使用 NADC24 与 M254 实现工业电子秤

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

应用简述	此范例程序是使用 NADC24 和 M254 实现工业电子秤
BSP 版本	M251_M252_M254_M256_M258_Series_BSP_CMSIS_V3.02.005
开发平台	NADC24_M254_Weighing_Scale_V1.0

1. 概述

本范例程序使用 NADC24 模拟数字转换器和 M254 配合不同的称重传感器，快速开发称重秤、计数秤、台秤、冷媒秤等应用。

1.1 原理

称重传感器是由应变片和电桥电路组成的传感器。当传感器受到拉力或压力时，会产生与力成比例的电压。使用者透过 NADC24（24bit ADC）准确读取电压，转换为数值，然后透过 SPI 接口传输到 M254 单芯片进行校正和计算，计算目前重量值并显示在液晶屏幕上。

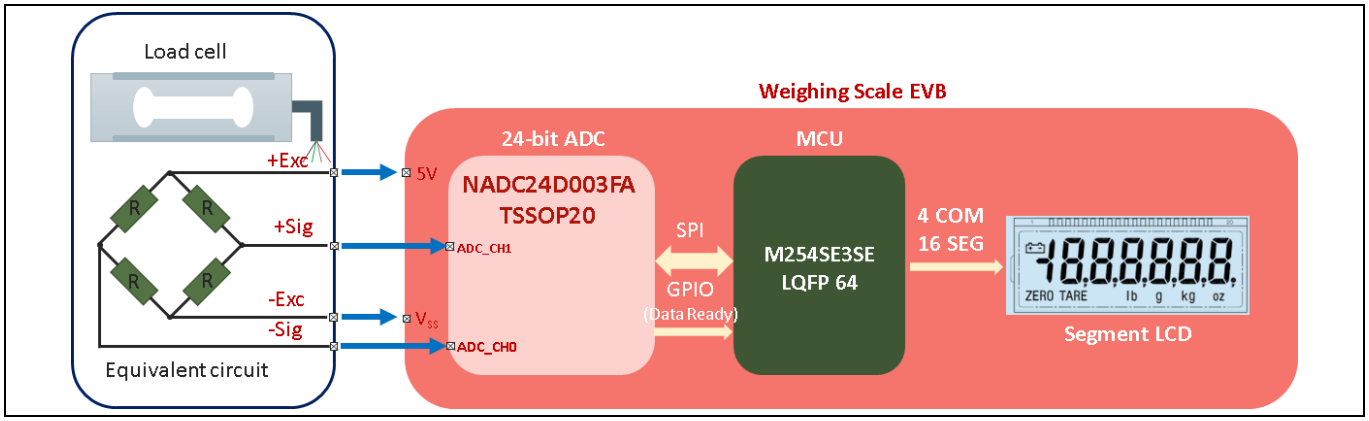


图 1-1 工业电子秤方块图

1.2 NADC24 (24-Bit Delta-Sigma ADC) 概述

NADC24 是一款高精度 24 位 ΔΣ 模拟数字转换器 (ADC)。它用于读取称重传感器的讯号。 使用者可以设定可程序增益放大器（PGA）、取樣速度、内部参考电压、数字滤波器（FIR）等功能，以获得稳定可靠的数据，并为 MCU 提供重量计算。

