M263 I²C驱动TDK CH101模块

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

应用简述	本范例代码使用 M263 I²C 驱动 TDK CH101，实现测距功能
BSP 版本	M263_Series_BSP_CMSIS_V3.00.003
开发平台	NuMaker-M263KI V1.1

1 概述

本示例代码使用M263的I²C接口驱动CH101模块实现测距的功能。

CH101是一种超声测距模块，可对距离最大1.2m的目标提供准确的距离测量。与其他类型的ToF测距仪（例如红外（IR）传感器）不同，CH101不受物体颜色或透明度的影响，并且可在所有照明条件下工作。

本示例代码中使用了TDK的EV_MOD_CH101-03-01模组。该模块板集成了CH101超声波传感器设备，全向声学外壳组件，电容和FPC / FFC连接器，通过I²C接口与主控MCU通信，可以在4 cm至1.2m的测距范围内执行音高捕捉和脉冲回波测距。

本文部分引用了TDK 网站关于CH101的描述介绍及资料。

1.1 原理

CH101使用压电微机械超声换能器（PMUT）将声音的短脉冲发射到空气中。在击中物体时，这些波朝PMUT反射，使其振动并产生电信号。通过内置的应用专用集成电路（ASIC）测量，声波从发出到返回PMUT的声音所需的时间，称为飞行时间（ToF）。使用声音速度（室温343米/秒），系统可以确定与对象的距离。

测距公式如下：

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

C(声音速度) = 343 m/s。

CH101可通过I²C接口对其进行控制和读取，读写框图如下图所示：

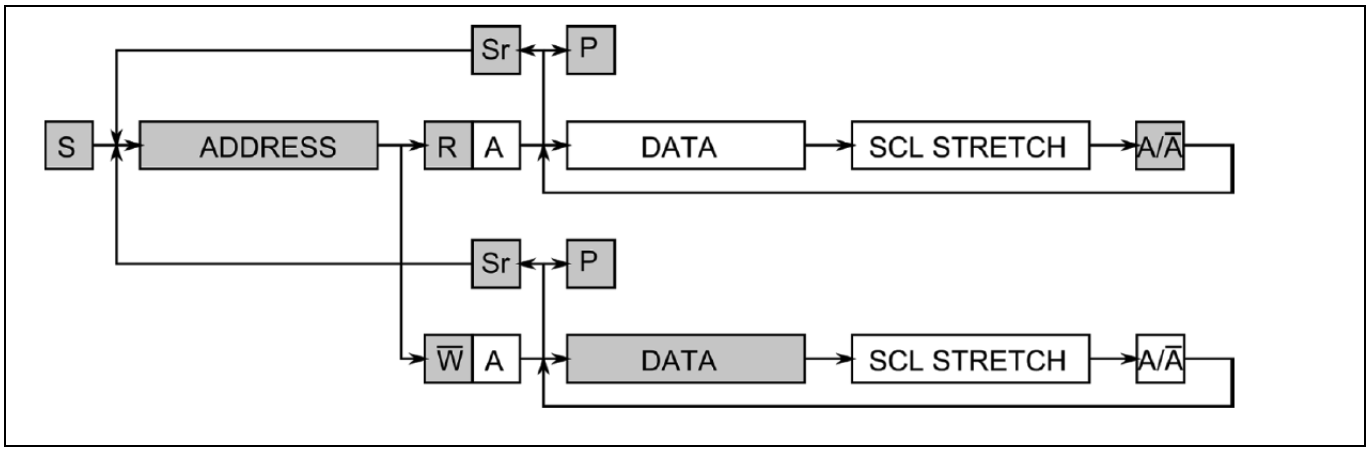

图 1-1 读写逻辑框图

关于 CH101 模块的详细信息，请参考的数据手册及应用手册。网址如下:

