傳輸時間。

MODBUS RTU 數據框架由四部分組成，設備位址、功能碼、資料和校驗，長度如表 1-1。

slave 地址	功能碼	資料(依功能碼決定)	CRC
1 位元組	1 位元組	0 ~ 252 位元組	2 位元組

表 1-1 MODBUS RTU 數據框架格式

表 1-2 列出常用功能碼如下:

功能碼	名稱	說明
0x1	Read coil Status	取得一組數位輸出的目前狀態
0x2	Read Input Status	取得一組數位輸入的目前狀態
0x3	Read Holding Registers	取得一組類比輸出的目前狀態
0x4	Read Input Registers	取得一組類比輸入的目前狀態

0x5	Force Single Coil	強制設定單個數位輸出的值
0x6	Preset Single Register	強制設定單個類比輸出的值
0xF	Force Multiple Coils	強制設定多個數位輸出的值
0x10	Preset Multiple Registers	強制設定多個類比輸出的值

表 1-2 MODBUS RTU 常用功能碼

slave address/slave ID 設備端 ID，當要讀寫僕機時，必須先知道該設備的 address/ID。常用的 MODBUS 功能碼– 0x03。在 0x03 所定義的功能為讀取多個暫存器，用來讀取一連續位址的資料。以下為 MODBUS RTU 通訊範例:

MODBUS RTU master 讀取格式:
設備 ID(slave address/ID) + 0x03 + 讀取起始位置(2 位元組) + 讀取的數量(2 位元組) + CRC16 。

MODBUS RTU slave 回覆格式:
設備 ID(slave address/ID) + 0x03 + 回覆資料的位元組數 + 資料 1(1 位元組) + … + 資料 n(1 位元組) + CRC16。

1.2 執行結果

在測試 MODBUS 協議，可以下載 Modpoll 將個人電腦(PC)作為主控端，使用 Numaker - M032SE 開發板的 NU-LINK2-VCOM 與 M032 的 UART 連接測試。

Modpoll 官網及下載連結：

Free Modbus Master Simulator and Test Tool (modbusdriver.com)

先下載並執行 M032 的 FreeMODBUS 移植代碼，在 modpoll 的資料夾下，在 PC 命令列上執行：

modpoll.exe -m rtu -a 10 -r 1000 -c 4 -t 3 -b 38400 -d 8 -p none COM[n, 串口編號]

其 modpoll 参数，请参考官网说明，COM[n] 请设定为 PC 实际使用串口。

```
\modpoll.exe -m rtu -a 10 -r 1000 -c 4 -t 3 -b 38400 -d 8 -p none  COM31
modpoll 3.4 - FieldTalk(tm) Modbus(R) Master Simulator
Copyright (c) 2002-2013 proconX Pty Ltd
Visit http://www.modbusdriver.com for Modbus libraries and tools.

Protocol configuration: Modbus RTU
Slave configuration...: address = 10, start reference = 1000, count = 4
Communication.........: COM31, 38400, 8, 1, none, t/o 1.00 s, poll rate 1000 ms
Data type.............: 16-bit register, input register table

-- Polling slave... (Ctrl-C to stop)
[1000]: 16513
[1001]: 0
[1002]: 0
[1003]: 0
-- Polling slave... (Ctrl-C to stop)
[1000]: 12967
[1001]: 0
[1002]: 0
[1003]: 0
-- Polling slave... (Ctrl-C to stop)
[1000]: 5544
[1001]: 0
[1002]: 0
[1003]: 0
```

