M252 PSIO 模擬 MDIO 波形

NuMicro® 32 位元系列微控制器範例代碼介紹

文件資訊

應用簡述	本範例代碼使用 M252 PSIO 模擬 MDIO 波形用以控制乙太網路 PHY (IP101GR)。
BSP 版本	M251_M252_M254_M256_M258_Series_BSP_CMSIS_V3.02.006
開發平台	NuMaker-M252SD V1.2

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

本範例代碼使用 M252 PSIO 介面來實作控制 MDIO 時序並實際用以控制乙太網路 PHY。PSIO 提供了一種簡單的方法來接收和傳送各種串列傳輸，透過設置時隙 (Slot)、檢查點 (Check point) 與 pin 腳相對應的電壓準位來控制 pin 腳在特定的時間輸出高電位或低電位，以及電位狀態要維持多長時間。

MDIO 介面包含兩條線，MDC 與 MDIO，其中 MDIO 是雙向資料線而 MDC 為時鐘線。

大多數 NuMicro® 家族微控制器不支援 MDIO 介面，本範例代碼提供一種存取乙太網路 PHY 的解決方案，可廣泛應用於 NuMicro® 家族微控制器。

本範例代碼需另接外部硬體模組，包含一九暘電子的乙太網路 PHY(IP101GR)。

1.1 原理

1.1.1 MDIO 資料格式

MDIO 資料格式如圖 1-1 所示，欄位說明如表 1-1 所示。

Preamble	ST	OP	PHY_ADR	REG_ADR	TA	DATA
32-bit	2-bit	2-bit	5-bit	5-bit	2-bit	16-bit

圖 1-1 MDIO 資料格式

欄位	資料寬度	說明
Preamble	32-bit	32 個連續 2'1
ST	2-bit	幀開始標誌 2'01
OP	2-bit	操作碼 01 = 寫操作 10 = 讀操作
PHY_ADR	5-bit	PHY 位址
REG_ADR	5-bit	暫存器位址

TA	2-bit	狀態轉換時間
DATA	16-bit	MDIO 資料

表 1-1 MDIO 資料欄位介紹

1.1.2 MDIO 時序

使用 PSIO 控制 MDIO 裝置，使用者必須先設置時隙 (Slot)、檢查點 (Check point)與 pin 腳相對應的電壓準位來控制 pin 腳在特定的時間輸出高電位或低電位，以及電位狀態要維持多長時間。

PSIO0 CH3 (PC.2) pin 腳用於模擬 MDIO 波形輸出輸入腳位; PSIO0 CH1(PC.4) pin 腳用於模擬 MDC 波形輸出腳位。要注意的是，在讀操作中，程式必須先將 PSIO0 CH3 (PC.2) pin 腳配置為輸入模式，才能讀取資料。

圖 1-2 與圖 1-3 分別標示 PSIO 在讀/寫操作的相關配置。本範例代碼設定 MDC 為 500 kHz，其 MDC 時序計算如下所示：

- PSIO 時鐘源設定為 HIRC，並除頻為 8 MHz; 一個時隙的時間為 0.125 µs (= 1 / 8 MHz)。
- 設定時隙長度為 8，則 MDC 半個脈波寬度時間為 0.125 µs x 8 = 1 µs。

基於以上設定，PSIO0 CH1(PC.4) pin 腳即可產生 500 kHz 的方波波形，PSIO0 CH3 (PC.2) pin 腳僅須在檢查點 0 時設定為輸出或輸入緩存。

