讀取感測器DHT11的溼度和溫度資料

NuMicro® 32 位系列微控制器範例代碼介紹

文件資訊

代碼簡述	使用 NUC240 的 GPIO (PA.0)和 TIMER0 讀取 DHT11 的溼度和溫度資料
BSP 版本	NUC230/240 Series BSP CMSIS V3.01.001
開發平台	NuTiny-SDK-NUC240V 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 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 功能介紹

1.1 簡介

使用GPIO (PA.0)和TIMER0讀取DHT11的溼度和溫度資料。DHT11是個常見的數位濕度、溫度感測器。DHT11使用簡化的單總線通訊方式交換資料。資料長度為40位元，資料格式如所下表所示：

1st 8 位元	2nd 8 位元	3rd 8 位元	4th 8 位元	5th 8 位元
濕度整數資料	濕度小數資料	溫度整數資料	溫度小數資料	前4筆資料的校驗碼

藉由GPIO和TIMER0，使用者可以產生與DHT11通訊的波形。請從以下連結，參考DHT11的規格資料，了解更詳細的時序圖。
https://osepp.com/downloads/pdf/DHT11-Technical-Data-Sheet-Translated-Version.pdf

1.2 執行結果

每隔1.5秒讀取DHT11的濕度和溫度資料。結果如下圖所示：

```
CPU @ 71884800 Hz
+------------------------------------------------+
|    Read DHT11 data Sample Code                 |
|    Use GPIO(PA.0) and TIMER0 to read DHT11 data|
|    Please connect PA.0 to DHT11 data pin       |
+------------------------------------------------+

Humidity: 72.0, Temperature: 24.9
Humidity: 73.0, Temperature: 24.9
Humidity: 73.0, Temperature: 24.9
Humidity: 73.0, Temperature: 24.9
Humidity: 73.0, Temperature: 24.9
```

2 代碼介紹

發送開始訊號給 DHT11，使 DHT11 從低功耗模式切換到執行模式。

```
    /* The Quasi-bidirection mode could be used as input with pull-up enable */
    GPIO_SetMode(Data_Port, Data_PinMsk, GPIO_PMD_QUASI);

    /* Send out start signal to DHT11 */
    /* Set data pin level from high to low */
    Data_Pin = 1;
    Data_Pin = 0;
    /* Take at least 18 ms to ensure DHT11's detection of signal */
    CLK_SysTickDelay(18000);
    /* Pull up data pin */
    Data_Pin = 1;

    /* Record current timer counter value */
    u32_poll_time = TIMER_GetCounter(TIMER0);

    /* Wait response from DHT11 by pulling down data pin */
    while (1)
    {
        if (Data_Pin == 0)
        {
            break;
        }

        if ((TIMER_GetCounter(TIMER0) - u32_poll_time) > 1000)
            /* DHT11 doesn't response over 1 ms */
            return FALSE;
    }

    /* Wait until DHT11 pulls up data pin level */
    while (1)
    {
        if (Data_Pin == 1)
        {
            break;
        }
    }
```

```
    /* Send start signal successfully */
    return TRUE;
```