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

1.2 执行结果

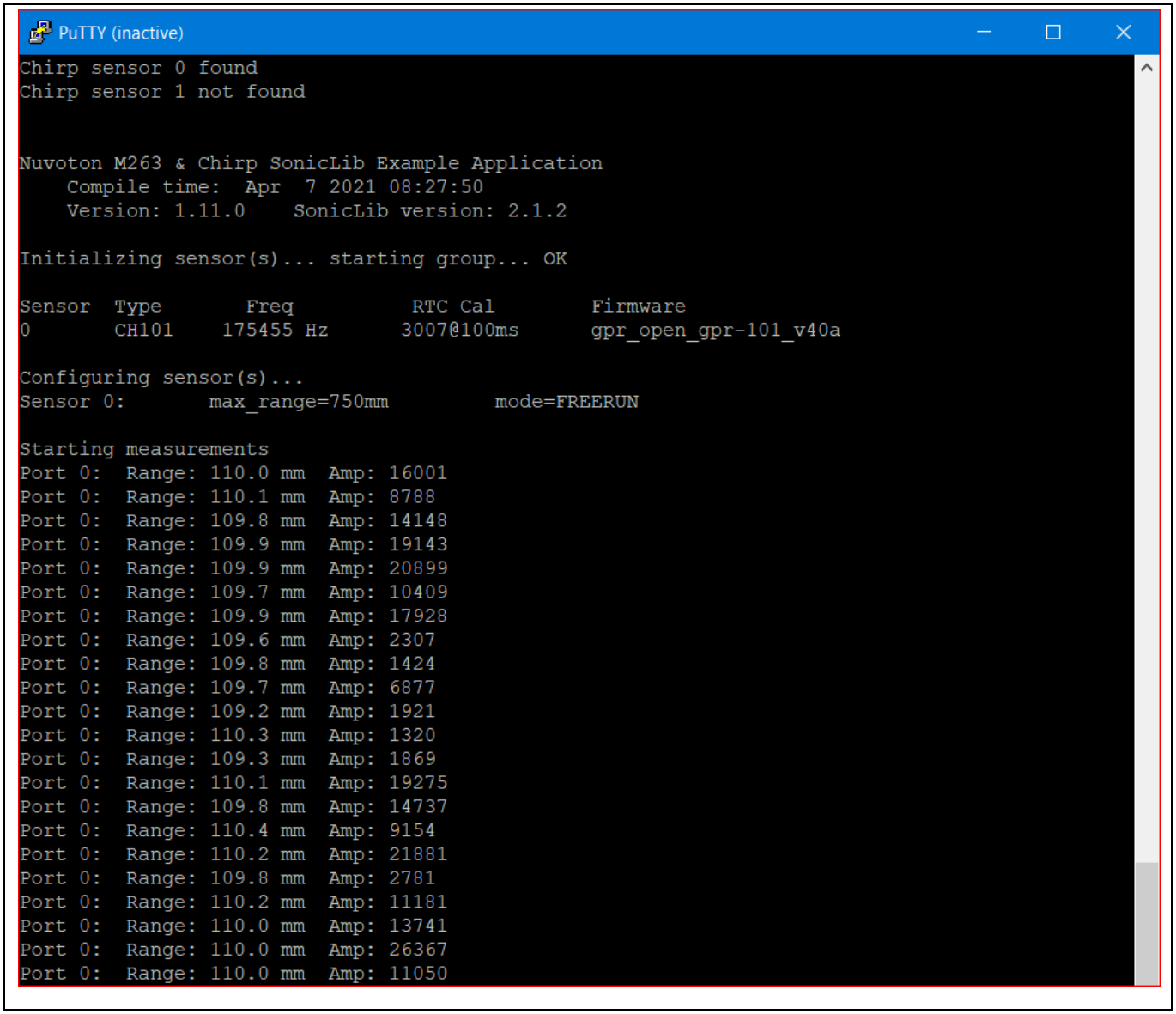

图 1-2 执行结果

2 代码介绍

芯片初始化，并查询 CH101 模块是否接在了开发板上：

```
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();
}
```

