Read Sensor DHT11 Humidity and Temperature Data

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This code uses NUC240 GPIO (PA.0) and TIMER0 to read DHT11 humidity and temperature data.
BSP Version	NUC230/240 Series BSP CMSIS V3.01.001
Hardware	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 Function Description

1.1 Introduction

This example code uses GPIO (PA.0) and TIMER0 to read DHT11 humidity and temperature data. DHT11 is a common digital humidity and temperature sensor. DHT11 uses a simplified single-bus communication to exchange data. The data length is 40 bits and the format is shown below table:

1st 8 bits	2nd 8 bits	3rd 8 bits	4th 8 bits	5th 8 bits
Humidity integral data	Humidity decimal data	Temperature integral data	Temperature decimal data	Checksum of first 4 data

By using GPIO and TIMER0, user can generate the waveform to communicate with DHT11. For more detail data timing diagram, please refer to the DHT11 datasheet from below link.
https://osepp.com/downloads/pdf/DHT11-Technical-Data-Sheet-Translated-Version.pdf.

1.2 Demo Result

This example code reads humidity and temperature data from DHT every 1.5 seconds. The result is shown below picture:

```
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 Code Description

Send a start signal to DHT11 to make DHT11 change from the low-power-consumption mode to the running-mode.

```
    /* 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;
```

After NUC240 sends start signal and receives response from DHT11, NUC240 begins to read data from DHT11.

```
    /* 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 reads humidity and temperature data from DHT11 every 1.5 seconds.

```
    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 Software and Hardware Environment

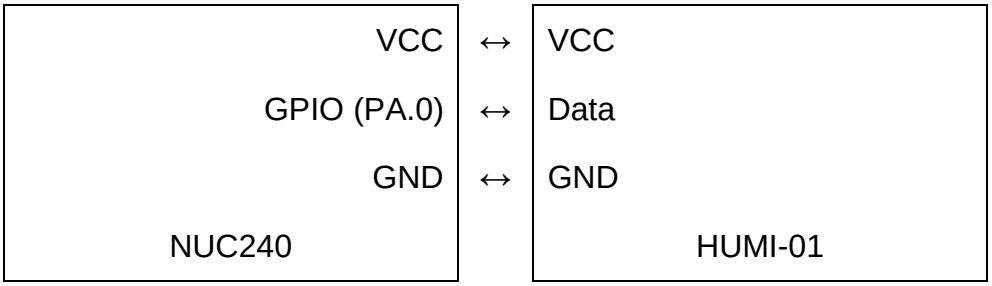
Software Environment

- BSP version
  - NUC230/240 Series BSP CMSIS V3.01.001
- IDE version
  - Keil uVersion 5.26

Hardware Environment

- Circuit components
  - NuTiny-SDK-NUC240V V1.2
  - Humidity and temperature sensor DHT11 (module HUMI-01)
- Diagram

NUC240 uses GPIO (PA.0) to connect with module HUMI-01 Data pin.



4 Directory Information

- 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 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click NUC240_Read_DHT11_Data.uvproj.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
June 26, 2019	1.00	1. Initially issued.

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.