圖 1-1 執行 modpoll 通訊結果

2. 代碼介紹

M032 的 FreeMODBUS 移植需要 M032 的晶片功能有 UART 及 TIMER。在 portserial.c 代碼中使用 UART 必須移植 vMBPortSerialEnable、vMBPortClose、xMBPortSerialInit、xMBPortSerialPutByte、xMBPortSerialGetByte 和 UART02_IRQHandler。範例代碼如下：

```
/* ---------------------- Start implementation -----------------------------*/
/**
  * @brief      vMBPortSerialEnable
  * @param[in]  BOOL xRxEnable, BOOL xTxEnable
  * @return     None
  * @details    This function to enable/disabele UART TX and RX interrupt.
  */
void  vMBPortSerialEnable(BOOL xRxEnable, BOOL xTxEnable)
{
    __disable_irq();

    if (xRxEnable)
    {
        UART_ENABLE_INT(UART0, UART_INTEN_RDAIEN_Msk);
    }
    else
    {
        UART_DISABLE_INT(UART0, UART_INTEN_RDAIEN_Msk);
    }
    if (xTxEnable)
    {
        UART_ENABLE_INT(UART0, UART_INTEN_THREIEN_Msk);
    }
    else
    {
        UART_DISABLE_INT(UART0, UART_INTEN_THREIEN_Msk);
    }
    NVIC_EnableIRQ(UART02_IRQn);
    __enable_irq();
}

/**
  * @brief      vMBPortClose
  * @param[in]  None
  * @return     None
  * @details    This function to Close UART port.
  */
void vMBPortClose(void)
{
    /* Disable RDA and THRE Interrupt */
    UART_DISABLE_INT(UART0, UART_INTEN_RDAIEN_Msk | UART_INTEN_THREIEN_Msk);
    NVIC_DisableIRQ(UART02_IRQn);
}

/**
  * @brief      xMBPortSerialInit
  * @param[in]  UCHAR ucPORT, ULONG ulBaudRate, UCHAR ucDataBits, eMBParity eParity
  * @return     TURE or FALSE
  * @details    This function to Open UART port and set ulBaudRate, ucDataBits, eParity.
  */
BOOL xMBPortSerialInit(UCHAR ucPORT, ULONG ulBaudRate, UCHAR ucDataBits, eMBParity
eParity)
{
    if (ucPORT != 0) return FALSE;
```

```
    /* Step 1. GPIO initial */
    /* Set PB multi-function pins for UART0 RXD=PB.12 and TXD=PB.13 */
    SYS->GPB_MFPH = (SYS->GPB_MFPH & ~(SYS_GPB_MFPH_PB12MFP_Msk|
                     SYS_GPB_MFPH_PB13MFP_Msk))|\
                    (SYS_GPB_MFPH_PB12MFP_UART0_RXD | SYS_GPB_MFPH_PB13MFP_UART0_TXD);

    /* Step 2. Enable and Select UART clock source*/
    /* Enable UART0 clock */
    CLK_EnableModuleClock(UART0_MODULE);

    /* UART0 clock source */
    /* Select UART module clock source as HIRC and UART module clock divider as 1 */
    CLK_SetModuleClock(UART0_MODULE, CLK_CLKSEL1_UART0SEL_HIRC, CLK_CLKDIV0_UART0(1));

    /* Step 3. Select Operation BaudRate */
    SYS_ResetModule(UART0_RST);
    UART_Open(UART0, ulBaudRate);

    /* Step 4. Set Configure parity*/
    switch (eParity)
    {
        case MB_PAR_NONE:
            UART0->LINE &= ~UART_PARITY_EVEN;
            break;

        case MB_PAR_EVEN:
            UART0->LINE &= ~UART_PARITY_EVEN;
            UART0->LINE |= UART_PARITY_EVEN;
            break;

        case MB_PAR_ODD:
            UART0->LINE &= ~UART_PARITY_EVEN;
            UART0->LINE |= UART_PARITY_ODD;
            break;
    }

    /* Step 5. Set Configure data length*/
    UART0->LINE &= ~UART_WORD_LEN_8;

    switch (ucDataBits)
    {
        case 5:
            UART0->LINE |= UART_WORD_LEN_5;      //5 bits Data Length
            break;

        case 6:
            UART0->LINE |= UART_WORD_LEN_6;      //6 bits Data Length
            break;

        case 7:
            UART0->LINE |= UART_WORD_LEN_7;      //7 bits Data Length
            break;

        case 8:
            UART0->LINE |= UART_WORD_LEN_8;      //8 bits Data Length
            break;
    }

    return TRUE;
}

/**
  * @brief      xMBPortSerialPutByte
  * @param[in]  CHAR ucByte
  * @return     TURE or FALSE
  * @details    This function to send ucByte to UART TX buffer.
  */
```

```
BOOL xMBPortSerialPutByte(CHAR ucByte)
{
    while (UART_GET_TX_FULL(UART0) != 0x00); //check Tx Fifo full.

    UART0->DAT = (uint8_t) ucByte;

    return TRUE;
}
/**
  * @brief      xMBPortSerialGetByte
  * @param[in]  CHAR *pucByte
  * @return     TURE or FALSE
  * @details    This function to get pucByte from UART RX buffer.
  */
BOOL xMBPortSerialGetByte(CHAR *pucByte)
{

    while (UART_GET_RX_EMPTY(UART0) != 0x00); //check Rx Empty

    *pucByte = UART0->DAT;
    return TRUE;
}

/**
  * @brief      UART02_IRQHandler
  * @param[in]  None
  * @return     None
  * @details    This function is UART TX and RX interrupt handler function.
  */
void UART02_IRQHandler(void)
{

    if (UART_GET_INT_FLAG(UART0, UART_INTSTS_RDAINT_Msk))
    {

        /* Get all the input characters */
        while (UART_IS_RX_READY(UART0))
        {
         pxMBFrameCBByteReceived();
        }

    }
    else if (UART_GET_INT_FLAG(UART0, UART_INTSTS_THREINT_Msk))
    {
    pxMBFrameCBTransmitterEmpty();
    }
}
```