软件设置模块为 CH_MODE_FREERUN 模式，由模块内部 RTC 产生 100ms 时间间隔，通过 INT 引脚通知 M263 通过 I²C 读取相关寄存器的值。

```
  /*
   * 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 软件与硬件需求

- 软件需求
  - BSP 版本
    - M261_Series_BSP_CMSIS_V3.00.003
  - IDE 版本
    - Keil uVersion 5.33
- 硬件需求
  - 电路组件
    - NuMaker-M263KI
    - EV_MOD_CH101-03-01
  - 示意图

M263KI 通过 I²C 接口与 EV_MOD_CH101-03-01 模组连接。

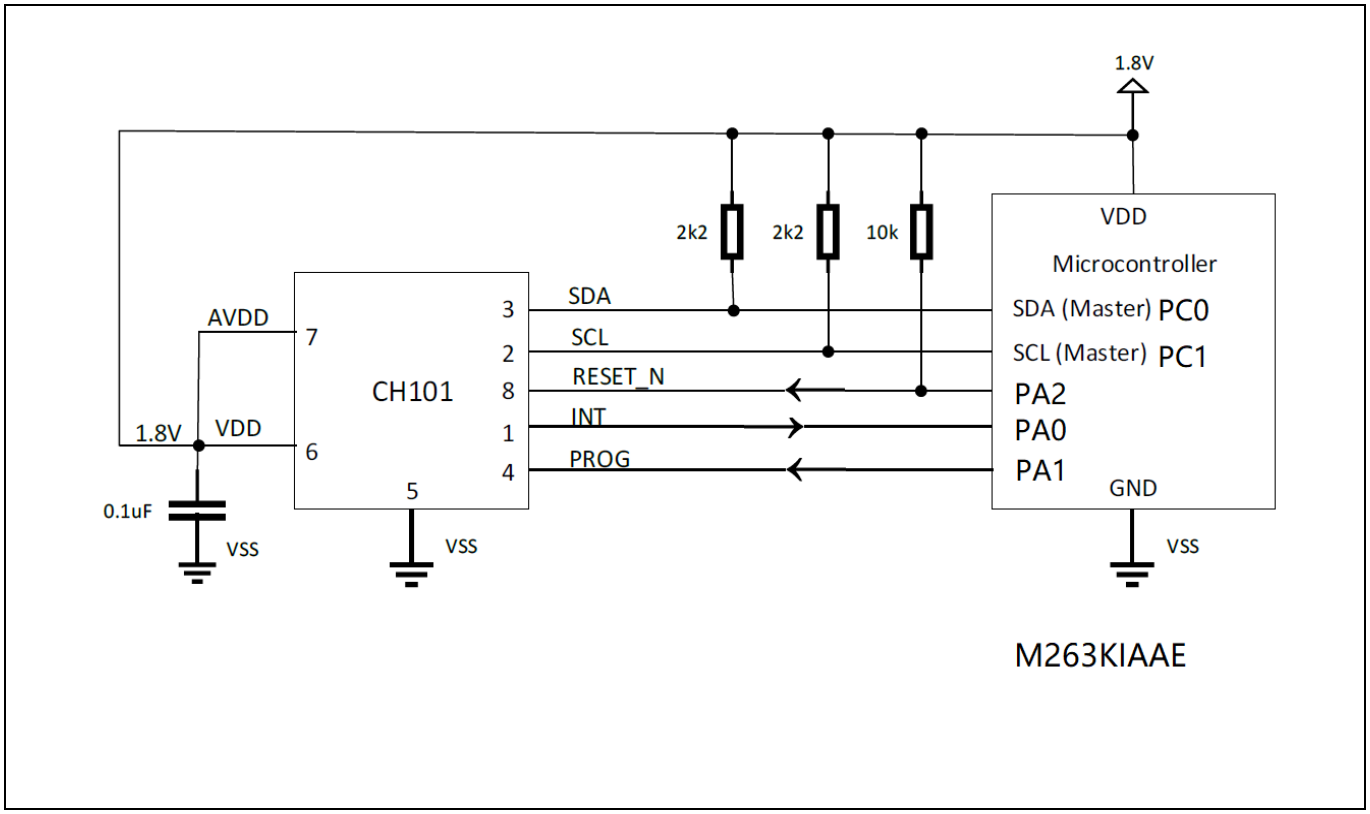

图 3-1 引脚连接

CH101 模块工作电压为 1.8V，故 NuMaker-M263KI 开发板为芯片 M263 供电电压必须更换为 1.8V。可以改变电路板的电阻以使 Nulink2-Me 输出 1.8V 电压。

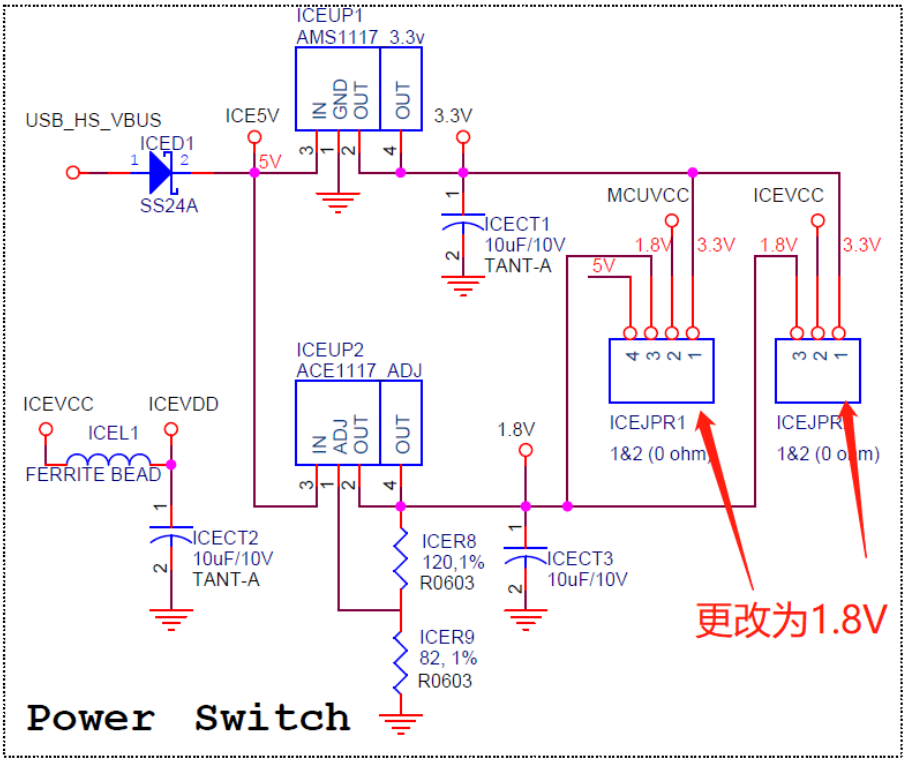

图 3-2 电源转换

4 目录信息

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

图 4-1 目录信息

5 范例程序执行

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 *M263_I2C_TDK_CH101.uvprojx*。
2. 进入编译模式界面
   a. 编译
   b. 下载代码至内存
   c. 进入 / 离开仿真模式
3. 进入仿真模式界面
   a. 执行代码

6 修订纪录

Date	Revision	Description
2021.05.06	1.00	1. 初始发布.

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.