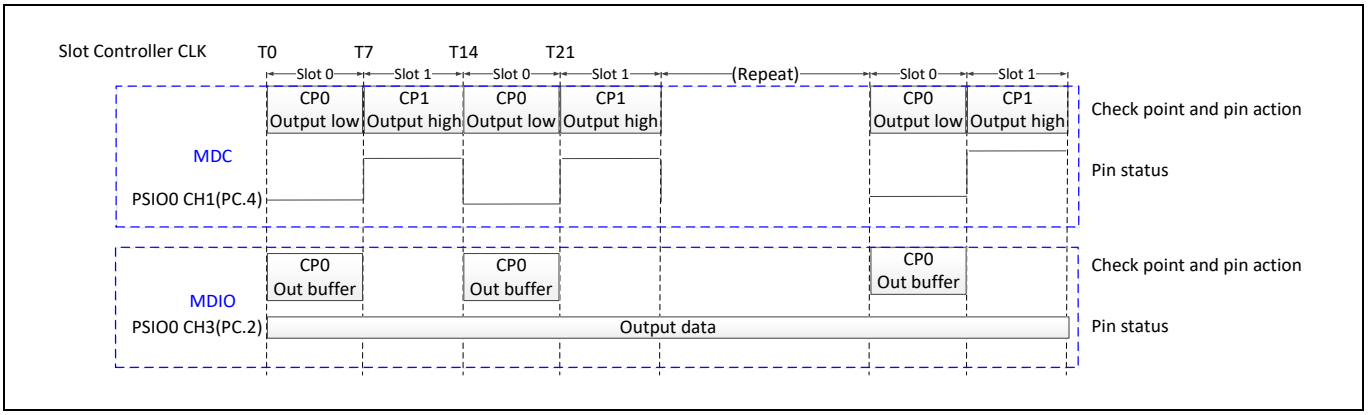

圖 1-2 PSIO 寫操作相關設定

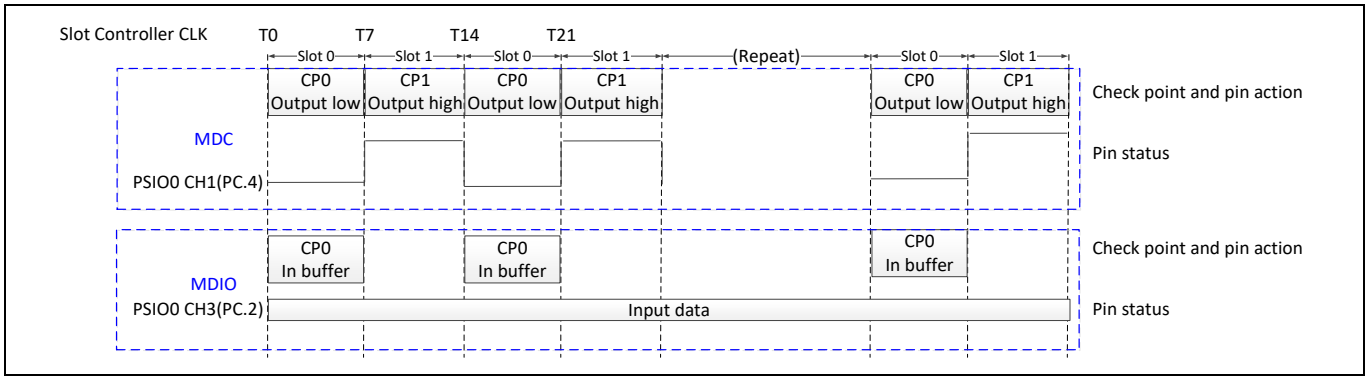

圖 1-3 PSIO 讀操作相關設定

1.2 執行結果

開啟電腦終端機軟體，設置 UART 傳輸速度為 115200bps，無校驗位元，數據長度為 8 位元，停止位元長度為 1 位元。執行程式後，終端機顯示操作指示。

圖 1-4 終端機螢幕截圖

本範例代碼執行編譯結果如圖 1-5 與圖 1-6 所示，分別對應 MDIO 寫操作和讀操作波形。演示將暫存器 0x04 配置為 0x0001 並進行驗證。

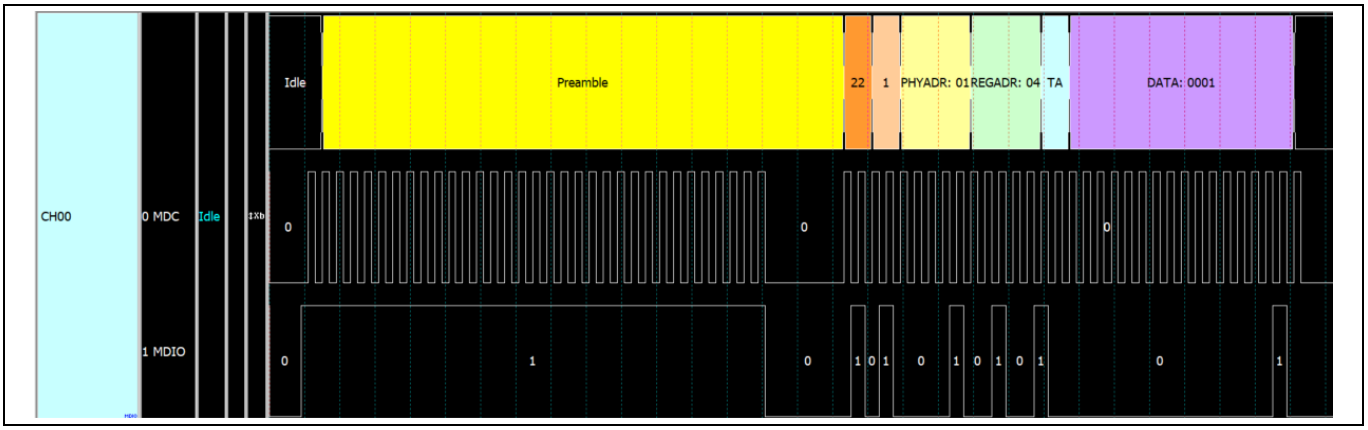

圖 1-5 MDIO 寫操作波形

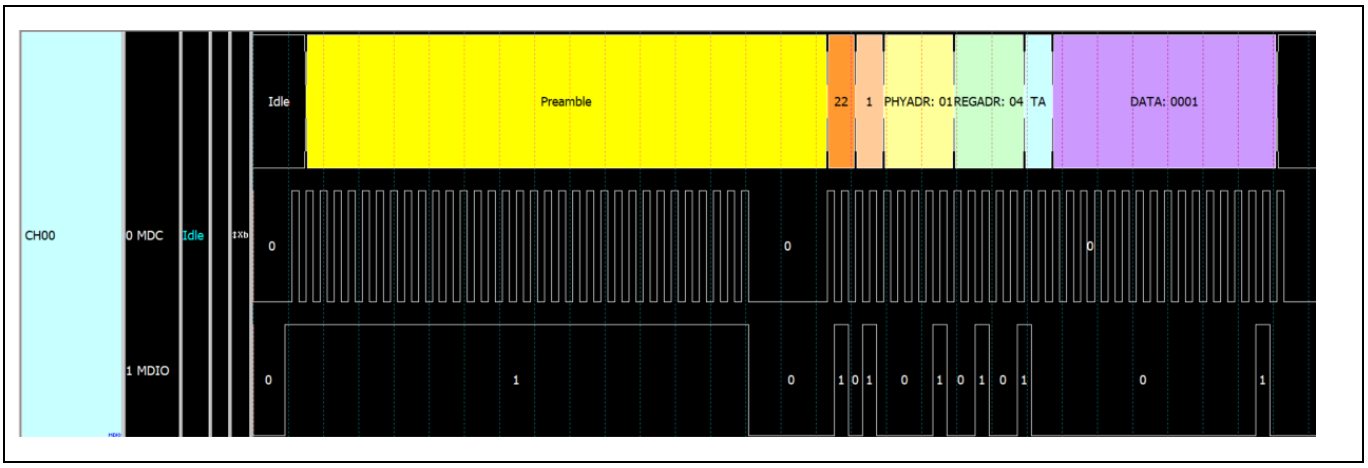

圖 1-6 MDIO 讀操作波形

2. 代碼介紹

寫操作與讀操作流程代碼位於 *main.c*。

寫操作

- 傳送 32 位元 Preamble。
- 傳送 ST (2 位元)、opcode(2 位元)、PHY 位址(5 位元)、暫存器位址(5 位元)、TA (2 位元)與 暫存器資料(16 位元)。

```
int main(void)
{
    …

    /* Prepare the MDIO package with 32-bit data, including 2-bit ST, 2-bit Op code, 5-bit
       PHY address, 5-bit Register address,  2-bit TA and 16-bit Data */
    u32TxData = (MDIO_ST_CAUSE_22 << MDIO_ST_OFFSET) | (MDIO_OP_W << MDIO_OP_OFFSET) |
                (MDIO_PHY_ADDR << MDIO_PHY_OFFSET) | (u32InputRegAdr << MDIO_REG_OFFSET) |
                (MDIO_TA << MDIO_TA_OFFSET) | u32InputData;

    /* Write Preamble field to Ethernet PHY */
    PSIO_MDIO_Write(&sConfig, &u32Preamble, 32);

    /* Write ST/Op code/PHY adress/TA fields to Ethernet PHY */