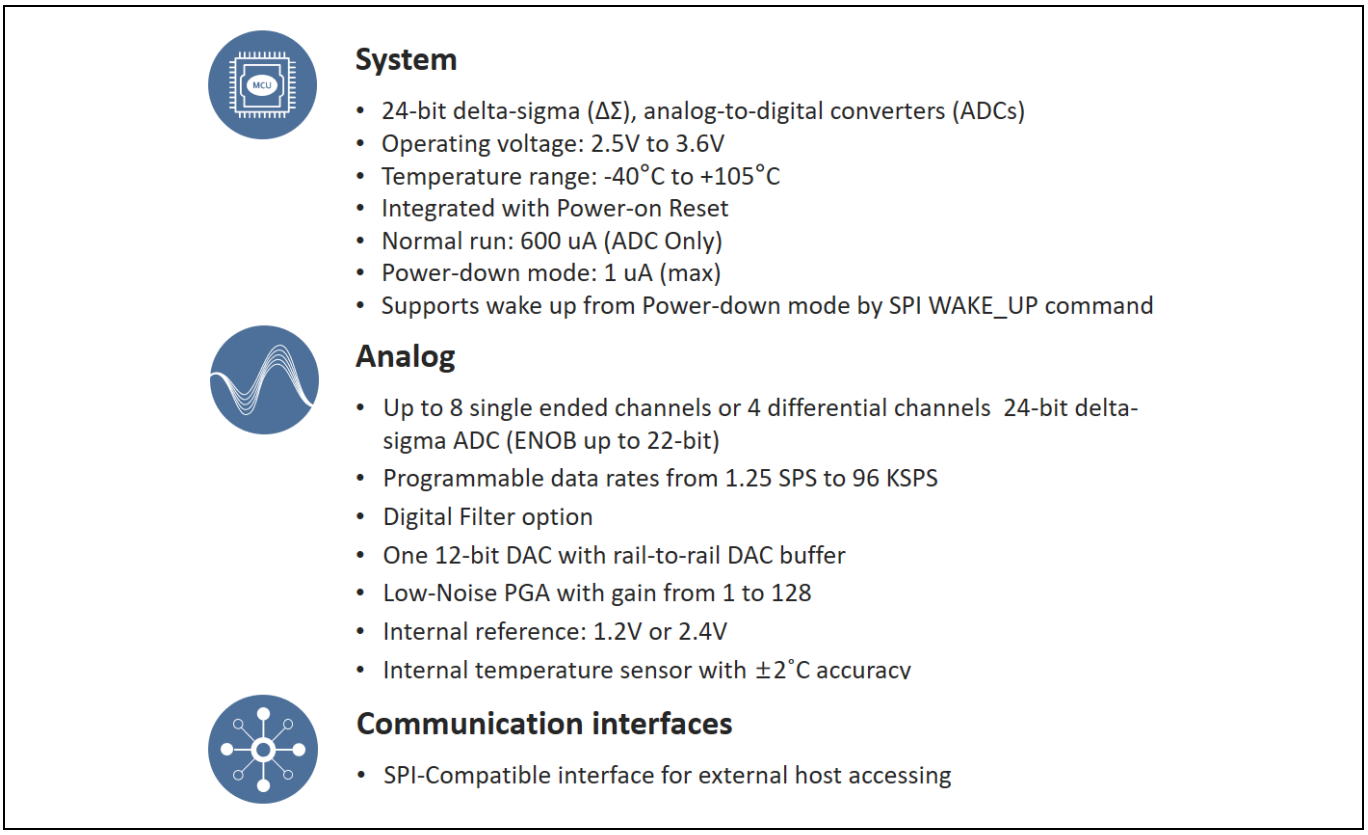
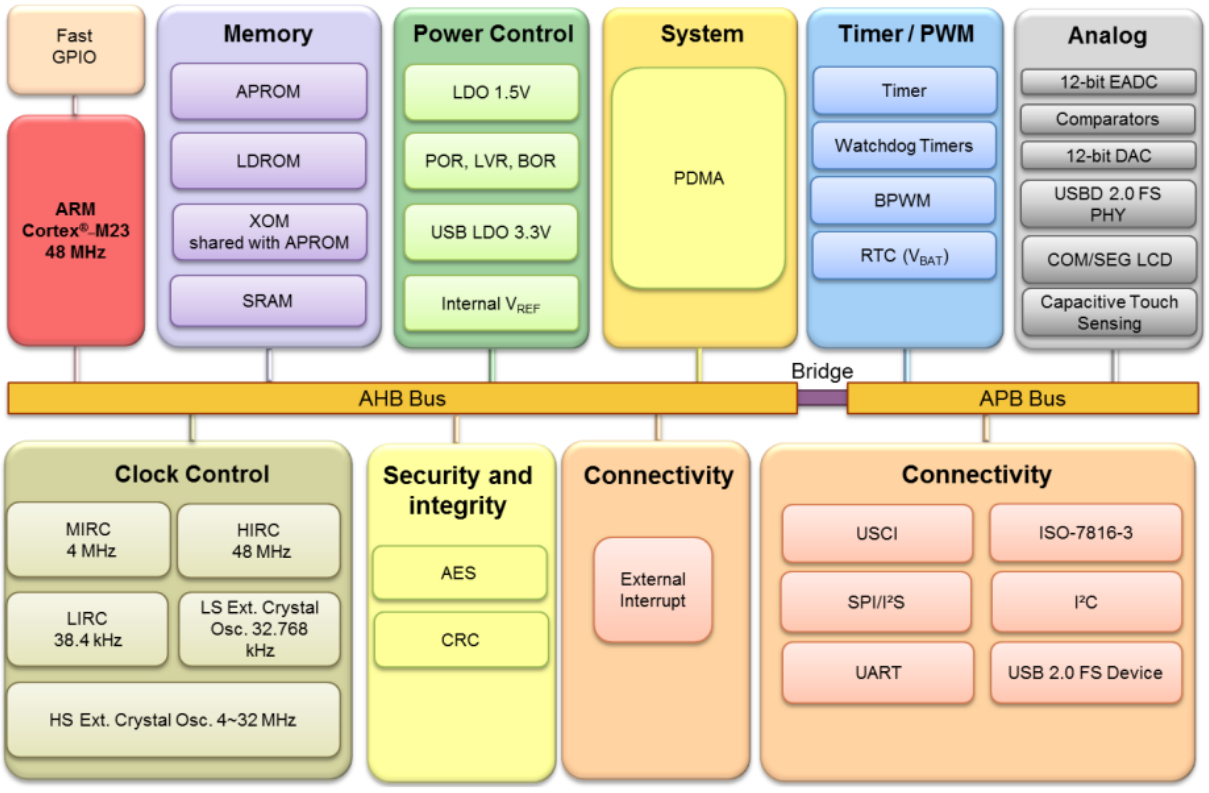


