M263 I²C驅動TDK CH101模組

NuMicro® 32 位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 M263 I²C 驅動 TDK CH101，實現測距功能
BSP 版本	M263_Series_BSP_CMSIS_V3.00.003
開發平台	NuMaker-M263KI V1.1

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

本示例代碼使用M263的I²C介面驅動CH101模組實現測距的功能。

CH101是一種超聲測距模組，可對距離最大1.2m的目標提供準確的距離測量。與其他類型的ToF測距儀（例如紅外（IR）感測器）不同，CH101不受物體顏色或透明度的影響，並且可在所有照明條件下工作。

本示例代碼中使用了TDK的EV_MOD_CH101-03-01模組。該模組板集成了CH101超聲波感測器設備，全向聲學外殼元件，電容和FPC / FFC連接器，通過I²C介面與主控MCU通信，可以在4 cm至1.2m的測距範圍內執行音高捕捉和脈衝回波測距。

本文部分引用了TDK 網站關於CH101的描述介紹及資料。

1.1 原理

CH101使用壓電微機械超聲換能器（PMUT）將聲音的短脈衝發射到空氣中。在擊中物體時，這些波朝PMUT反射，使其振動並產生電信號。通過內置的應用專用積體電路（ASIC）測量，聲波從發出到返回PMUT的聲音所需的時間，稱為飛行時間（ToF）。使用聲音速度（室溫343米/秒），系統可以確定與物件的距離。

測距公式如下：

$$Range\ (R) = C\ x \frac{ToF}{2}$$

C(聲音速度) = 343 m/s。

CH101可通過I²C介面對其進行控制和讀取，讀寫框圖如下圖所示：

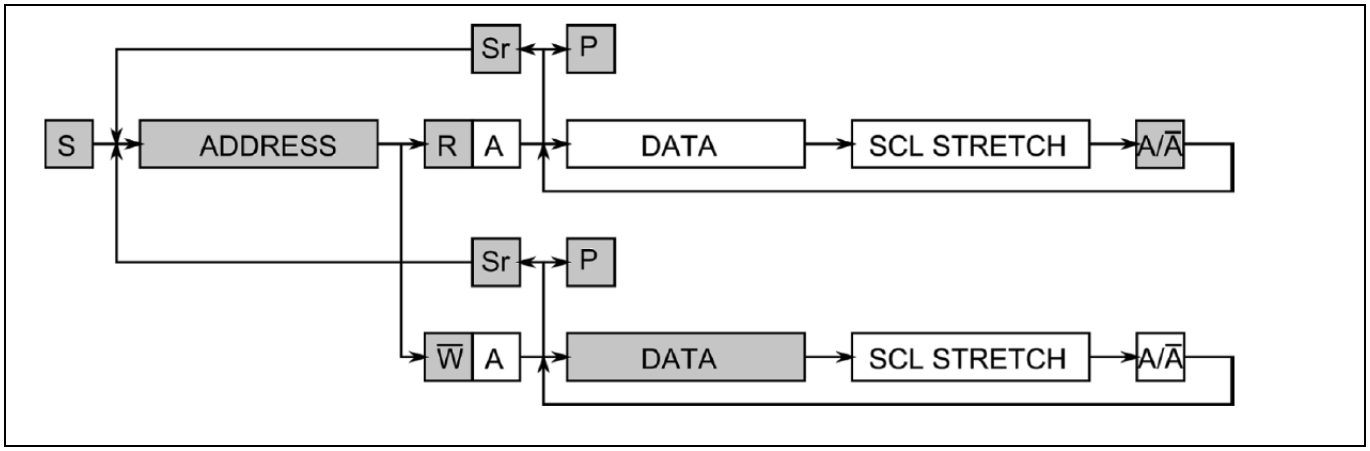

圖 1-1 讀寫邏輯框圖

關於 CH101 模組的詳細資訊，請參考的資料手冊及應用手冊。網址如下:

https://invensense.tdk.com/products/ch101/#documentation

1.2 執行結果

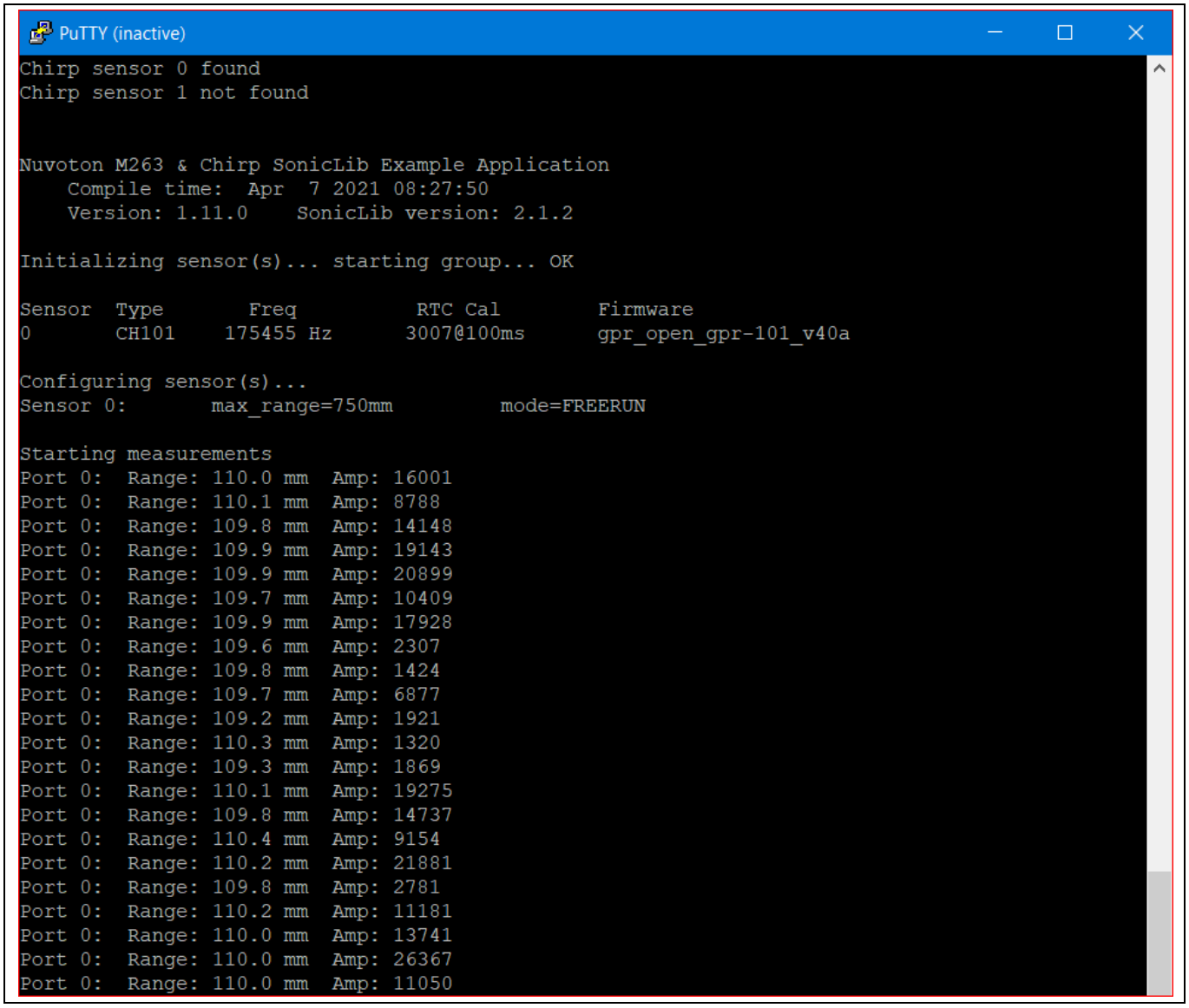

圖 1-2 執行結果

2 代碼介紹

晶片初始化，並查詢 CH101 模組是否接在了開發板上：

```
void chbsp_board_init(ch_group_t *grp_ptr) {

    /* Make local copy of group pointer */
    sensor_group_ptr = grp_ptr;

    /* Initialize group descriptor */
    grp_ptr->num_ports = CHBSP_MAX_DEVICES;
    grp_ptr->num_i2c_buses = CHBSP_NUM_I2C_BUSES;
    grp_ptr->rtc_cal_pulse_ms = CHBSP_RTC_CAL_PULSE_MS;

    /* Unlock protected registers */
    SYS_UnlockReg();
    /* Init System, IP clock and multi-function I/O. */
    SYS_Init();
    /* Lock protected registers */
    SYS_LockReg();

    /* Init I2C0 */
    I2C0_Init();

    /* Configure UART0: 921600, 8-bit word, no parity bit, 1 stop bit. */
    UART_Open(UART0, 921600);

    ext_int_init();

    /* Probe I2C bus to find connected sensor(s) */
    find_sensors();
}
```

軟體設置模組為 CH_MODE_FREERUN 模式，由模組內部 RTC 產生 100ms 時間間隔，通過 INT 引腳通知 M263 通過 I²C 讀取相關寄存器的值。

```
  /*
   * handle_data_ready() - get and display data from all sensors
   *
   * This routine is called from the main() loop after all sensors have
```

```
 * interrupted. It shows how to read the sensor data once a measurement is
 * complete.  This routine always reads out the range and amplitude, and
 * optionally will read out the amplitude data or raw I/Q for all samples
 * in the measurement.
 *
 * See the comments in app_config.h for information about the amplitude data
 * and I/Q readout build options.
 *
 */
static uint8_t handle_data_ready(ch_group_t *grp_ptr) {
  uint8_t  dev_num;
  int      num_samples = 0;
  uint8_t  ret_val = 0;

  /* Read and display data from each connected sensor
   *   This loop will write the sensor data to this application's "chirp_data"
   *   array.  Each sensor has a separate chirp_data_t structure in that
   *   array, so the device number is used as an index.
   */

  for (dev_num = 0; dev_num < ch_get_num_ports(grp_ptr); dev_num++) {
      ch_dev_t *dev_ptr = ch_get_dev_ptr(grp_ptr, dev_num);

      if (ch_sensor_is_connected(dev_ptr)) {

          /* Get measurement results from each connected sensor
           *   For sensor in transmit/receive mode, report one-way echo
           *   distance,  For sensor(s) in receive-only mode, report direct