    PSIO_MDIO_Write(&sConfig, &u32TxData, 32);

    …
}
```

讀操作

- 傳送 32 位元 Preamble。
- 傳送 ST (2 位元)、opcode(2 位元)、PHY 位址(5 位元)、暫存器位址(5 位元)與 TA (2 位元)。
- 接收 16 位元資料。

```
int main(void)
{
    …

    /* Prepare the MDIO package with 32-bit data, including 2-bit ST, 2-bit Op code, 5-bit
       PHY address, 5-bit Register address and 2-bit TA */
    u32TxData = (MDIO_ST_CAUSE_22 << MDIO_ST_OFFSET) | (MDIO_OP_R << MDIO_OP_OFFSET) |
    (MDIO_PHY_ADDR << MDIO_PHY_OFFSET) | (u32InputRegAdr << MDIO_REG_OFFSET) | (MDIO_TA <<
    MDIO_TA_OFFSET);
```

```
    u32TxData = u32TxData >> 16;

    /* Write Preamble field to Ethernet PHY */
    PSIO_MDIO_Write(&sConfig, &u32Preamble, 32);

    /* Write ST/Op code/PHY adress/TA fields to Ethernet PHY */
    PSIO_MDIO_Write(&sConfig, &u32TxData, 16);

    /* Read data from Ethernet PHY */
    PSIO_MDIO_Read(&sConfig, &u32InputData);

    …
}
```

PSIO 初始化設定、寫操作與讀操作代碼位於 *MDIO_driver.c*。

使用 PSIO_MDIO_Init() 來設定每一個時隙 (Slot) 的寬度與檢查點 (Check point) 相對應的 pin 腳狀態。

```
void PSIO_MDIO_Init(S_PSIO_MDIO *psConfig)
{
    const S_PSIO_CP_CONFIG sMDCConfig =
    {