图 1-2 NADC24 規格

1.3 M254 概述

M254 以 Arm Cortex-M23 为核心，工作频率 48 MHz，Flash 128KB，SRAM 16KB，COM/SEG LCD Driver 以及 UART、I2C、SPI 等外围界面。可透过 SPI 接口读取 NADC24 转换后的数值，进行计算后，就可以透過按壓 UNIT 按鍵在段碼 LCD 上依序切換四种不同的重量单位，克（g）、公斤（kg）、磅（lb）和盎司（oz）。



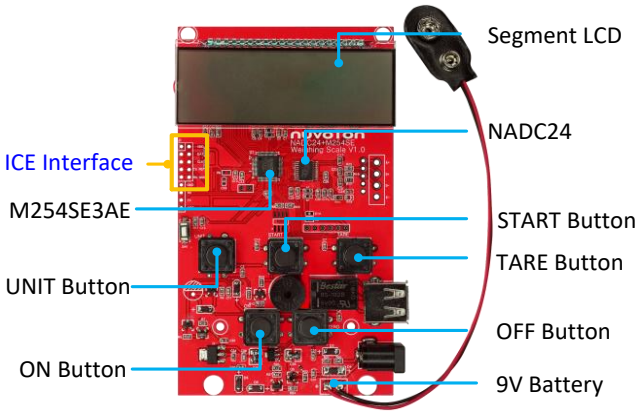
USB* : 只支援 M258 系列

Capacitive Touch Sensing* : 只支援M256/M258系列

图 1-3 M254 方块图

1.4 Evaluation Board 描述

此開發板可以透过按钮来开启电源、关闭电源、扣重以及转换单位。 板子连接 50 公斤重量传感器后，最大测量重量为 50 公斤，分辨率为 1 克。 另外，程序会判断 5 分钟内重量值没有变化，则自动关机，实现省电功能。



Key Description:

- ON Button: To turn on the power and start measurement
- OFF Button: To turn off the power
- START Button: To enter calibration mode
- TARE Button: To start calibration second point and reset the weight on the weighing pan to zero
- UNIT Button: To start calibration first point and switch unit of measure on the scale

Specifications:

- Maximum capacity: 50kg, resolution: 1g
- Operating temperature: 5~35 degrees Celsius
- Power supply: 9V alkaline battery
- Automatic shutdown: 5 minutes without weight change

图 1-4 Evaluation Board 描述

2. 代码介绍

初始化系统频率、开机设定、蜂鸣器设定和 LCD 设定。

```
int main(void)
{
    uint32_t i;
    uint32_t u32DataCount;

    /* Unlock protected registers */
    SYS_UnlockReg();

    /* Init System, IP clock and multi-function I/O. */
    SYS_Init();

    /* Power On */
    GPIO_SetMode(PC, BIT0, GPIO_MODE_OUTPUT);
    GPIO_SetMode(PD, BIT15, GPIO_MODE_OUTPUT);
    PWR_OFF_SIG = 1;
    PWR_SCAN = 1;

    /* Buzzer */
    GPIO_SetMode(PA, BIT8, GPIO_MODE_OUTPUT);
    Beep_20ms();

    /* Init LCD multi-function pins and settings */
    LCD_Init();
    LCD_Start();
    ……
}
```