在 porttimer.c 代碼中使用 TIMER 必須移植代碼，xMBPortTimersInit、vMBPortTimersEnable、vMBPortTimersDisable、TMR0_IRQHandler。

```
/**
  * @brief      xMBPortTimersInit
  * @param[in]  USHORT usTim1Timerout50us
  * @return     TURE or FALSE
  * @details    This function to Open Timer0 and time out count(50us unit).
  */
BOOL xMBPortTimersInit(USHORT usTim1Timerout50us)
{
    float freq_exp = 0;
    freq_exp = 1.0 / ((float)usTim1Timerout50us * 0.00005);//convert time to frequence
```

```
    /* Enable Timer0 clock source */
    CLK_EnableModuleClock(TMR0_MODULE);
    /* Select Timer0 clock source as internal 48M */
    CLK_SetModuleClock(TMR0_MODULE, CLK_CLKSEL1_TMR0SEL_HIRC, 0);

    /* Reset IP TMR0 */
    SYS_ResetModule(TMR0_RST);

    /* Open Timer0 in periodic mode, enable interrupt and interrupt tick per second */
    TIMER_Open(TIMER0, TIMER_PERIODIC_MODE, (uint32_t)freq_exp);

    /* Enable Timer0 interrupt */
    TIMER_EnableInt(TIMER0);
    /* Enable Timer0 NVIC */
    NVIC_EnableIRQ(TMR0_IRQn);
    return TRUE;
}

/**
  * @brief      vMBPortTimersEnable
  * @param[in]  None
  * @return     None
  * @details    This function to enable Timer0 and time count reset.
  */
void vMBPortTimersEnable(void)
{
    /* Reset Timer0 counter */
    TIMER_ResetCounter(TIMER0);
    /* Enable Timer0 */
    TIMER_Start(TIMER0);
}

/**
  * @brief      vMBPortTimersDisable
  * @param[in]  None
  * @return     None
  * @details    This function to disable Timer0 and time count reset.
  */
void vMBPortTimersDisable(void)
{
    /* Disable Timer0 */
    TIMER_Stop(TIMER0);
    /* Reset Timer0 counter */
    TIMER_ResetCounter(TIMER0);
}

/**
  * @brief      TMR0_IRQHandler
  * @param[in]  None
  * @return     None
  * @details    This function is Timer0 interrupt handler
  */
void TMR0_IRQHandler(void)
{
    /* Clear Timer0 interrupt flag */
    TIMER_ClearIntFlag(TIMER0);
```

```
    (void)pxMBPortCBTimerExpired();
}
```