              /* Check Point0 */ /* Check Point1 */  /* Check Point2 */  /* Check Point3 */
/* Slot */   PSIO_SLOT0,        PSIO_SLOT1,         PSIO_SLOT_DISABLE,  PSIO_SLOT_DISABLE,
/* Action */ PSIO_OUT_LOW,      PSIO_OUT_HIGH,      PSIO_NO_ACTION,     PSIO_NO_ACTION,

              /* Check Point4 */ /* Check Point5 */  /* Check Point6 */  /* Check Point7 */
/* Slot */   PSIO_SLOT_DISABLE, PSIO_SLOT_DISABLE,  PSIO_SLOT_DISABLE,  PSIO_SLOT_DISABLE,
/* Action */ PSIO_NO_ACTION,    PSIO_NO_ACTION,     PSIO_NO_ACTION,     PSIO_NO_ACTION
    };

    /* Enable MDC pin and setting general configuration */
    PSIO_SET_GENCTL(PSIO, psConfig->u8MDC, PSIO_PIN_ENABLE, psConfig->u8SlotCtrl
                    , PSIO_OUTPUT_MODE, PSIO_LOW_LEVEL, PSIO_LOW_LEVEL);

    /* Set data order and MSB */
    PSIO_SET_ORDER(PSIO, psConfig->u8MDC, PSIO_MSB);
    PSIO_SET_ORDER(PSIO, psConfig->u8MDIO, PSIO_MSB);

    /* Set slot counter number */
    PSIO_SCSLOT_SET_SLOT(PSIO, psConfig->u8SlotCtrl, PSIO_SLOT0, 8);
    PSIO_SCSLOT_SET_SLOT(PSIO, psConfig->u8SlotCtrl, PSIO_SLOT1, 8);

    /* Set check point configuration */
    PSIO_SET_CP_CONFIG(PSIO, psConfig->u8MDC, &sMDCConfig);
}
```