初始化用于显示的定时器和用于除错的 UART。

```
    ……
    /* Set timer to 333 ms */
    TIMER_Open(TIMER0, TIMER_PERIODIC_MODE, 3);

    /* Enable timer interrupt and run */
    TIMER_EnableInt(TIMER0);
    NVIC_EnableIRQ(TMR0_IRQn);
    TIMER_Start(TIMER0);

    /* Configure UART0: 115200, 8-bit word, no parity bit, 1 stop bit. */
    UART_Open(UART0, 115200);
    ……
```

初始化 SPI 并设定 NADC24。

```
……
/* Init SPI */
SPI_Init();

/* Initial NADC24 */
Reset_NADC24();
NADC24_Calibration_and_Initial(CALI_NADC24);
NADC24_Calibration_and_Initial(INIT_NADC24);

/* IO Interrupt */
GPIO_SetMode(PF, BIT15, GPIO_MODE_INPUT);
GPIO_EnableInt(PF, 15, GPIO_INT_FALLING);
NVIC_EnableIRQ(GPF_IRQn);
……
```

初始化按钮并读取校准数据。

```
……
/* GPIO */
GPIO_SetMode(PC, BIT1, GPIO_MODE_QUASI);    /* BUTTON_START   */
GPIO_SetMode(PC, BIT4, GPIO_MODE_QUASI);    /* BUTTON_UINT    */
GPIO_SetMode(PC, BIT3, GPIO_MODE_QUASI);    /* BUTTON_TARE    */
GPIO_SetMode(PC, BIT5, GPIO_MODE_INPUT);    /* BUTTON_ON      */
GPIO_SetMode(PC, BIT2, GPIO_MODE_QUASI);    /* BUTTON_OFF     */

/* Print */
printf("Nuvoton M254 + NADC24 for weighing scales.\n");
printf("CPU @ %d Hz\n\n", SystemCoreClock);

/* Read the calibration data */
Read_Data_from_APROM(Flash_Data);
Calibration_0g = Flash_Data[0];
printf("Calibration_0g   = %d\n", Calibration_0g);
Calibration_2000g = Flash_Data[1];
printf("Calibration_2000g = %d\n", Calibration_2000g);

/* Start conversion */
SPI_Send_ADC_Commnad(ADC_START_CONVERSION_CMD);

/* Button status */
Button_Start_Status = BUTTON_START;
Button_Uint_Status  = BUTTON_UINT;
Button_Tare_Status  = BUTTON_TARE;
Button_On_Status    = BUTTON_ON;
Button_Off_Status   = BUTTON_OFF;
……
```

侦测按钮（下降沿）用于开启电源、关闭电源、扣重、校准和单位转换。

```
……
while (1)
{
     /* Power off */
    if (Button_Off_Status && !BUTTON_OFF)
    {
        printf("Power Off ...\n");
        PWR_OFF_SIG = 0;
        BUTTON_ON = 1;
        PWR_SCAN = 0;

        /* Buzzer */
        Beep_20ms();
    }

    Button_Off_Status = BUTTON_OFF;

    if (Button_Start_Status && !BUTTON_START)
    {
        Measurement_Flag = 0;
        printf("Enter calibration mode ...\n");
        LCDLIB_Printf(ZONE_SevenSeg_DIGIT, "0     ");
        LCDLIB_SetSymbol(SYMBOL_g,  0);
        Calibration_Mode_Flag = 1;

        /* Buzzer */
        Beep_20ms();
    }

    Button_Start_Status = BUTTON_START;

    /* Calibration mode */
    if (Calibration_Mode_Flag)
    {
        /* Calibration 1P */
        if (Button_Uint_Status && !BUTTON_UINT)
        {
            printf("Calibration 1P ...\n");
            LCDLIB_Printf(ZONE_SevenSeg_DIGIT, "1     ");
            LCDLIB_SetSymbol(SYMBOL_g,  0);
            Calibration_1P_Flag = 1;

            /* Buzzer */
            Beep_20ms();
        }

        Button_Uint_Status = BUTTON_UINT;
        /* Calibration 2P */
        if (Button_Tare_Status && !BUTTON_TARE)
        {
            printf("Calibration 2P ...\n");
            LCDLIB_Printf(ZONE_SevenSeg_DIGIT, "2     ");
            LCDLIB_SetSymbol(SYMBOL_g,  0);
            Calibration_2P_Flag = 1;
```

```
            /* Buzzer */
            Beep_20ms();
        }

        Button_Tare_Status = BUTTON_TARE;

    }
    else
    {
        /* Measurement */
        if (Button_On_Status && !BUTTON_ON)
        {
            printf("\nMeasurement ...\n");
            Measurement_Flag = 1;

            /* Buzzer */
            Beep_20ms();
        }

        Button_On_Status = BUTTON_ON;
        /* Unit translation */
        if (Button_Uint_Status && !BUTTON_UINT)
        {
            printf("Change unit ...\n");
            Change_Unit_Flag++;

            if (Change_Unit_Flag >= 4)
                Change_Unit_Flag = 0;

            /* Buzzer */
            Beep_20ms();
        }

        Button_Uint_Status = BUTTON_UINT;
        /* Tare weight */
        if (Button_Tare_Status && !BUTTON_TARE)
        {
            printf("Tare weight ...\n");
            LCDLIB_SetSymbol(SYMBOL_TARE, 1);
            Tare_Weight_Flag = 1;

            /* Buzzer */
            Beep_20ms();
        }

        Button_Tare_Status = BUTTON_TARE;
    }
}
……
```