在 NUC240 送出開始訊號並且收到 DHT11 的回覆後，NUC240 開始讀取 DHT11 的資料。

```
    /* Init TIMER0 to measure high level time */
    TMR0_Init();

    /* Start communicate with DHT11 */
    if (DHT11_Reset() == TRUE)
    {
        /* Wait until DHT11 starts data transmission */
        while (Data_Pin == 1);

        /* A complete data transmission is 40 bits, DHT11 sends MSB first */
        /* 1st 8 bits is humidity integral data */
        g_u8Humidity_Int = 0;

        for (i = 0; i < 8; i++)
        {
            /* Shift current data for next bit */
            g_u8Humidity_Int = g_u8Humidity_Int << 1;

            /* Wait until DHT11 pull high data pin */
            while (Data_Pin == 0);

            /* Measure the high level time to determine the data bit */
            u32_prev_time = TIMER_GetCounter(TIMER0);

            while (Data_Pin == 1);

            if ((TIMER_GetCounter(TIMER0) - u32_prev_time) > Max_Data0_Time)
            {
                g_u8Humidity_Int |= 0x01;
            }
        }

        /* 2nd 8 bits is humidity decimal data */
        g_u8Humidity_Dec = 0;

        for (i = 0; i < 8; i++)
```

```
    {
        /* Shift current data for next bit */
        g_u8Humidity_Dec = g_u8Humidity_Dec << 1;

        /* Wait until DHT11 pull high data pin */
        while (Data_Pin == 0);

        /* Measure the high level time to determine the data bit */
        u32_prev_time = TIMER_GetCounter(TIMER0);

        while (Data_Pin == 1);

        if ((TIMER_GetCounter(TIMER0) - u32_prev_time) > Max_Data0_Time)
        {
            g_u8Humidity_Dec |= 0x01;
        }
    }

    /* 3rd 8 bits is temperature integral data */
    g_u8Temperature_Int = 0;

    for (i = 0; i < 8; i++)
    {
        /* Shift current data for next bit */
        g_u8Temperature_Int = g_u8Temperature_Int << 1;

        /* Wait until DHT11 pull high data pin */
        while (Data_Pin == 0);

        /* Measure the high level time to determine the data bit */
        u32_prev_time = TIMER_GetCounter(TIMER0);

        while (Data_Pin == 1);

        if ((TIMER_GetCounter(TIMER0) - u32_prev_time) > Max_Data0_Time)
        {
            g_u8Temperature_Int |= 0x01;
        }
    }

    /* 4th 8 bits is temperature decimal data */
```

```
    g_u8Temperature_Dec = 0;

    for (i = 0; i < 8; i++)
    {
        /* Shift current data for next bit */
        g_u8Temperature_Dec = g_u8Temperature_Dec << 1;

        /* Wait until DHT11 pull high data pin */
        while (Data_Pin == 0);

        /* Measure the high level time to determine the data bit */
        u32_prev_time = TIMER_GetCounter(TIMER0);

        while (Data_Pin == 1);

        if ((TIMER_GetCounter(TIMER0) - u32_prev_time) > Max_Data0_Time)
        {
            g_u8Temperature_Dec |= 0x01;
        }
    }

    /* 5th 8 bits is checksum */
    u8Checksum = 0;

    for (i = 0; i < 8; i++)
    {
        /* Shift current data for next bit */
        u8Checksum = u8Checksum << 1;

        /* Wait until DHT11 pull high data pin */
        while (Data_Pin == 0);

        /* Measure the high level time to determine the data bit */
        u32_prev_time = TIMER_GetCounter(TIMER0);

        while (Data_Pin == 1);

        if ((TIMER_GetCounter(TIMER0) - u32_prev_time) > Max_Data0_Time)
        {
            u8Checksum |= 0x01;
        }
```

```
        }

        /* Stop TIMER0 */
        TIMER_Stop(TIMER0);

        /* Compare received data and checksum */
        if ((uint8_t)(g_u8Humidity_Int + g_u8Humidity_Dec + g_u8Temperature_Int +
g_u8Temperature_Dec) == u8Checksum)
        {
            return TRUE;
        }
        else
        {
            printf("Checksum error!\n");

            return FALSE;
        }
    }
    else
    {
        /* Stop TIMER0 */
        TIMER_Stop(TIMER0);

        printf("DHT11 is no response.\n");

        return FALSE;
    }
```

NUC240 每隔 1.5 秒讀取 DHT11 的溼度和溫度資料。

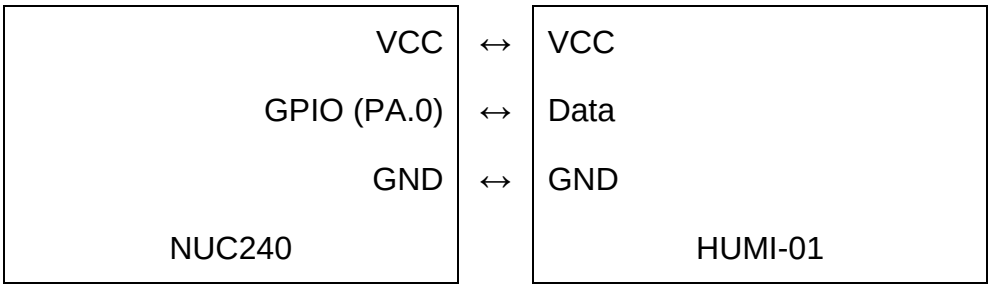
```
while (1)
    {
        /* Delay 1.5 seconds */
        Delay_mS(1500);

        if (DHT11_Read() == TRUE)
        {
            printf("Humidity: %d.%d, ", g_u8Humidity_Int, g_u8Humidity_Dec);
            printf("Temperature: %d.%d\n", g_u8Temperature_Int, g_u8Temperature_Dec);
        }
    }
```

3 軟體與硬體環境

- **軟體環境**
  - BSP 版本
    - NUC230/240 Series BSP CMSIS V3.01.001
  - IDE 版本
    - Keil uVersion 5.26
- **硬體環境**
  - 電路元件
    - NuTiny-SDK-NUC240V V1.2
    - 濕度和溫度感測器 DHT11 (模組 HUMI-01)
  - 示意圖

NUC240 使用 GPIO (PA.0)連接模組 HUMI-01 的 Data 腳位。



4 目錄資訊

- EC_NUC240_Read_DHT11_Data_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 — Source file of example code

5 如何執行範例程式

1. 根據目錄資訊章節進入 ExampleCode 路徑中的 KEIL 資料夾，雙擊 NUC240_Read_DHT11_Data.uvproj。
2. 進入編譯模式介面
   a. 編譯
   b. 下載代碼至記憶體
   c. 進入 / 離開除錯模式
3. 進入除錯模式介面
   a. 執行代碼

6 修訂紀錄

Date	Revision	Description
June 26, 2019	1.00	1. 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.