使用 PSIO_MDIO_Write() 產生輸出波形，其時序可參考圖 1-2。

```
void PSIO_MDIO_Write(S_PSIO_MDIO *psConfig, uint32_t *pu32Data, uint32_t bit_length)
{
    const S_PSIO_CP_CONFIG sDataOutConfig =
```

```
    {
            /* Check Point0 */ /* Check Point1 */  /* Check Point2 */  /* Check Point3 */
/* Slot */  PSIO_SLOT0,         PSIO_SLOT_DISABLE,  PSIO_SLOT_DISABLE,  PSIO_SLOT_DISABLE,
/* Action */PSIO_OUT_BUFFER,    PSIO_NO_ACTION,     PSIO_NO_ACTION,     PSIO_NO_ACTION,

            /* Check Point4 */ /* Check Point5 */  /* Check Point6 */  /* Check Point7 */
/* Slot */   PSIO_SLOT_DISABLE, PSIO_SLOT_DISABLE,  PSIO_SLOT_DISABLE,  PSIO_SLOT_DISABLE,
/* Action */ PSIO_NO_ACTION,    PSIO_NO_ACTION,     PSIO_NO_ACTION,     PSIO_NO_ACTION
    };

    /* Enable MDIO pin and setting general configuration */
    PSIO_SET_GENCTL(PSIO, psConfig->u8MDIO, PSIO_PIN_ENABLE, psConfig->u8SlotCtrl
                    , PSIO_OUTPUT_MODE, PSIO_LOW_LEVEL, PSIO_LOW_LEVEL);

    PSIO_SET_CP_CONFIG(PSIO, psConfig->u8MDIO, &sDataOutConfig);

    /* Loop slot0~slot1 times */
    PSIO_SET_SCCTL(PSIO, psConfig->u8SlotCtrl, PSIO_SLOT0, PSIO_SLOT1, bit_length - 1,
PSIO_REPEAT_DISABLE);

    /* Set output data width */
    PSIO_SET_WIDTH(PSIO, psConfig->u8MDIO, 0, bit_length);

    /* Set output data */
    PSIO_SET_OUTPUT_DATA(PSIO, psConfig->u8MDIO, *pu32Data);

    /* Trigger slot controller */
    PSIO_START_SC(PSIO, psConfig->u8SlotCtrl);

    /* Wait for slot controller is not busy */
    while (PSIO_GET_BUSY_FLAG(PSIO, psConfig->u8SlotCtrl));
}
```

使用 PSIO_MDIO_Read()產生讀取波形，其時序可參考圖 1-3。

```
void PSIO_MDIO_Read(S_PSIO_MDIO *psConfig, uint32_t *pu32Data)
{
    uint32_t u32Data;

    const S_PSIO_CP_CONFIG sDataInConfig =
    {
            /* Check Point0 */ /* Check Point1 */  /* Check Point2 */  /* Check Point3 */
/* Slot */   PSIO_SLOT0,         PSIO_SLOT_DISABLE,  PSIO_SLOT_DISABLE,  PSIO_SLOT_DISABLE,
/* Action */ PSIO_IN_BUFFER,    PSIO_NO_ACTION,     PSIO_NO_ACTION,     PSIO_NO_ACTION,

            /* Check Point4 */ /* Check Point5 */  /* Check Point6 */  /* Check Point7 */
/* Slot */   PSIO_SLOT_DISABLE, PSIO_SLOT_DISABLE,  PSIO_SLOT_DISABLE,  PSIO_SLOT_DISABLE,
/* Action */ PSIO_NO_ACTION,    PSIO_NO_ACTION,     PSIO_NO_ACTION,     PSIO_NO_ACTION
    };

    /* Enable MDIO pin and setting general configuration */
    PSIO_SET_GENCTL(PSIO, psConfig->u8MDIO, PSIO_PIN_ENABLE, psConfig->u8SlotCtrl
                    , PSIO_INPUT_MODE, PSIO_LOW_LEVEL, PSIO_LOW_LEVEL);

    PSIO_SET_CP_CONFIG(PSIO, psConfig->u8MDIO, &sDataInConfig);
```

```
    /* Loop slot 0~slot 1 15 times */
    PSIO_SET_SCCTL(PSIO, psConfig->u8SlotCtrl, PSIO_SLOT0, PSIO_SLOT1, 15,
PSIO_REPEAT_DISABLE);

    /* Set input data width as 16 */
    PSIO_SET_WIDTH(PSIO, psConfig->u8MDIO, 16, 0);

    /* Set input data depth as 1 */
    PSIO_SET_INPUT_DEPTH(PSIO, psConfig->u8MDIO, PSIO_DEPTH1);

    /* Trigger slot controller */
    PSIO_START_SC(PSIO, psConfig->u8SlotCtrl);

    /* Wait for data buffer is full */
    while (!PSIO_GET_TRANSFER_STATUS(PSIO, PSIO_TRANSTS_INFULL0_Msk << (psConfig->u8MDIO *
4)));

    /* Read data from device */
    u32Data = PSIO_GET_INPUT_DATA(PSIO, psConfig->u8MDIO);
    *pu32Data = u32Data;
}
```