3. 软件与硬件需求

3.1 软件需求

- BSP 版本
  - M251_M252_M254_M256_M258_Series_BSP_CMSIS_V3.02.005
- IDE 版本
  - Keil uVersion 5.36

3.2 硬件需求

- 电路组件
  - NADC24_M254_Weighing_Scale_V1.0
- 线路示意图
  - 将 UART0 TX (PB.13) pin 脚连接到 PC UART RX，以显示范例代码的执行结果。

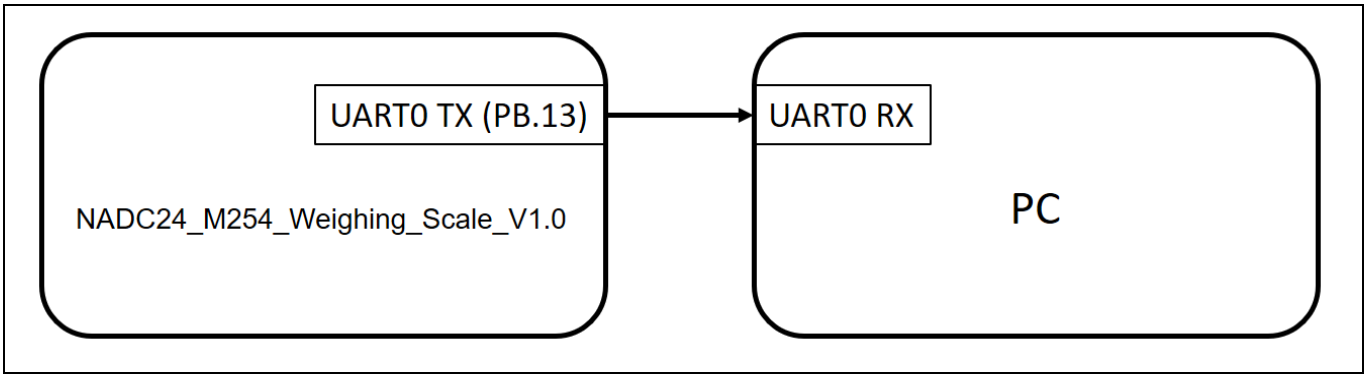


图 3-1 线路示意图

4. 目录信息

EC_NADC24_M254_Weighing_Scales_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
LCDLib	COM/SEG LCD source files
StdDriver	All peripheral driver header and source files
SampleCode	
ExampleCode	
Project	Source file of example code
Schematic	Schematic file of evaluation board

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 *NADC24_M254_Weighing_Scales.uvprojx*。