           *   one-way distance from transmitting sensor
           */

          if (ch_get_mode(dev_ptr) == CH_MODE_TRIGGERED_RX_ONLY) {
              chirp_data[dev_num].range = ch_get_range(dev_ptr,
                                                       CH_RANGE_DIRECT);
          } else {
              chirp_data[dev_num].range = ch_get_range(dev_ptr,
                                                       CH_RANGE_ECHO_ONE_WAY);
          }

          if (chirp_data[dev_num].range == CH_NO_TARGET) {
              /* No target object was detected - no range value */
```

```
                chirp_data[dev_num].amplitude = 0;  /* no updated amplitude */

                printf("Port %d:          no target found        ", dev_num);

            } else {
                /* Target object was successfully detected (range available) */

                 /* Get the new amplitude value - it's only updated if range
                  * was successfully measured.  */
                chirp_data[dev_num].amplitude = ch_get_amplitude(dev_ptr);

                printf("Port %d:  Range: %0.1f mm  Amp: %u  ", dev_num,
                        (float) chirp_data[dev_num].range/32.0f,
                        chirp_data[dev_num].amplitude);
            }

            /* Store number of active samples in this measurement */
            num_samples = ch_get_num_samples(dev_ptr);
            chirp_data[dev_num].num_samples = num_samples;

            /* Optionally read amplitude values for all samples */
#ifdef READ_AMPLITUDE_DATA
            uint16_t     start_sample = 0;
            ch_get_amplitude_data(dev_ptr, chirp_data[dev_num].amp_data,
                                  start_sample, num_samples, CH_IO_MODE_BLOCK);

#ifdef OUTPUT_AMPLITUDE_DATA
            printf("\n");
            for (uint8_t count = 0; count < num_samples; count++) {

                printf("%d\n",  chirp_data[dev_num].amp_data[count]);
            }
#endif
#endif

            /* Optionally read raw I/Q values for all samples */
#ifdef READ_IQ_DATA
            display_iq_data(dev_ptr);
#endif
            /* If more than 2 sensors, put each on its own line */
```

```
            if (num_connected_sensors > 2) {
                printf("\n");
            }
        }
    }
    printf("\n");

    return ret_val;
}
```