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_V3.02.006
- IDE 版本
  - Keil uVersion 5.28

3.2 硬體需求

- 電路元件
  - NuMaker-M252SD V1.2
- 線路示意圖
  - PSIO0_CH3 (PC.2) pin 腳連接到乙太網路 PHY MDIO pin 腳。
  - PSIO0_CH1 (PC.4) pin 腳連接到乙太網路 PHY MDC pin 腳。

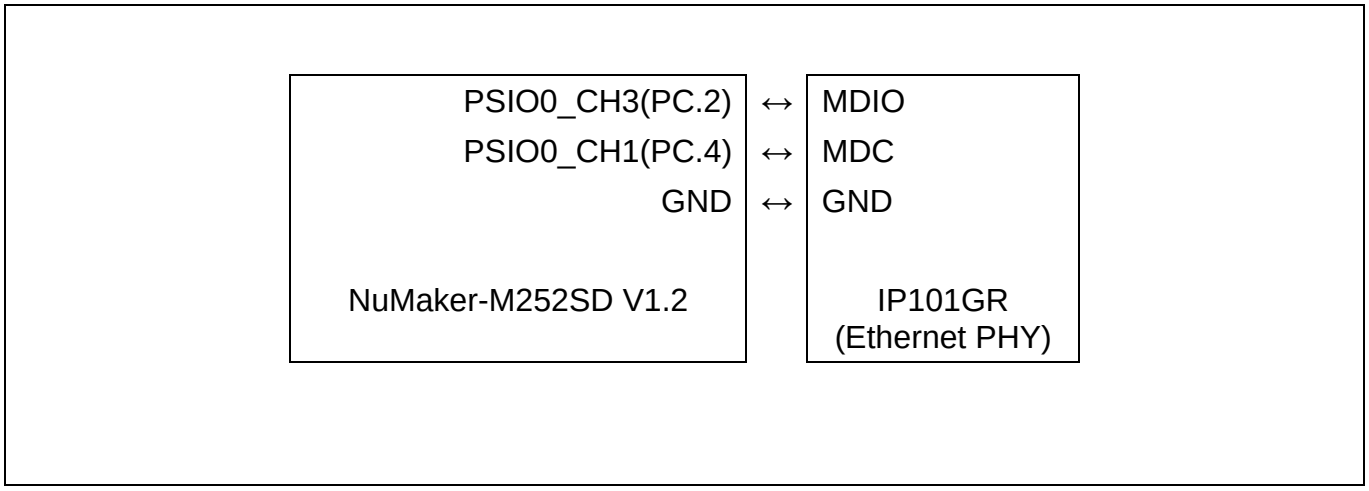

圖 3-1 腳位連接圖

4. 目錄資訊

EC_M252_PSIO_Emulate_MDIO _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 *M252_PSIO_Emulate_MDIO.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
2024.08.22	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.