2. 进入编译模式接口

   - 编译
   - 下载代码至内存
   - 进入 / 离开除错模式

3. 开启超级终端机，超级终端机設定如图 5-1 所示：

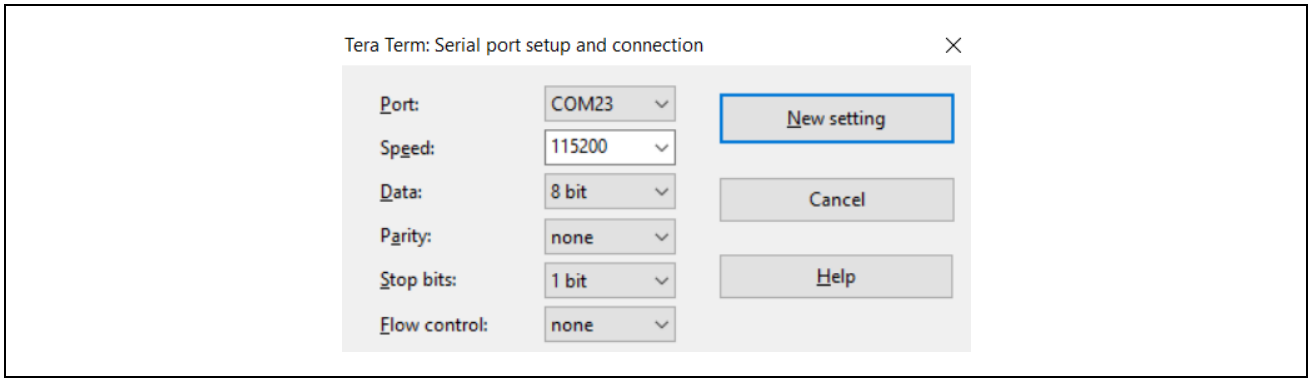


图 5-1 超终端机设定

4. 进入除错模式接口

   - 执行代码

5. 超级终端机显示系统信息

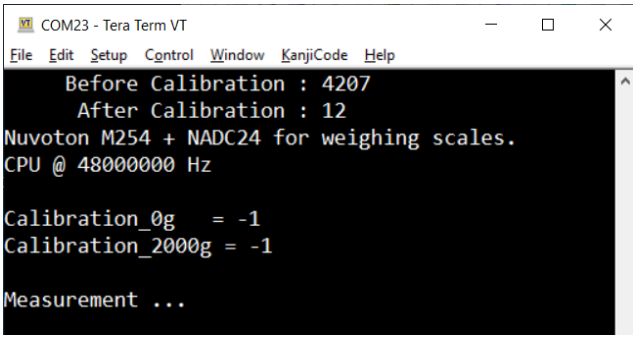


图 5-2 系统信息

6. 按下 **START** 进入校正模式

图 5-3 校正模式

7. 按下 **UNIT** 开始第一点校正

图 5-4 第一点校正

图 5-5 第一点 ADC 数值

8. 放置 **2Kg** 砝码到 LOADCELL 平台

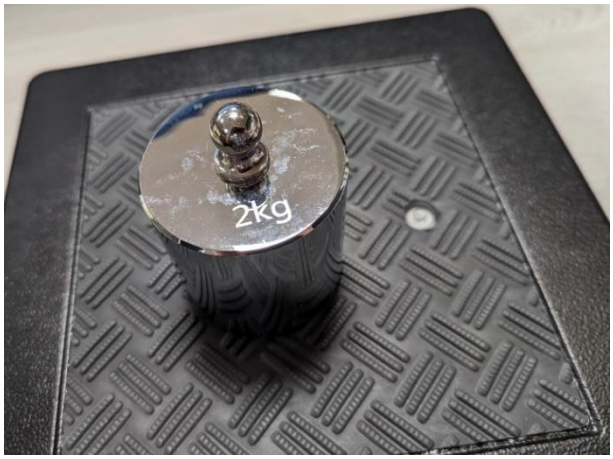


图 5-6 2Kg 砝码

9. 按下 **TARE** 开始第二点校正

图 5-7 第二点校正

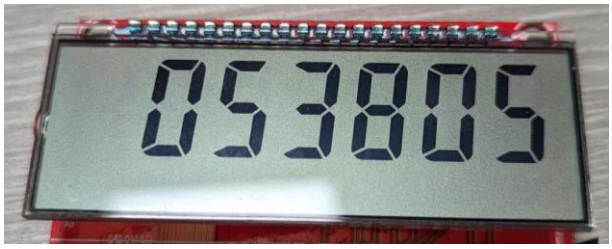


图 5-8 第二点 ADC 数值

10. 按下 **ON** 开始量测重量

图 5-9 量测结果

6. 修订纪录

Date	Revision	Description
2023.10.04	1.00	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.