3 軟體與硬體需求

● 軟體需求

- ■ BSP 版本
  - ◆ M261_Series_BSP_CMSIS_V3.00.003
- ■ IDE 版本
  - ◆ Keil uVersion 5.33

● 硬體需求

- ■ 電路元件
  - ◆ NuMaker-M263KI
  - ◆ EV_MOD_CH101-03-01
- ■ 示意圖

M263KI 通過 I²C 介面與 EV_MOD_CH101-03-01 模組連接。

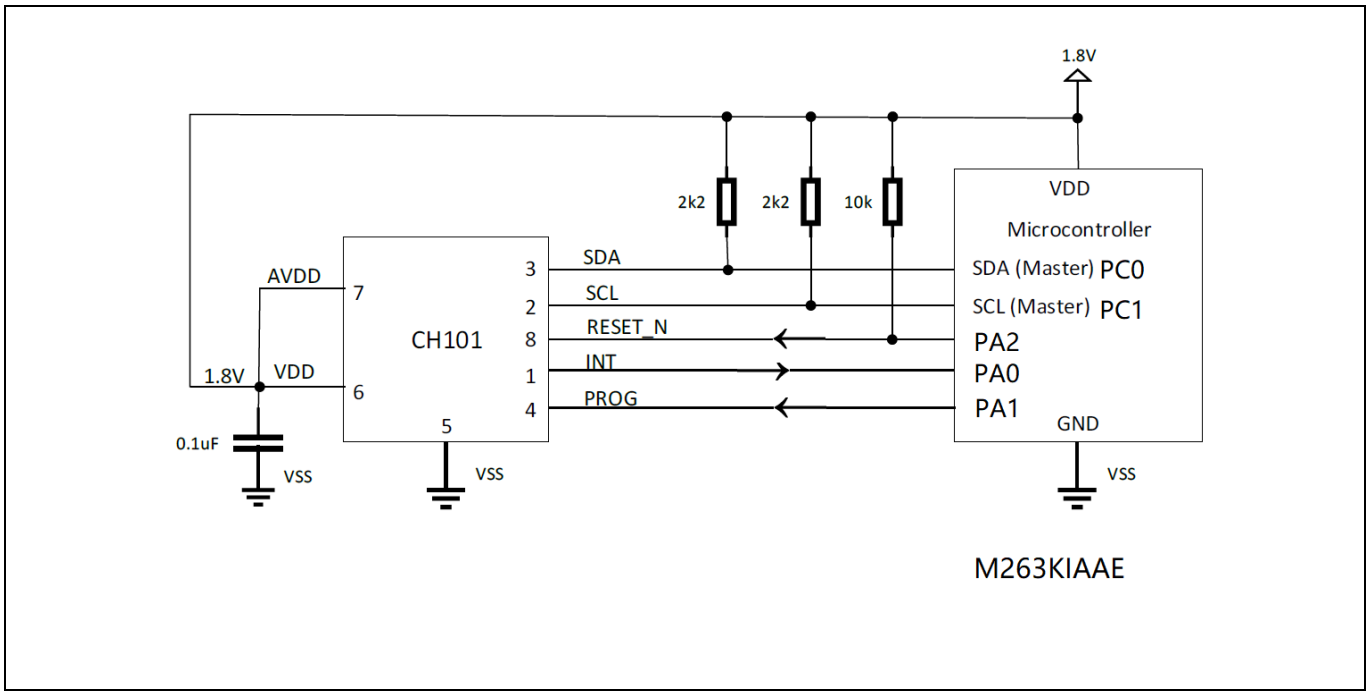

圖 3-1 引腳連接

CH101 模組工作電壓為 1.8V，故 NuMaker-M263KI 開發板為晶片 M263 供電電壓必須更換為 1.8V。可以改變電路板的電阻以使 Nulink2-Me 輸出 1.8V 電壓。