在 main 函數中，呼叫 eMBInit，處理設定 RTU 模式，設定 slave address， UART 開啟及設定 baudrate，parity 設定，開啟 MODBUS 協議棧，最後在主迴圈中處理 eMBPoll 判斷 MODBUS 封包事件及處理。

```
int32_t main(void)
{
    …
    /* Enable the MODBUS RTU mode, MODBUS slave Address, UART port, BaudRate, Parity*/
    eMBInit(MB_RTU, 0x0A, 0, 38400, MB_PAR_NONE);

    /* Enable the Modbus Protocol Stack. */
    eMBEnable();

    for (;;)
    {
        /*eMBPoll for MODBUS event process*/
        (void)eMBPoll();

        /* Here we simply count the number of poll cycles. */
        /* The usRegInputBuf[0] will be accumulated in the main loop, allowing the master
        to read the data changes.*/
        usRegInputBuf[0]++;
    }
}
```

3. 軟體與硬體需求

3.1 軟體需求

- BSP 版本
  - M032_Series_BSP_CMSIS_V3.06.000
- IDE 版本
  - Keil uVersion 5.28

3.2 硬體需求

- 電路元件
  - NuMaker-M032SE V1.3
- 線路示意圖
  - 將 NuMaker-M032SE 的 Nu-link2-Me 開發板，在 VCOM 的 SW1, SW2 切換到 “ON” 位置致能 VOM，以主機端與 M032 作 MODBUS 通訊。

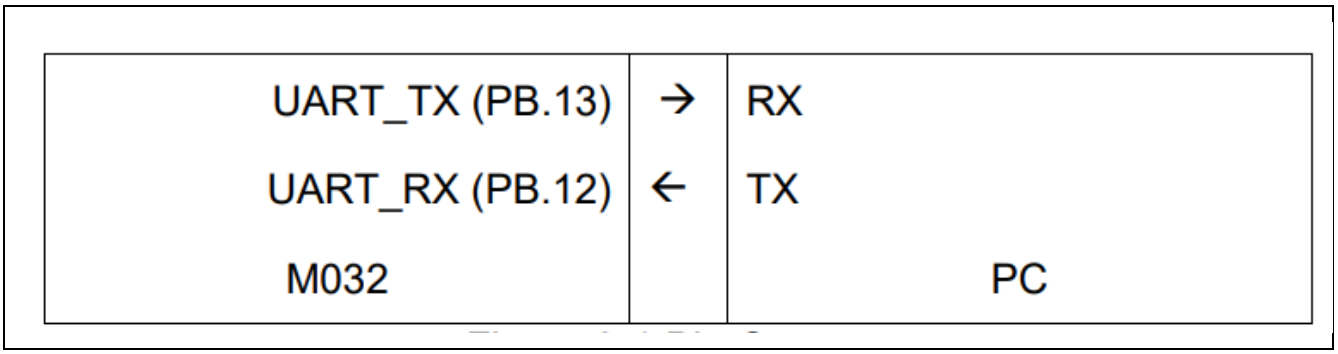

圖 3-1 線路示意圖

4. 目錄資訊

EC_M032_FreeMODBUS_RTU_Slave_Porting_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	
Project	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 *FreeMODBUS_RTU_Slave_Porting.uvprojx*。
2. 進入編譯模式介面
   - 編譯
   - 下載代碼至記憶體
   - 進入 / 離開除錯模式
3. 進入除錯模式介面
   - 執行代碼

6. 修訂紀錄

Date	Revision	Description
2024.05.20	1.00	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.