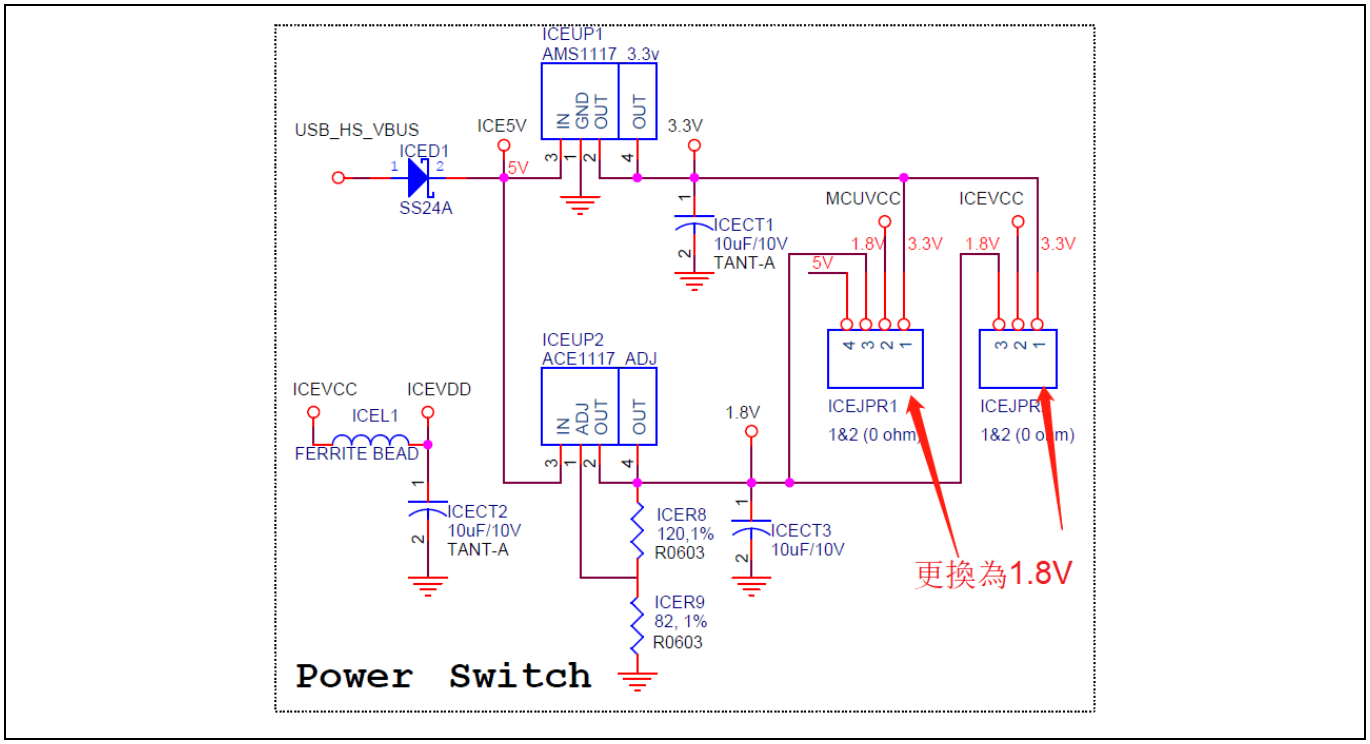

圖 3-2 電源轉換

4 目錄資訊

EC_ M263_I2C_TDK_CH101_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
ThirdParty	
Chirpmicro	Chirp SonicLib sensor API

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 *M263_I2C_TDK_CH101.uvprojx*。
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
2021.05.06	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.